UNIVERSITA' DEGLI STUDI DI PAVIA

FACOLTA' DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

DOTTORATO DI RICERCA IN BIOINGEGNERIA E BIOINFORMATICA

XXVII CICLO - 2014

INNOVATIVE TECHNOLOGIES FOR THE MANAGEMENT AND INTERPRETATION OF SECOND GENERATION SEQUENCING DATA

PhD Thesis by

Ivan Limongelli

Advisor:
Prof. Riccardo Bellazzi

PhD Program Chair:
Prof. Riccardo Bellazzi

There's no gene for fate.

(GATTACA, 1997)

..but many others to be worried..

(unknown)

Non esiste un gene per il destino.

(GATTACA, 1997)

..ma molti altri di cui preoccuparsi..

(unknown)

Abstract (Italiano)

La diffusione delle tecnologie di nuova generazione per il sequenziamento del DNA ha rivoluzionato il mondo della ricerca biomedica.

La crescente capacità di *throughput*, in termini di basi sequenziate nell'unità di tempo, insieme ai costi sempre più ridotti per base è stato il *trend* degli ultimi dieci anni ed ha portato al sequenziamento di migliaia di esseri viventi.

A oggi, siamo in grado di leggere l'intero genoma di una persona base per base, identificare quelle varianti genomiche che possono, ad esempio, spiegare una singola patologia genetica o che aumentano in maniera significativa il rischio di sviluppare un tumore. Possiamo suggerire quali farmaci potrebbero rivelarsi più efficaci dato il profilo genomico del paziente e quali terapie anti tumorali potrebbero fallire a causa di una particolare combinazione di mutazioni, espressione ed evoluzione di geni sequenziati a partire da particolari tessuti.

La "medicina personalizzata" o "medicina di precisione", che consiste nella pratica medica ottimizzata per i dati clinici e molecolari del paziente, ha tutte le carte in regola per essere applicata in ambito diagnostico, prognostico e terapeutico, al patto che essa venga supportata da un'adeguata tecnologiadi Bioinformatica.

Questa tesi vuole essere un contributo in questa direzione affrontando due importante sfide che ciascun laboratorio, facente uso di queste tecnologie a scopo di ricerca o clinico, deve affrontare: la *gestione* e l'*interpretazione* dei dati di sequenziamento, con particolare focus sulle varianti genomiche.

Nel Capitolo 1 sono discusse brevemente le motivazioni dell'attività di ricerca di questa tesi e sono riassunte le soluzioni sviluppate.

Nel Capitolo 2 è presentato lo stato dell'arte della tecnologia di sequenziamento, le sue applicazioni ed il conseguente impatto sulla comunità scientifica negli ultimi anni. Sono quindi introdotte le tematiche della gestione e dell'interpretazione dei dati genomici, e infine spiegati nel dettaglio alcuni dei principali database per i dati genomici ed algoritmi adibiti all'annotazione e all'interpretazione di varianti genomiche.

Nel Capitolo 3 è discussa l'architettura del sistema sviluppato per gestire i campioni sequenziati e le relative varianti genomiche. Vengono presentate

le principali tecnologie e le risorse utilizzate. Viene poi descritta l'interfaccia web sviluppata. Dopo aver discusso delle limitazioni di questo sistema, si introduce e si descrive una nuova architettura per la memorizzazione di varianti genomiche e le relative componenti tecnologiche. Sono infine discussi i risultati di test effettuati con tale piattaforma.

Nel Capitolo 4 viene presentato l'algoritmo realizzato per classificare le varianti genomiche descrivendo componenti e risorse utilizzate. Vengono discussi i risultati ottenuti in termini di confronto con altri algoritmi di predizione e casi particolari dove l'algoritmo mostra il suo valore aggiunto.

Nel Capitolo 5 sono presentate le applicazioni cliniche del primo sistema di gestione delle varianti genomiche. Per ciascun caso, sono riportati il background clinico, i metodi ed i risultati.

Nel Capitolo 6 sono presentate le conclusioni e possibili sviluppi futuri.

L'attività illustrata nel Capitolo 3, è frutto della collaborazione con Angelo Nuzzo per il primo sistema di gestione delle varianti genomiche e della collaborazione con Matteo Gabetta, Daniele Segagni, Ettore Rizzo e Riccardo Bellazzi per il secondo sistema basato su NoSQL e i2b2.
La parte metodologica dell'algoritmo per la predizione delle varianti genomiche presentato nel Capitolo 4 è stata implementata in collaborazione con Simone Marini.
Gli studi sperimentali presentati nel Capitolo 5 sono stati effettuati dal Dipartimento di Genetica Medica dell'Università di Pavia con la tecnologia di sequenziamento (Illumina Genome Analyzer IIx) dell'IRCCS Istituto Nazionale Neurologico C.Mondino di Pavia.

Abstract (English)

The advent of new generation sequencing technologies has revolutionized the scenario of omics-research.

Increasing throughput in terms of sequenced bases and reducing costs per base have been the trends in the latest ten years and allowed the sequencing of thousands of living beings.

We are now able to read the whole genome of a person at single-base resolution in a day, looking for the genomic variants that can explain his particular disease or that can increase significantly his risk of cancer. We can suggest which drugs would be more efficient on the basis on the patient genomic profile and which cancer therapies may be fail due to his particular expressions and mutations of the genes in particular tissues.

"Personalized medicine" or "precision medicine", which is the medical practice tailored on clinical and molecular patient data, has now all the potential to be applied into diagnostic, prognostic and therapeutic patient clinical course, in particular if supported by a solid Bioinformatics technology.

This thesis aims to be a contribution to this achievement by dealing with two important technological challenges that each research or diagnostic molecular laboratory making use of new sequencing technologies has to face: the *management* and the *interpretation* of sequencing data, focusing on genomic variants.

In Chapter 1 the motivations of the research activity of this thesis are briefly discussed along with the adopted solutions and their practical applications.

In Chapter 2 the state of the art of the technology dealing with sequencing and its impact on the scientific community through several applications are described. Data management and interpretation issues are introduced. The most important genomic databases, variant annotation and prediction algorithm technologies are discussed as well.

In Chapter 3 the overall architecture of the system developed to manage sequenced samples and genomic variants is described. The main technologies and resources adopted are discussed and the web interface is presented. Drawbacks of this system are highlighted, the paradigm of a new developed system is introduced and its underlying technologies are described in details. Test results on this system are finally discussed.

In Chapter 4 the algorithm developed to classify genomic coding variants is described by providing details about its components and used data resources. Results in terms of comparison to other existing algorithms and the cases in which our algorithm best performs are discussed.

In Chapter 5 the applications of the variants management system developed in the thesis are presented. For each application theclinical background, methods and results are reported.

In Chapter 6 concluding remarks are presented and future challenges and directions are discussed.

The activity illustrated in Chapter 3,was carried out in collaboration with Angelo Nuzzo for the first developed genomic variant management system and in collaboration with Matteo Gabetta, Daniele Segagni, Ettore Rizzo and Riccardo Bellazzi for the second one, based on NoSQL and i2b2.
The methodological part of the algorithm for variant prediction presented in Chapter 4 was carried out in collaboration with Simone Marini.
The experimental studies presented in Chapter 5 were carried out by the Department of Molecular Medicine, University of Pavia by using the sequencing technology (Illumina Genome Analyzer IIx) of the IRCCS National Neurological Institute C. Mondino in Pavia, Italy.

Contents

Chapter 1

Introduction

New generation sequencing technologies are able to produce huge amount of data related to the nucleotide sequences of the DNA molecule.

The latest sequencing machines can reach up to ten Terabytes of sequence data in a single run experiment making possible to read the DNA, in parallel, of hundreds of samples.

It is clear that a fundamental issue is to address the management of such large amount of data from a computational, storage and accessibility point of view.

Assuming we have the hardware and the software infrastructure to process and reduce sequencing data into a human readable format (e.g. genomic variants) we need to store them in such a way that would be possible to answer to several questions posed by the geneticists. Herby some examples:

- *Which are the genomic variants of the sample A?*
- *Which are the genomic variants of the sample A in the gene B?*
- *Does the sample A have the genomic variant C known in literature ?*
- *Which are the genomic variants of the sample A in the gene B that the samples D,E,F and G do not have?*
- *Which are the genomic variants of the sample A with an allele frequency below a given threshold respect toa selected subset of samples?*

By looking at these questions it is possible to derive the main requirements needed by the software systems that should be able to manage and retrieve genomic variants:

- A data model able to link genomic variants to samples

- A data model able to link variants to their genomic context (e.g. genes)
- An accessible user interface allowing for data extraction

Intuitively, the system needs to integrate different genomic data sources, e.g. public omics-database and repositories, and integrate them with the sequencing data of interest, in our case, genomic variants.

Let’s consider another common task:

- *Which are the genomic variants of the sample A in gene B and gene H that could affect the protein stability ?*

By sequencing a whole human genome we have to face with a number of variants in the order of millions. Even if we filter out intergenic or non-coding variants, we deal with dozen of thousands variants per sample.

Considering that only a small number of variants of the human genome have been linked with a known phenotypic trait [1] and that for a certain kind of diseases only one or two variants can be fully or partially explanatory, this task is similar to looking for a needle in a haystack.

Our prior knowledge about biological pathways and genes involved in a particular disease can help us to create a subset of genes that may allow us in going further with a deeper investigation, but typically there is the need to distinguish and weight those rare or unseen genomic variants that can alter the protein structure and function from those that do not.

Despite the existence of several algorithms made available in the last years to this end, more accurate, fast, exhaustive and accessible solutions are needed.

1.1. Genomic Variants Management Systems

To store and manage genomic variants, a web-based interactive framework was developed. Based on a J2EE architecture, it relies on a Relational Database Management System (RDBMS), that is MySQL.

Genomic variants are uploaded trough a web interface along samples and experiments data. The data model of the RDBMS was built in order to integrate several genomic resources allowing to enrich or “annotate” variants with useful information such as mRNA transcripts, genes and allele frequencies from public variant databases.

An import data layer is able to import standard files in the Variant Calling Format (VCF) and additional modules that allow to compute, at the importing stage, several variant attributes related to its genomic position and type, such as possible changes in the protein sequence.

Other application modules are able to generate automatic queries to web resources, such as the University of California Santa Cruz (UCSC) database or variant prediction tools such as PolyPhen2 [2]and MutationTaster [3]. Once queries are performed, results are stored into the database and can be re-used by further analysis.

Thanks to the web interface, the user can select the subset of samples for the variant retrieval and another one to compute data aggregates for each reported variant (e.g. allele frequency in the subset). Several filtering criteria on the variant attributes can be set in order to retrieve only the subset of variants of interest. Final results can be exported in a tab-delimited plain text files.

Such developed system allowed to manage 437 sequenced samples and 33,799,523 genomic variants for a total amount of 293GB of data on a single workstation with an Intel i3 CPU and 4GB of RAM.

The system allowed extracting and combining data on the filtering criteria and to finally end up with a candidate genomic variant lists for each study. In particular, for several disease studies (see Chapter 5) it was possible to determine the underlying genetic causes.

Despite the system was built to be light in terms of CPU and RAM requirements, it showed its drawbacks in terms of computational time performances. Furthermore, the choice to integrate portions of public genomic databases in order to annotate genomic variants, can introduce data consistency issues when the same genomic databases have to be up to date.

For these reasons, a completely different approach was developed, both in terms of workflow and technology.

The main idea was to consider the annotation task as a pre-processing step without involving database resources and to pre-compute each possible variant attribute at this stage working on a high-parallel environment. The public genomic databases containing features tracks are represented by text files, compressed in a binary format and indexed by genomic positions. The annotation step is performed by querying the indexed resources one-by-one through genomic positions and variant type. Because each variant is independent from the others, the process can be high parallelized on batches of variants. Once this step is completed, data are imported into the NoSQL database CouchDB in form of JSON files, where each JSON represents a genomic variant with every pre-computed annotation. Each variant attribute is then indexed for a fast retrieval of the JSON document when a single attribute is queried on its values. Complex queries (on multiple fields) are obtained by the combination of each result set on a single attribute. Import and query processes showed high performances in terms of computational time if compared to the relational database.

The system interfaces with the i2b2 [4] framework. An ad hoc software module guarantees the communication between i2b2 and the CouchDB in order to execute the queries on the database and works on an XML-structured messaging standard sent by HTTP both to build queries and retrieve results. On the i2b2 webclient, an ad-hoc plug-in, based on a visual programming, allows the user to build simple and complex queries and has the power to couple patient result sets extracted by the i2b2 phenotype queries with the genetic data stored in CouchDB.

The system was tested on 500 sequenced samples from 1000 Genome Project [5]public resources resulting in about 1,500,000 genomic variants for a total amount of 160 GB on a single AWS EC2 machine with 8 virtual CPUs and 16 GB of RAM (for data query). Both importing and querying times were very promising; furthermore, the space of the possible queries increased and the system usability improved respect the previous system because of the coupling with the i2b2framework and of a new intuitive visual plug-in.

1.2. Genomic Variants Interpretation

In order to deal with the multitude of genomic variants identified by second generation sequencing experiments and to assign to each one a score that correlates with the possible perturbation induced in the codified proteins, a software that aims to classify genomic variants was developed.

The main idea relies on the use of the known changes in amino acid sequences linked to several diseases, assuming that each is fully explanatory of the pathology and therefore causes a strong modification of the protein behavior. The amino acid sequence changes were represented under a discrete form by using Pseudo Amino Acids Composition (PseAAC) [6]. This allowed to train a Random Forest [7]classifier on the aforementioned known genomic variants by using the PseAAC values as features.

The classification results were combined with two well-known variant prediction algorithms in order to improve accuracy, i.e. PolyPhen-2 and SIFT [8],which rely on different approaches.

The algorithm, we called PaPI, showed prediction performances in terms of accuracy significantly greater than the other considered tools on three different independent test sets and as an additional proof of concept it was run on several well-known pathogenic variants for which both PolyPhen-2 and SIFT were discordant, giving back the right classification for each case.

In order to let PaPI be accessible by the scientific community, a web service was developed. The web interface allows uploading data about a single genomic variant or a list up to thousands of them. Asynchronous processes manage the requests that are put in queues depending on the analysis type. Results, once ready, are sent back by e-mail.

The algorithm logic was implemented by Java, Perl and Weka [9] and the web service by a J2EE architecture.

Chapter 2

Background

This Chapter introduces, in brief, the principles and technologies of new generation sequencing systems and their impact on the scientific community through their applications in the last years.

Several open issues about these technologies from the bioinformatics point of view are discussed, highlighting those on data management and interpretation.

Last sections describe a) the genomic databases that have been strategic for the variant management system (VMS) development and b) different kind of variant prediction tools that have been integrated into VMS and used to the develop the variant prediction algorithm discussed in Chapter 4.

2.1. New Sequencing Technologies

Since their introduction in 2004 with the Roche 454 pyrosequencing machine, the so-called “Next Generation Sequencing” (NGS) technologies have been undergoing a tremendous development.

The Human Genome Project, carried out by the International Human Genome Consortium, needed more than ten years (1991-2003) to sequence the whole genome of a human being and cost about 1 billion US dollars. In 2014 state-of-the-art instruments process a whole genome in less than a week and for nominally less than ten thousand dollars [10].

As a consequence, these technologies had an extraordinary impact on scientific community and led to ever-growing investments by the major biotech vendors: today, NGS market has a worth of $2.5 billion, poised to reach $8.7 billion by 2020.

Moore's law states that for computer industry the compute power doubles every 24 months. Sequencing technologies have out paced Moore's law by far (see
Figure 1).

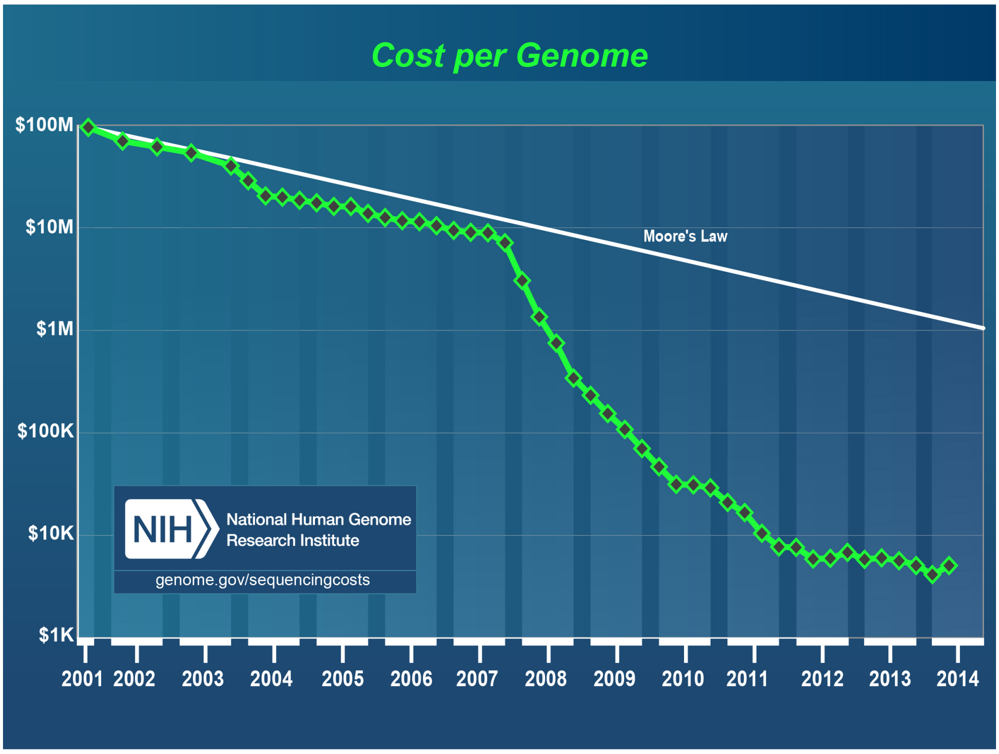


Figure 1. Sequencing technologies costs and Moore'law. Source:http://www.genome.gov/sequencingcosts/

Sequencing instruments can be distinguished by their implemented techniques that include pyrosequencing, sequencing-by-ligation and sequencing-by-synthesis developed by the three leading biotech companies Roche, Applied Biosystems (now Life Technologies) and Illumina, respectively.

In the last five years, Illumina sequencing instruments gained market dominance with a NGS market share of 71% during 2013.
The Illumina success can be explained by the well-balanced combination between sequencing accuracy and reproducibility plus the market strategy of product segmentation (and prices) that allowed reaching a wide gamma of customers with different needs. Moreover, the Illumina's sequencer MiSeq was the first new generation sequencer to be authorized by Food and Drug Administration (FDA) for broad clinical use.

Nevertheless, sequencing technology is under a continuous evolution, and the recent introduction of a further new method based on single molecule real time sequencing by Pacific Biosciences seems to be a milestone for the third generation sequencing platforms in the next years.

Explain the principles of the different new sequencing technologies is beyond the scope of this thesis; therefore, only a brief description of the methodology at the basis of the most used sequencing platforms, i.e. Illumina's ones, is reported hereby.
Notably, the same sequencing technology was used in the application studies discussed in Chapter 5.

2.1.1. Illumina Sequencing

Illumina (San Diego, CA) is an American company that since 1998 develops systems for the analysis of genetic variation and biological function. Very soon, Illumina began to offer micro-array based products such as SNP genotyping, gene expression and protein analysis.
In 2007 it acquired the Solexa company that developed a new genome sequencing technology, the Solexa machine.
In seven years (2007-2014) Illumina developed seven different sequencing platforms including their updated versions.
The state-of-the-art of Illumina sequencing products are: MiSeq, NextSeq500, HiSeq2500 and HiSeq X Ten ordered by increasing throughput capacity in terms of sequenced bases.
In Table 1, the evolution of Illumina technology in the last seven years is reported: costs and throughput trends confirm the statements of the previous section.

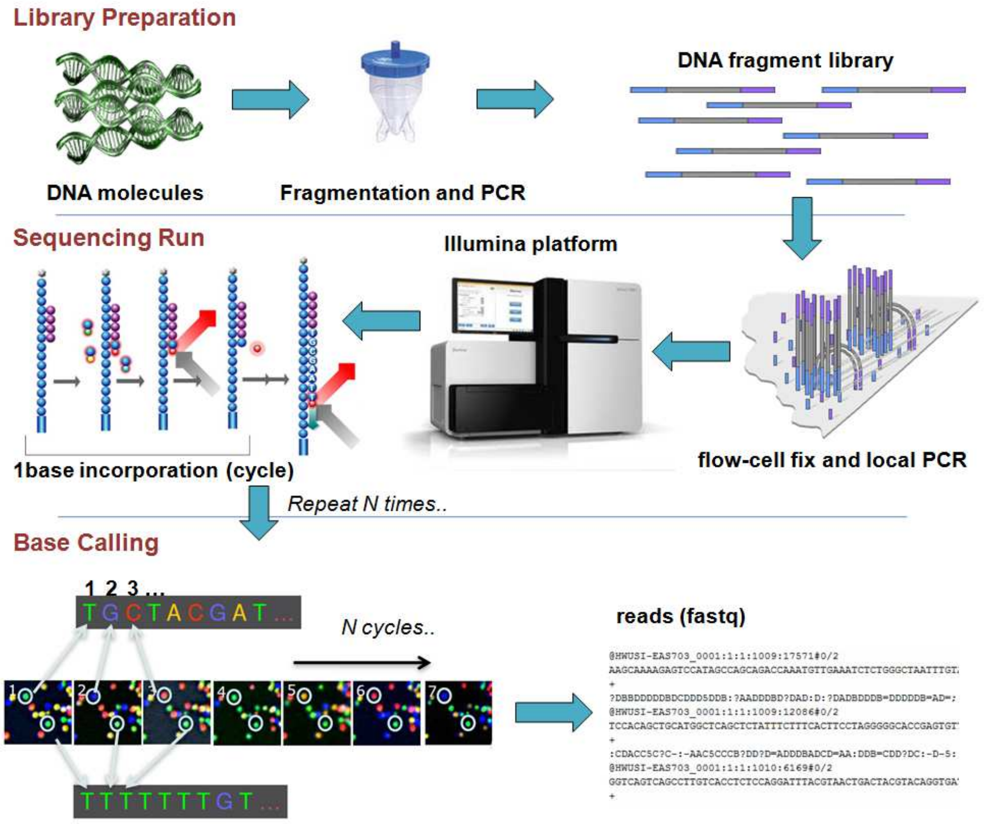
Year	2007	2009	2011	2012	2014
Platform	GA	GAIIx	MiSeq, HiSeq2000	HiSeq2500	NS500, XT
Costs* ($)	800K	16K	- , 6K	5K	4K, 1K
Output (GB)	10	80	15 , 200	600	129 , 1800

Table 1. Development of Illumina sequencing technology in the last seven years. GA= Genome Analyzer; NS= NextSeq; XT=HiSeq X Ten; K=1x10^3; GB=billions of sequenced bases. *Costs to sequencing a whole human genome.

We can split the Illumina sequencing process into three main steps:

- DNA library preparation
- Sequencing run
- Base calling

Note that the first step is almost the same for every other sequencing platform. The three steps are briefly described below and Figure 2represent an overview of the whole process.



```
@HWUSI-EAS703_0001:1:1:1009:17571#0/2
AAGCAAAAGAGTCCATAGCCAGCAGACCAAATGTTGAAATCTCTGGGCTAATTTGT
+
?DBBDDDDDBDCDDD5DDB:?AADDDBD?DAD:D:?DADBDDDB=DDDDDB=AD=;
@HWUSI-EAS703_0001:1:1:1009:12086#0/2
TCCACAGCTGCATGGCTCAGCTCTATTTCTTTCACTTCCTAGGGGGCACCGAGTGT
+
:CDACC5C?C-:-AAC5CCCB?DD?D=ADDDBADCD=AA:DDB=CDD?DC:-D-5:
@HWUSI-EAS703_0001:1:1:1010:6169#0/2
GGTCAGTCAGCCTTGTCACCTCTCCAGGATTTACGTAACTGACTACGTACAGGTGA
+
```

Figure 2. Sequencing process – from DNA sample to sequence reads

2.1.1.1. DNA Fragment Library Preparation

Once extracted from tissue cells, the pool (sample) of DNA molecules is broken into millions of pieces. Nebulization or sonication methods [11]are typically used for this aim. After fragmentation step, DNA sample is amplified by Polymerase Chain Reaction (PCR) technique [12]. Finally, only the fragments within a certain length range are selected through gel electrophoresis, a method able to order DNA fragments by their mass, therefore, their length.

The resulting sample consists of a “library” of DNA fragments ready to be sequenced. Indeed, sequencing technologies, even the newest ones, are not able to read consecutively a high number of bases, but only few hundreds.

The library is fixed by ligation to a glass chip, the so-called flow-cell, the core of the sequencing machine. This is a microfluidic device with few distinct micro channels or billions of nanowells, as in the case of the last Illumina technologies.

Once fragments are fixed on the flow cell surface, a particular in-loco PCR replicates each fragment and generates cluster, that is, a certain number of the very same fragment copies in its neighborhood. This will allow enhancing the signal coming from each fragment/cluster in the sequencing step.

2.1.1.2. Sequencing Run

The flow-cell with the DNA library fixed on it is inserted inside the sequencing platform. Here the sequencing starts and proceeds by cycles.

Each cycle corresponds to read a single DNA nucleotide base for each DNA fragment fixed on the flow-cell, in a parallel fashion. For this reason, new generation sequencing systems are also called “massive parallel” sequencing.

The DNA fragment is read by a technique called sequencing-by-synthesis. In brief, each fragment is literally copied by an enzyme, the DNA polymerases, capable to incorporate a DNA base that is complementary to a given one.

The Illumina technology peculiarity is the capacity to stop the DNA polymerases at each incorporated base, through the use of particular modified nucleotide called “reversible chain terminators”, and to re-start DNA polymerases incorporation for the next one in a process called “single-nucleotide addiction”.

This allows to hit by a laser the flow-cell surface at each cycle (corresponding to a nucleotide incorporation) and to stimulate the fluorescence of the modified nucleotide. In fact, each type of modified nucleotide (A,C,G,T) has a particular fluorophore that emits fluorescence with a specific wave length.

By a Charge-Coupled Device (CCD) camera, four pictures of the flow-cell at each cycle are taken, corresponding to four applied filters able to enhance the fluorescence by its wave length.

For each picture, fluorescence dots are present: a dot represents the fluorescence signal coming from a DNA fragment cluster.

The final result of the sequencing step is a number *N* of flow-cell image quadruplets, where *N* corresponds to the number of bases that are read from each DNA fragment on the flow-cell.

2.1.1.3. Base Calling

Images produced in sequencing step are then analyzed by the Illumina proprietary software installed on a workstation connected to the sequencing machine. This software proceeds to:

a) filter the images background noise
b) enhance luminescence signal for each cluster
c) identify cluster positions
d) assign the most probable base for each cluster at each cycle

The final result of this procedure, or “base calling”, is a multitude of sequences with A,C,G or T characters, corresponding to the DNA bases of each DNA fragment on the flow-cell.
Together with the set of sequences, or “reads”, the quality for each sequenced base is reported as well. This is computed by estimating the probability that the base has been wrongly assigned, on the basis of the difference with the second probable base. Quality scores are finally logarithmically related to the base calling error probability by Phred scale [13].

DNA reads are stored in plain text files, which can be compressed, following the fastq standard format (see AppendixA.1for more details).

2.2. Sequencing Applications

Sequencing platforms basically take DNA fragments in input and generate nucleotide sequences in output.
Nonetheless, the range of the possible sequencing applications is wide.

2.2.1. Same Data for Different Scopes

Each possible application differs from the others in two main aspects:

1. Library preparation
2. Secondary analysis

In the previous section we briefly described sample library preparation. Actually, several steps were omitted on purpose, in order to list only the most common ones, shared by the major part of sequencing applications. Next section discusses in more details one particular application along with its library preparation.

Secondary analysis typically refers to the whole data analysis process afterwards the base calling step previously described. Intuitively, each sequencing application has its own goals in terms of genomic features to detect, and requires ad-hoc data analysis.

Shendure and Aiden [14] listed twenty types of next generation sequencing applications.
Figure 3 summarizes and stratifies them at the level of species, organism, cells and biological mechanisms of the cell. While the earliest sequencing projects aimed to assembly the genome of a particular specie, new technologies enables the study of biological systems at a finer scale: we can explore genomic variations between individual members or at population scale; highlight genetic and epigenetic differences between cells of a single individual; provide insights into several cell processes (Figure 3).

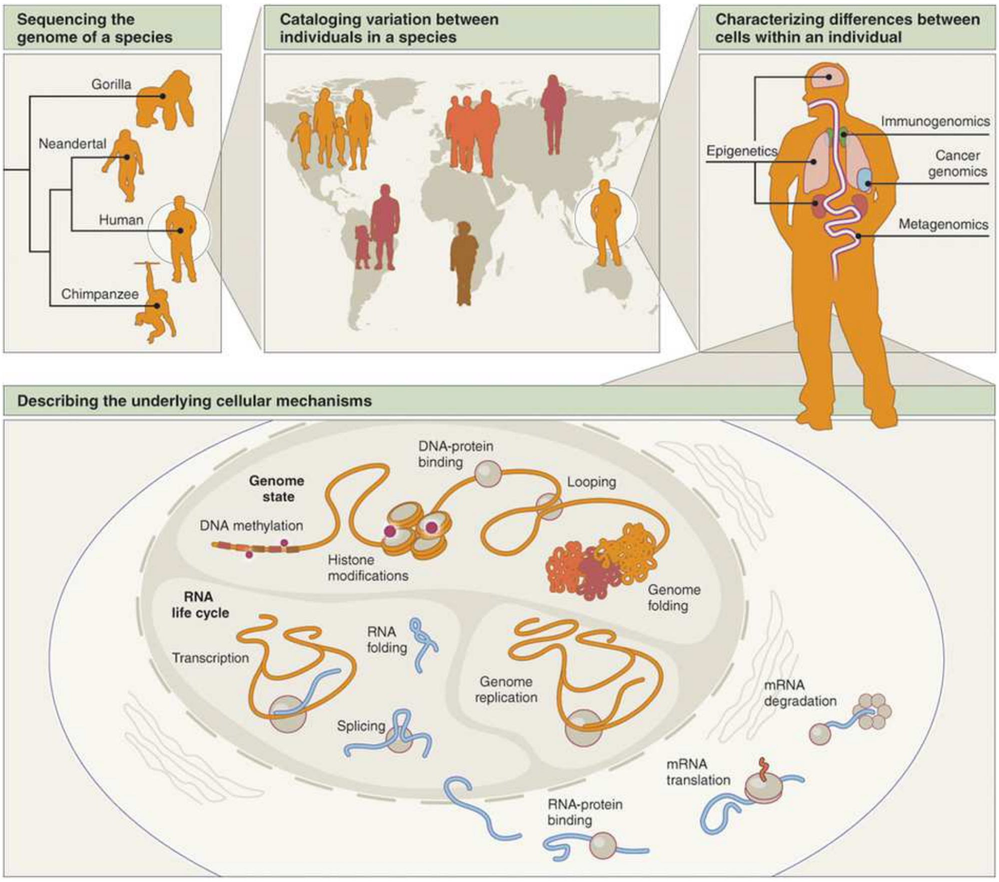


Figure 3. NGS applications for species, individuals, organisms, cells and cell processes [14].

Briefly, next generation sequencing can be applied to study the genomics, transcriptomics and epigenomics of germ and somatic cells.
Figure 4shows several common applications belonging to these three main categories and a hint for each is given below.

- Methylation sequencing

  Used to determine methylation patterns that regulates gene expression [15], library preparation requires to trait DNA with the bisulfate ion (HSO_3^-) through which DNA comes under nucleotide modifications in specific genomic regions (CpG islands).

- ChIP sequencing

  Used to identify DNA binding sites for proteins[16], library preparation requires to capture only the DNA pieces bonded to proteins by the Chromatin Immunoprecipitation (ChIP) technique.

- Whole Genome Sequencing

  Used to identify genomic variants ranging from one to millions bases in length, is the most exhaustive protocol applicable to a genome and library preparation is basically the one showed in Section 2.1.1.1.

- Targeted Enrichment sequencing

  Used to identify genomic variants ranging from one to hundreds bases in length, library preparation requires to capture only the DNA pieces belonging to regions of interest.

- RNA sequencing

  The whole transcriptome is converted into cDNA and by selecting fragments by range size, a specific type of RNA can be analyzed: mRNA sequencing aims to identify differentially expressed transcripts, splice-junctions and new transcripts [17]; micro-RNA (miRNA) sequencing aims to identify differentially expressed miRNAs, predict novel miRNAs and mRNA targets[18].

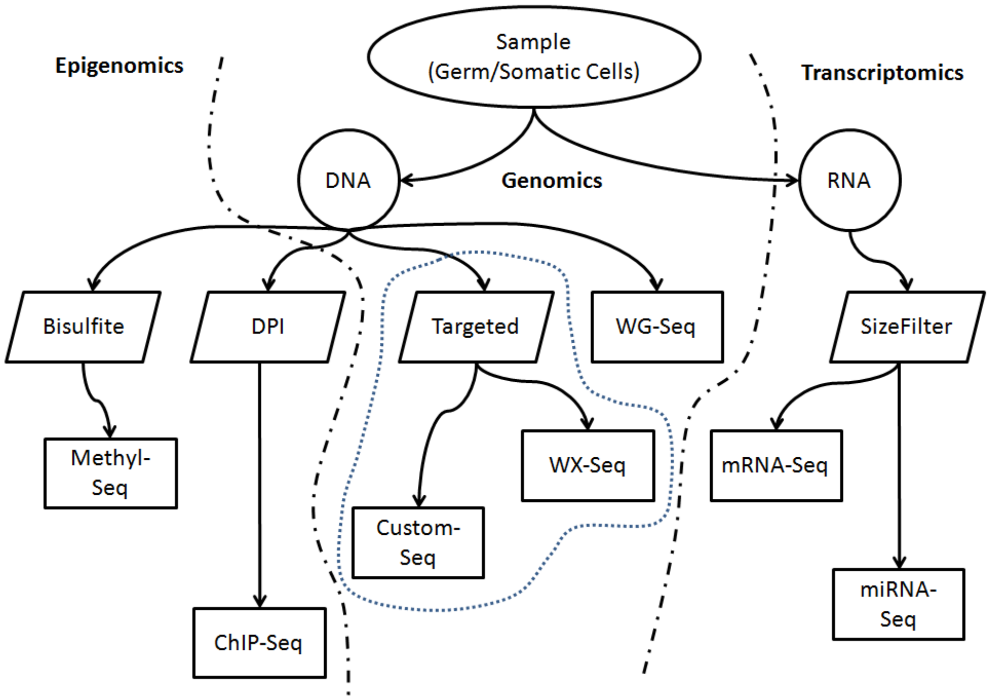


Figure 4. Main NGS applications. Seq=Sequencing; Methyl=Methylation;DPI=DNA-Proteins Interactions; WX=Whole-Exome;WG=Whole-Genome.

Among these applications, targeted enrichment sequencing is the only one discussed in deep, because of the goal of this thesis.

2.2.1.1. Targeted Enrichment DNA Sequencing

Whole genome sequencing is the gold standard to detect all possible genomic variants in the genome, nevertheless its costs, if not amortized by sequenced sample quantities using the last technologies (e.g. HiSeq X Ten), makes its routinely use really hard for the major part of molecular laboratories, which typically deal with a modest number of samples to sequences (hundreds or less per year).

Therefore, in the last five years, cheaper targeted enrichment strategies were widely adopted and successfully applied to a broad range of genetic context [19-22].
By applying targeted enrichment sequencing it is possible to chemically select only those DNA fragments belonging to region of interest (ROI). Typically, ROIs correspond to DNA regions whose sequences (exons) codify for genes.

Whole-exome sequencing (WES) is the most exhaustive targeted enrichment strategy since it allows capturing all exons, therefore genes, of the human genome.
Even if the whole set of human genes corresponds about the 1% of the whole human genome, coding genomic variations are much more likely to have severe consequences than in the remaining 99%[22].

WES has been widely used to discover the genetic cause of Mendelian diseases [23-26], but also to provide insights into complex traits[27-29]and cancer as well[30-32].

An alternative to WES, even cheaper, is to sequence only a reduced panel of genes chosen a priori, basing on the genetic knowledge of the trait of interest. This strategy is actually the most used and appreciated by molecular clinical laboratories, both for costs and practical use in diagnostic[33],since can be ideally treated as an extension of the classic Sanger sequencing technique. Benchtop sequencing instruments such as the Illumina MiSeq are typically used for such purpose.
In the last years, targeted gene panels were massively applied with success to a broad range of complex diseases [34-36]and to study genomic variation even at clone and sub-clone level of somatic cells coming from cancer tissues [37-39].

WES can be considered an extension of targeted genes panel; therefore library preparation and secondary analysis steps are identical.

Library Preparation

The peculiarity of targeted enrichment library preparation is the so-called “capturing” step (Figure 5).

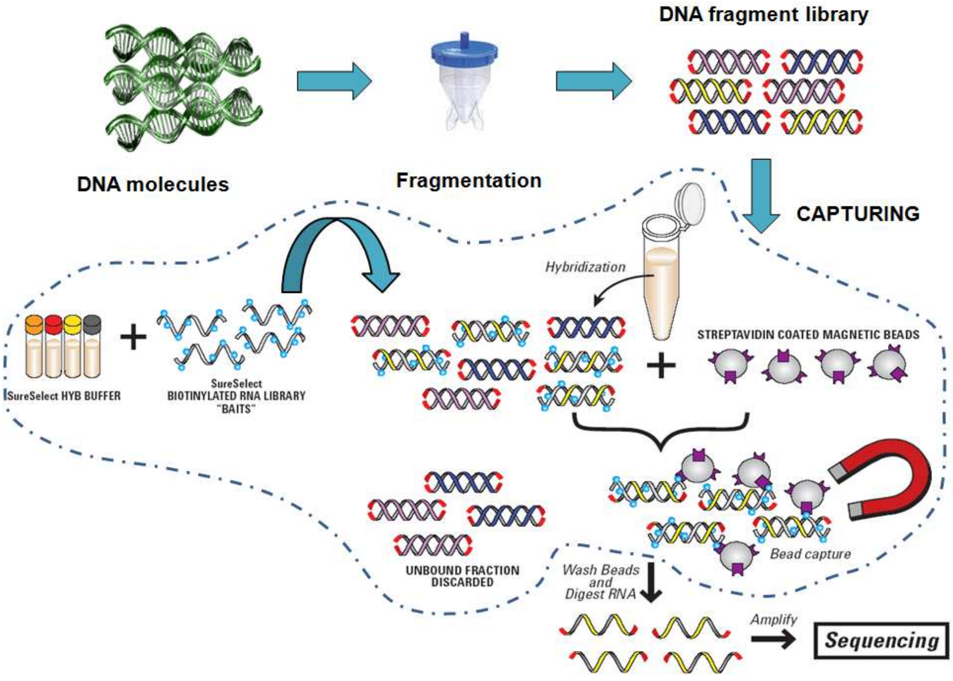

DNA fragment library
DNA molecules
Fragmentation
CAPTURING
Hybridization
STREPTAVIDIN COATED MAGNETIC BEADS
SureSelect HYB BUFFER
SureSelect BIOTINYLATED RNA LIBRARY "BAITS"
UNBOUND FRACTION DISCARDED
Wash Beads and Digest RNA
Bead capture
Amplify
Sequencing

Figure 5. Targeted Enrichment Library Preparation

The DNA library is mixed up with synthesized DNA fragments (probes) able to hybridize only those similar DNA fragments.

Probes have been previously biotinylated and, due to biotin affinity properties, streptavidin beads bind only those coupled DNA fragments consisting of at least one probe.

Beads and probes are therefore washed out and selected DNA fragments are ready to be sequenced.

Secondary Analysis

Once DNA reads have been produced by the sequencing platform, they are processed by a quasi-standard data analysis protocol whose main steps are reported below.

- Genome Reference Mapping
- Mapping Correction

- Variant calling

Genome Reference Mapping

The Human Genome Project ended up in 2001 with the first release of the human genome reference assembly. In the last years, next generation sequencing platforms allowed to sequence many human genomes and allowed to refine the original assembly producing more updated and precise versions of the human genome model as a DNA sequence. The most update version at the time of this writing is the GRCh38 released on February 2014 by the Genome Reference Consortium[40].

Assembling sequencing reads is likely to build up a huge and complicated genomic puzzle. The use of a genomic reference can speed up this process by positioning (mapping) each read to the most likely portion of the reference basing on its sequence similarity.

In the last years lots of next generation sequencing mapping algorithms were developed [41-46], applying heuristic methods with the aim to reach a good compromise between accuracy and time and/or computational performances, giving the high number (billions) of reads to map.

In targeted enrichment protocols a pool of DNA cells is sequenced randomly only for the selected genomic regions: probes used for this aim are exactly designed basing on the reference genome, therefore we expect the major part of reads to map within these regions we define “target”. Moreover we expect them to redundantly map a target genomic locus given the presence of many DNA molecules coming from different cells of the same sample, or given the same artificially DNA replicas due to PCR at library preparation stage.

The number of reads that overlap the same target genomic locus is called “coverage”. Its average across the whole target is a typical indicator used to assess experiment quality.

Mapped reads are stored in a plain-text standard format called Sequence Alignment Map (SAM)that is typically compressed in binary format and indexed in the Binary Alignment Map (BAM)[47].

Mapping Correction

The heuristic reads mapping algorithms have several limitations, especially to correctly map reads over problematic regions of the genome reference[48]. To overcome these issues they are complemented with a series of post-mapping steps such as PCR duplicates removal and more accurate re-mapping over these problematic regions[47, 49, 50].

Variant Calling

Human beings share, on average, the 99.9% of their genome[51]. Variant calling aims to identify the 0.1% of the human genomic differences among which can be included those linked to a trait or a disease.

Genomic differences, we call "variants", are related to the genomic reference used for reads mapping. As previously discussed, each genomic target locus (base) is covered by a certain number of overlapping reads: we expect to observe on the corresponding base "pileup" the same kind of base, given the sequence similarity found by mapping. Actually, mapping algorithms admit a certain degree of freedom in terms of sequence similarity, therefore is possible that each read hold variants respect to the reference. Variants can be of three kinds:

1. Single Nucleotide Variants (SNV)
2. Nucleotides Insertions
3. Nucleotides Deletions

Insertions and deletions are generally grouped under the term Deletion Insertion Variations (DIV) or "indel" due to the fact that is possible to observe, at a single genomic locus, a variant event consisting of an insertion followed by a deletion or conversely.

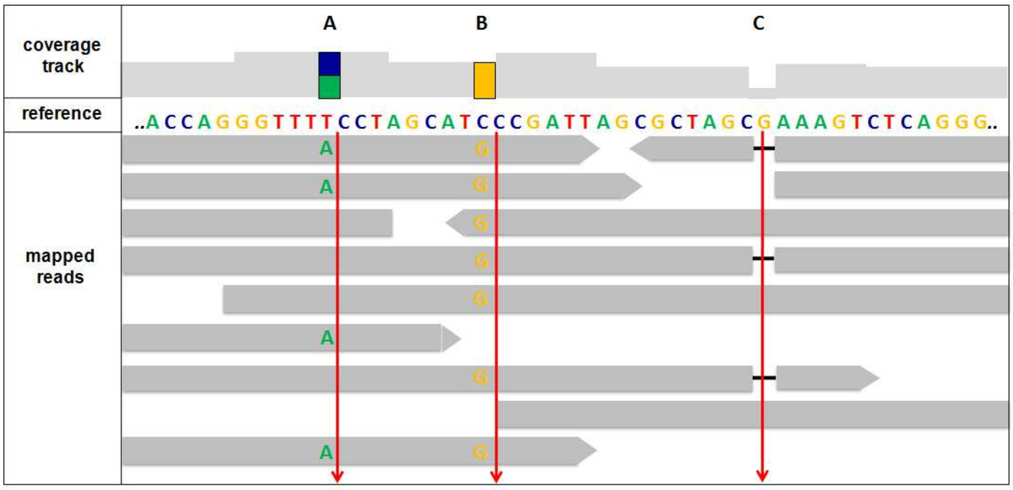


Figure 6. Example of variants along mapped reads. Mapped reads are represented in grey color in case of base equality to the reference, a colored base is shown otherwise. A: SNV in heterozygous state; B: SNV in homozygous state; C: a single base deletion.

Variant calling aims at scanning in an efficient way the whole set of mapped reads and calls a variant at a given genomic locus when at least one overlapping read hold a difference for it.

Sequencing platform base caller error rate, poor quality of sequenced bases and mapping errors introduce noise that heavily can reduce variant calling accuracy, resulting into high false positive rate[1][52, 53]. Therefore is not advisable to rely on single read difference to call a variant, but is essential to combine the information coming from the overall set of mapped reads covering a candidate variant locus.

Another point should be considered: genomic reference is an arbitrary mix of two haploid[2] genomes, while the DNA fragment library is usually derived by a diploid one. This requires determining the so-called "genotype" for a genomic locus: given the variant site, variant calling should be able to assess which is the most reliable combination of two alleles. An allele can be a single base or an indel.

Several variant calling algorithms have been developed and can be divided by methodology: heuristic methods[54, 55], probabilistic frameworks[47, 49, 56], supervised [57]and unsupervised machine learning[58]. Nonetheless scientific community agrees that the optimum variant caller does not exist yet and only ensemble strategies reach the best accuracy[59].

Identified variants can be stored in a plain-text standard format called Variant Calling Format (VCF)[60], see AppendixA.1for more details.

2.3. Challenges in NGS bioinformatics

Next generation sequencing technologies challenge bioinformatics in different aspects including: i) computational resources and tools for data processing ii) data archival and retrieval solutions iii) analysis and interpretation of NGS data.

2.3.1. Computational resources and tools for data processing

The huge amount of genomic data requires appropriate computational resources and tools for data processing. High Performance Computing (HPC) based on physical or virtualized computer clusters along with high parallelized methods have been successfully applied to NGS bioinformatics [61-64]although originally developed to manage huge web data on high parallelized environments [65]. Such systems allow to process genomic data in a reasonable time (e.g. few hours for a whole genome analysis). Recently, an ad-hoc developed processor aims to further reduce

[1] Positive event = presence of a truly variant

[2] Haploidy/diploidy = One/Two sets of chromosomes

computational time to less than 20 minutes for a whole genome with a single hardware and software integrated optimized card [66]. The challenge here is to reduce computational processing time and related costs. Computational time should be lower than sequencing time to avoid bottlenecks. Costs should respect the actual ceiling price of 1000$ per genome (comprehensive of sequencing costs as well).

2.3.2. Data archival and retrieval solutions

Managing data of this magnitude requires a well-defined policy, but assessment of data storage needs is complicated by the variability of data formats [67] despite the use of standards such as BAM and VCF: secondary analysis varies for each kind of application and, for many genomic data types, a standard format does not exist yet. Moreover, metadata regarding NGS experiments, samples, tissues, analysis protocols should be archived as well, in order to ensure results reproducibility. Storing and retrieving in an efficient way this multitude and different type of data requires both hardware and software dedicated solutions. Over the last few years several online, control accessed NGS repositories have been developed [68-70]especially by national or international consortia of research institutions due to the big efforts and resources needed to manage these petabytes of data.

In order to reduce data of orders of magnitude one could think to store and retrieve only a certain subset of processed NGS data, such as secondary analysis results that for some applications, including targeted enrichment, are the identified genomic variants.

Recently, several public and commercial solutions able to store and query genomic variants have been developed[71-74]. The challenge here is to develop efficient genomic variant management systems in terms of costs and computational time, able to integrate genomic and clinical data plus the results from software-based genetic analyses that can be helpful to determine which variant candidates are more likely to be causative of the disease of interest.

2.3.3. Analysis and interpretation of NGS data

The analysis and interpretation of secondary analysis results, also called tertiary analysis, aims to unravel the identified genomic variation and, in case of whole genome/exome sequencing, interpret the large amount of genetic variants by determining those that are likely to contribute to the phenotypic trait under study. Filtering and annotation are two important steps in this sense: filtering consists in removing variants that fit a specific genetic model (e.g. inheritance patterns), while annotation looks up all possible information about variants to fit the biological process [75]. Annotation step typically searches into the existing biological knowledge and uses methods able to infer variant potential pathogenic effects to prioritize the whole set of variants[76]. The challenge here is to develop annotation-prediction tools able to accurately identify disease-related genomic variants both when phenotypic traits are known and when they are not.

The analysis and interpretation of the genomic variation cannot be disjointed from the genomic variant management system (VMS): the latter should be able to integrate the whole set of genomic variants per genome, individual phenotypic traits, variant annotation data, prediction tools results and filtering procedures.

2.4. Genomic Databases

Sequencing and many other high throughput based research projects have generated an explosive growth in biological data, which diversity and complexity revealed them as one of the Big Data sources [77]. As a consequence, the number of genomic databases, aimed to store and publicly share this amount of genomic measures and findings, grew up within their users and User Services (see Figure 7).

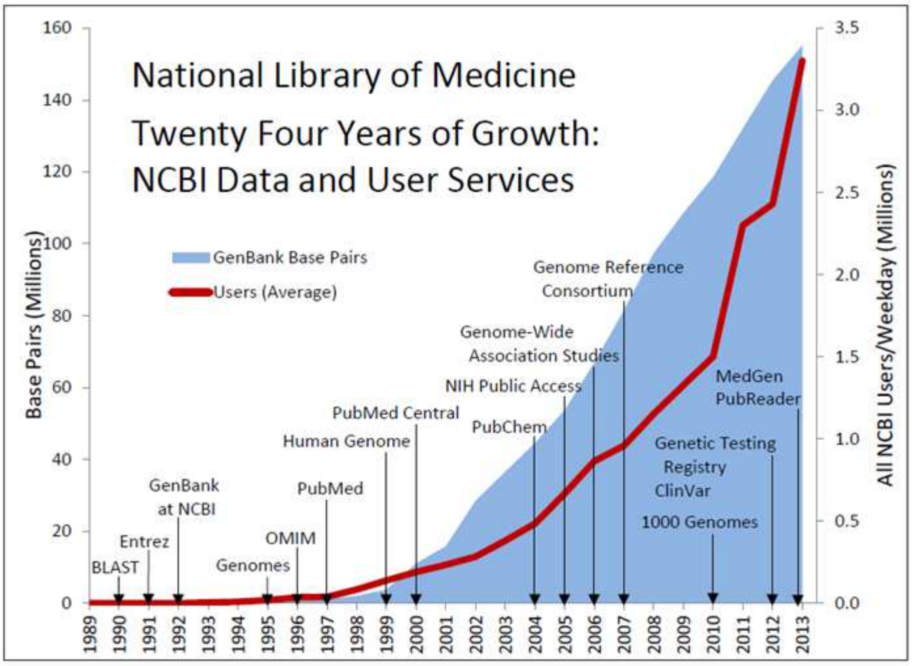


Figure 7. Data for Twenty 24 Years of Growth: NCBI Data and User Services (http://www.nlm.nih.gov/about/2015CJ.html)

Human genomic databases can be catalogued by their content (see Table 2), but drawing precise borderlines between them is not trivial. Moreover, the sheer volume of the raw sequence data in these different repositories has led to attempts to reorganize this information into smaller, specialized databases such as genome browsers [78-80].

In the next sections, one of the main genome browsers and several NGS genomic variant resources will be discussed in more detail.

Category	Brief Description	Examples
Nucleotide Sequences Databases	Collect, annotate, release and exchange original DNA sequence data both for assembled genomes and raw short reads coming from NGS platforms.	• The Sequence Read Archive (SRA) • DDBJ - DNA Data Bank of Japan • European Genome-phenome Archive (EGA)
RNA Sequences Databases	Collect, annotate, release and exchange microRNA, non-coding RNA, transfer RNA and other sequencing-derived transcriptome features.	• HMDDv2.0 • ncRNAs database • RNAJunction • deepBase
Protein Sequences Databases	Provide resources for protein sequences, functional, feature annotations and literature-based evidence attribuitions.	• UniProt • NCBI Protein database • InterPro
Structure Databases	Databases for annotated 3D protein structure models deriving from computational predictions, X-ray crystallography, NMR spectroscopy etc.	• PDBe • SWISS-MODEL Repository
Metabolic and Signaling Pathways	Integrated databases that establish links due to interactions or relationships between genes, higher-level systemic functions of the cell, organism and ecosystem.	• BioCarta • KEGG • Reactome • BioGrid • String
Human Genomes	Databases for gene-specific information. Contain all annotations (nomenclature, map location, gene products, expression etc.) that are constantly updated.	• Ensemble • Entrez Gene • ENCODE • GeneBank • UCSC Genome Browser
Human Genes and Diseases	Resources of sequences data, genomic variants, polymorphisms related to human diseases. In this category general polymorphisms databases are included (e.g. 1000 Genomes Project)	• dbGaP • OMIM • dbSNP • 1000 Genomes Project • HapMap • PharmaGKB

Table 2. Some molecular biology databases categories and examples [81]

2.4.1. UCSC Genome Browser

Genomic variants can be represented at the genomic reference mapping level, which is the format for variant calling as discussed in Section 2.2.1.1. It is therefore straightforward to abstract variants and, in general, genomic annotations by simply considering them as intervals of the genomic reference. This is the principle on which genomic browsers have been developed, with the aim to visualize and browse entire genomes with annotated data from multiple diverse resources.

The University of California Santa Cruz (UCSC) Genome Browser is a web-based platform that repacks genome and gene annotation data sets from GenBank and other databases in order to provide a genomic context for individual genome features, such as genes or disease loci [82].

The user can search for a specific region of a genome, such as a gene, and the sequence plus annotation data are displayed graphically as ‘tracks’ aligned to the genomic reference and grouped by shared characteristics such as gene predictions, comparative genomics or regulatory elements (see Figure 8).

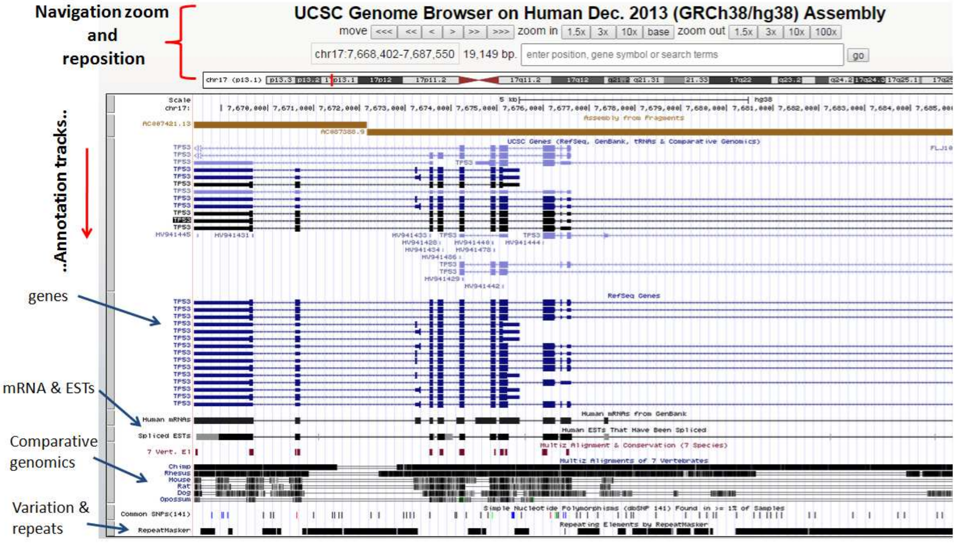

Navigation zoom and reposition
UCSC Genome Browser on Human Dec. 2013 (GRCh38/hg38) Assembly
move <<< << < > >> >>> zoom in 1.5x 3x 10x base zoom out 1.5x 3x 10x 100x
chr17:7,668,402-7,687,550 19,149 bp.
enter position, gene symbol or search terms
go
chr17 (p13.1)
..Annotation tracks..
Scale
chr17:
Assembly from Fragments
UCSC Genes (RefSeq, GenBank, tRNAs & Comparative Genomics)
TPS3
RefSeq Genes
genes
mRNA & ESTs
Human mRNAs from GenBank
Human ESTs That Have Been Spliced
Multiz Alignment & Conservation (7 Species)
Multiz Alignments of 7 Vertebrates
Comparative genomics
Chimp
Rhesus
Mouse
Rat
Dog
Opossum
Simple Nucleotide Polymorphisms (dbSNP 141) Found in >= 1% of Samples
Common SNPs(141)
Repeating Elements by RepeatMasker
RepeatMasker
Variation & repeats

Figure 8.A snapshot from UCSC Genome Browser. For a given genomic region genes, mRNA, evolutionary conservation and variation tracks are displayed and mapped to the genomic reference.

Other annotation tracks include expression, epigenetics and tissue differentiation, phenotype and disease association data and text-mined data from publications.

The UCSC Genome Browser also offers advanced research capabilities such as the UCSC Table Browser [83], built on the top of the Genome

Browser Database. It consists of MySQL relational databases, each containing sequence and annotation data for a genome assembly. Tables within each database can be based on genomic start-stop coordinates or be referenced by internal ids. The whole database set is free downloadable within SQL scripts to build up table structures and indexes.

2.4.1.1. The UCSC Binning Scheme

The UCSC Genome Browser holds annotation data amounting to several Terabytes of tables in the MySQL database [84].

The database has been built under a “read-mostly” purpose and the user queries mainly consist of range queries over genomic intervals identified by three attributes: chromosome, genomic start and stop positions in the chromosome.

In order to efficiently retrieve the whole set of annotation data that map over a requested genomic interval, a suitable binning scheme has been implemented [85].

Let us suppose to perform a query on a Table with the aim to retrieve all data within a genomic interval. The resulting range query would be something similar to:

```
select * from Table where chrom=’chr1’ and
chromStart<20000 and chromEnd>10000
```

To speed up the data retrieval it is possible to index each queried field; this solution works for table with up to dozen of thousands rows. For larger tables with millions of rows, performances decrease even if we try to split tables by chromosomes.

The binning scheme splits each chromosome into consecutive equal-size intervals, called ‘bins’. By changing the bin size we can obtain a hierarchical bin structure (see Figure 9)where each bin is enumerated.

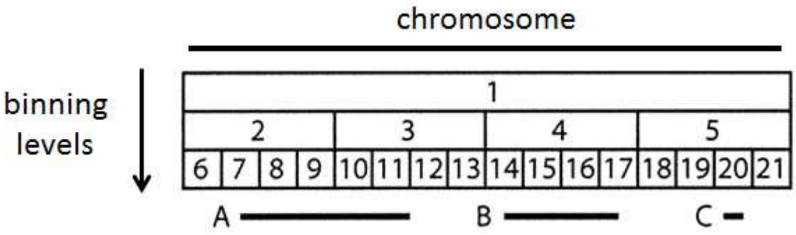


Figure 9.A simplified version of the binning scheme.

In the Genome Browser five different size of bin are used: 128Kb, 1Mb, 8Mb, 64Mb and 512Mb.

Each genomic feature (or annotation) reports the smallest genomic bin in which it fits. Looking at Figure 9, features A, B and C are associated to bin 1,4 and 20 respectively. When the browser needs to access features in a region, it must look in bins of all different sizes: to access all the features that overlap or are enclosed by range B, the browser looks into bins 1, 4, 14, 15, 16 and 17.

Because bins are pre-computed, it is possible to pre-calculate the smallest fitting bin, given the genomic coordinates of the feature to store in the database. When a range query is performed, all the possible bins of different size are pre-selected basing on the range query coordinates.

Therefore, the previous query becomes:

```
select * from Table where chrom=’chr1’ and
chromStart<20000 and chromEnd>10000 and (bin=1 or bin=2
or bin=10 or bin=74 or bin=586)
```

Even if the query appears more complex than before, it runs much faster thanks to the reduced searching space.

Binning scheme combined to B-tree index (by indexing the bin field within the database) finally provides a crude approximation to a R-tree [86] that, notably, is implemented into MySQL as an index scheme. Actually, when the UCSC Genome Browser has been firstly implemented, R-trees were not supported yet into the MySQL engine and later attempts to use the engine built-in R-trees failed [87].

2.4.2. Human Genomic Variant Resources

In the last three decades, high throughput single nucleotide polymorphism[3] (SNP) genotyping has produced a great amount of data in terms of genomic polymorphisms.

Efforts to catalogue these data at population level resulted into the International HapMap Consortium [88] that aimed to build linkage maps and identify chromosomal regions where genetic variants were shared [89].These variations have been the core around which genome wide association studies (GWAS) were built.

The advent of sequencing technologies, allowed exploring the wide range of human variations, including rare variants, too.

[3] A genomic variant for which one of the allele has a frequency greater than 5% in the reference population

Hereby are briefly described the main genomic variant resources based on NGS data. In addition to merely cataloguing human variation, these databases serve many purposes such as estimating linkage disequilibrium in a given population or reducing the number of variants used in association tests.

2.4.2.1. The Single Nucleotide Polymorphism database

The Single Nucleotide Polymorphism Database (dbSNP) is a free public resource of genomic variation developed by the National Center for Biotechnology Information (NCBI) and the National Human Genome Research Institute (NHGRI).

Published in 2001 [90], it pursues the challenging goal to catalogue every found nucleotide sequence variations through different experimental settings, including next generation sequencing. Despite its name that quotes only polymorphisms, in fact, there is no requirement or assumption about minimum allele frequencies or functional neutrality for the genomic variants in the database: it includes both human disease-causing clinical mutations and neutral polymorphisms as well. Moreover, genomic locus is cross-linked with other information resources such as GenBank, LocusLink, the human genome sequence and PubMed.

DbSNP collects genomic variations through submissions from public and private sources that have to follow a specific data format protocol (e.g. organism, population, observed alleles, 5’ and 3’ flanking sequences, gene name etc.). An accession number (ss#) is assigned to each submitted variation. A reference SNP (refSNP) cluster ID (rs#) will is also assigned to each unique variation in an organism reference genome.

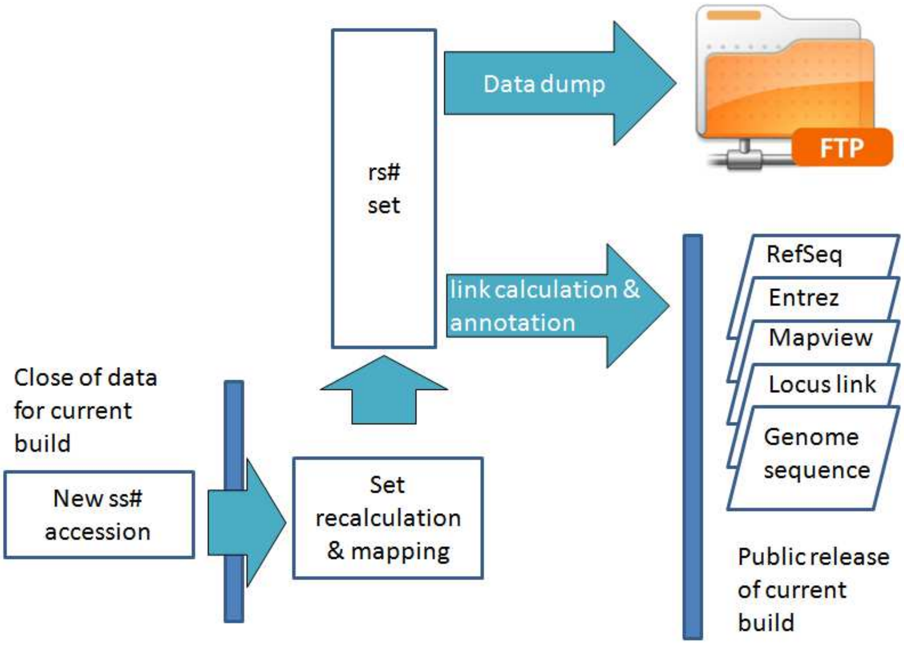


Figure 10. The dbSNP build cycle. Inspired by The NCBI Handbook, http://lgmb.fmrp.usp.br/cbab/NCBIHandBook/ch5d1.pdf

When a new version of dbSNP is going to be released, the dbSNP build cycle begins (see Figure 10).

Newly submitted variants (#ss) plus the whole set of the refSNPs are mapped into the reference genome sequence. Map data are used to merge (cluster) submissions into existing refSNP clusters or to create new ones. New refSNPs are annotated trough genomic resources (e.g. RefSeq and Entrez) and the release content is delivered in diverse formats on the dbSNP FTP site.

Working with dbSNP data, some issues should be taken into account.

Firstly, refSNP are based on the current genome assembly because of the mapping process and therefore are subjected to the updates of the same reference: each time there is a new version of the genomic assembly refSNPs must be update or reclustered and it is not rare that different refSNPs are clustered together and both assigned only one refSNP id (generally the one with lower number). As a consequence, refSNP is not a stable id over different dbSNP versions.

Secondly, a refSNP points to a genomic locus, but not to an univocal variant (respect to the genomic reference): e.g. rs1800001 is mapped to the chrX at the genomic position ‘18644526’ on the GRCh38 assembly; by exploring dbSNP (v.142) it is associated to three different alleles (G/C/A) and the reference allele is ‘A’; indeed, this refSNP holds two variants with respect to the reference: from A to G (A>G) and from A to C (A>C).

Thirdly, refSNPs can be reported on the plus or minus strand of the DNA and the strand orientation can be challenging to the researchers when it comes to consistency and comparing variant sets, especially when misleading the “forward” and “reverse” terms are used [91]. The Illumina TOP/BOT strand convention has been introduced, for example to solve G/C and A/T allele ambiguities [92].

2.4.2.2. The 1000 Genomes Project

Launched in 2008, the 1000 Genomes Project is an international research effort to characterize the human genome sequence variation with the aim to provide a foundation for investigating the relationship between genotype and phenotype [5].

Scheduled in three sequential phases, it has the goal to sequence and characterize more than 2500 “healthy” individuals belonging to 26 different ethnicity groups around the world. At the time of this writing, a first release of the phase 3 is available containing over 79 million variant sites from the whole genome sequencing of 2504 individuals.

The 1000 Genomes Project (1000GP) resources is downloadable via two the mirrored EBI and NCBI FTP sites and data can be directly viewed through the dedicated 1000 GP web browser.

Genomic variants for each individual, identified by the combination of several variant callers, are reported by the VCF format (see 2.2.1.1) and comprise SNVs, indels and structural variations (SVs) that, in this case, group deletions, insertions or copy number variants for genomic region encompassing generally more than 50 bases.

2.4.2.3. The Exome Sequencing Project

The NHLBI GO Exome Sequencing Project (ESP) is actually the biggest integrated resource of whole exome sequencing (WES) data comprising, in the last version (ES6500), genomic variations from 6503 samples belonging to African-Americans and European-Americans ethnicity groups.

Made up by the cooperation of several USA research institutions, it aims to discover and characterize novel genes and mechanisms contributing to heart, lung and blood disorders [93].

Genomic variants with data aggregates are free-accessible through the Exome Variant Server: data can be viewed by the integrated data browser or can be downloaded directly via HTTP. Individual variants and other related genomic data are publicly accessible only via dbGap, a controlled access resource for research purpose [69].

2.4.2.4. The Ensemble Variant Database

Ensembl is a joint project between EMBL-EBI and the Wellcome Trust Sanger Institute that aims to develop a software system able to produce and maintain automatic annotations on eukaryotic genomes. The Ensemble core database has been developed for the latter purpose.

The Ensembl Variant Database [72] is based on MySQL [94] engine and has been developed specifically to deal with genotyping and sequencing data (see Figure 11 for a simplified version of the database schema). Data are accessible through the Ensemble Application Program Interface (API) written in Perl [95] and allows to connect to the requested database and represent database entities as Perl objects.

Variation data, such as a SNV or DIV is defined by using its upstream and downstream flanking sequences ad at least one variant allele. Flanking sequences are aligned to one or more positions of the reference genome.*Variation*, *flanking_sequences* and *allele* tables (see Figure 11) represent a variant independently from the genome assembly, while the *variation_feature* holds reference genome variant mapping information. This data division has been made in order to update only *variation_feature* table in case of genome assembly update.

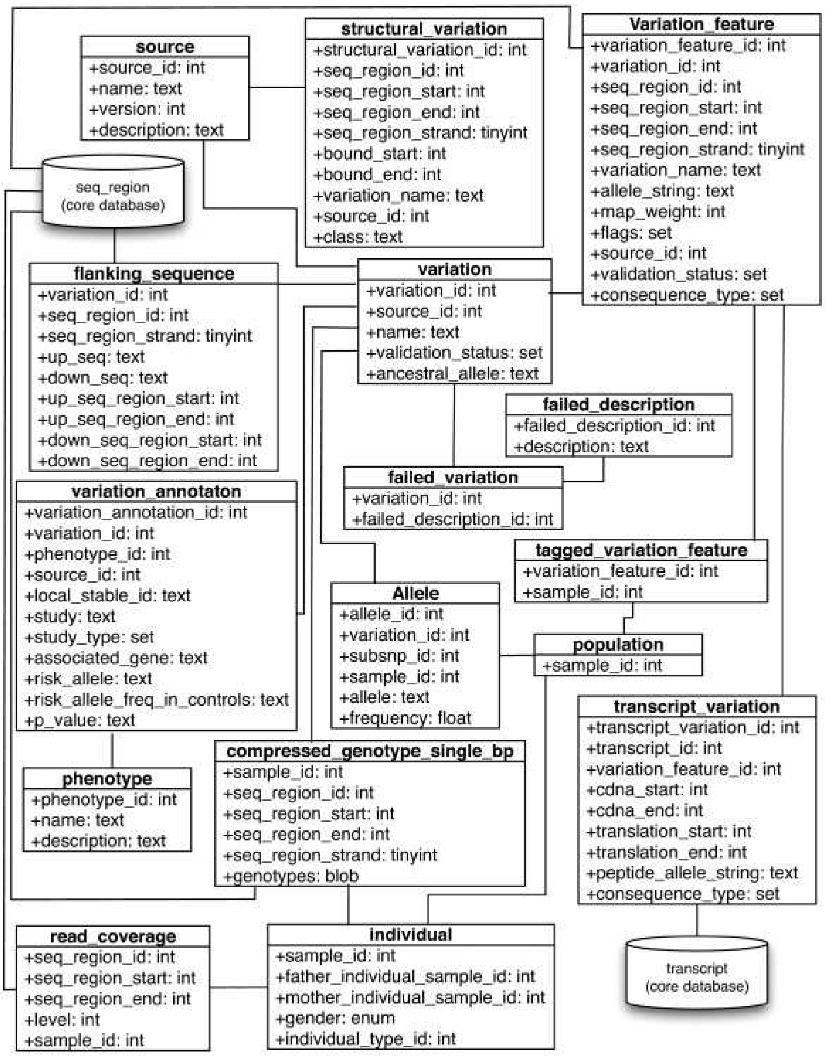


Figure 11.A simplified schema of the Ensemble Variant Database. Variants are stored into the Variation Table in the top center of the figure. Connections to the Ensembl core database are shown.

Ensemble variant database has been initially developed with the aim to collect genotyping data from different resources such as dbSNP and HapMap and therefore it has several annotation data in common with such databases (e.g. flanking sequences) deriving from the experimental methods.
Sequencing data are represented as variations, along with some sequence read information (*read_coverage* table) regarding alignment position of reads, coverage levels and differences between alignments and the genomic reference.

2.4.3. Variant Annotation Tools

Sequencing pipelines regarding WES or WGS applications end up with a plethora of differences between the sequenced genome and the genomic reference used to map sequencing reads.
In 2.2.1.1the standard format (VCF) used to represent genomic variants has been introduced. Despite VCF has been designed to group every information regarding variants, it is limited to the sequencing context and it reports data such as reads coverage, quality and flanking sequences. VCF it has not been designed and standardized to hold additional data regarding knowledge on functional genomic region to which a variant could overlap or the variant allele frequency in a specific public resource (such as dbSNP).

In order to deal with the integration of different genomic data sources and enrich variants with the related content, several variant annotation software have been developed in the last years[96-100]. Note that the Ensembl Variant Database supports annotation as well, being linked to the Ensembl core database containing genomic annotations. However, in the last years the trend was to separate genomic annotation at sequence level from variant annotation. The first relies on genomic databases such as UCSC, RefSeq and Ensembl. The second uses data from the genomic databases, but previously manipulated (a priori or on the fly) in such a way to guarantee a fast data retrieval in the variant annotation process. Data pre-processing became the key word and this concept will be remarked in Chapter 3.

Hereby two public open-source and most common used variant annotation tools are showed in detail. The first (VEP) is based on a genomic structured database while the second (ANNOVAR) relies on indexed pre-computed files of genomic data tracks.

2.4.3.1. Variant Effect Predictor (VEP)

The Variant Effect Predictor (VEP) [96] has been built on the Ensembl Variant Database. It consists of an API extension of the aforementioned Ensembl API. Written in Perl, it allows matching genomic annotation from the Ensembl Variant Database with a given list of variants. Variants are represented by their genomic coordinates and alleles, thus allowing the API to query transcript related data (table *transcript_variation* in Figure 11) and to match the variants with their possible overlapping transcripts. The latter step permits determining if a variant falls within an exon: in this case, a new codon is derived for the variant allele. Moreover, the API assesses whether a variant falls into a splice-site, intronic, regulatory, untranslated genomic region.

VEP can be run in a standalone fashion; however its dependency on Ensembl database limits its portability.

2.4.3.2. Annotate Variation Software (ANNOVAR)

ANNOVAR (ANNOtate VARiation) is a standalone software tool written in Perl with portability as a main goal. In fact, it is not based on structured database but it simply consists of two components: i) several Perl scripts constituting the “business logic” and ii) indexed plain text files with the pre-computed genomic annotations.

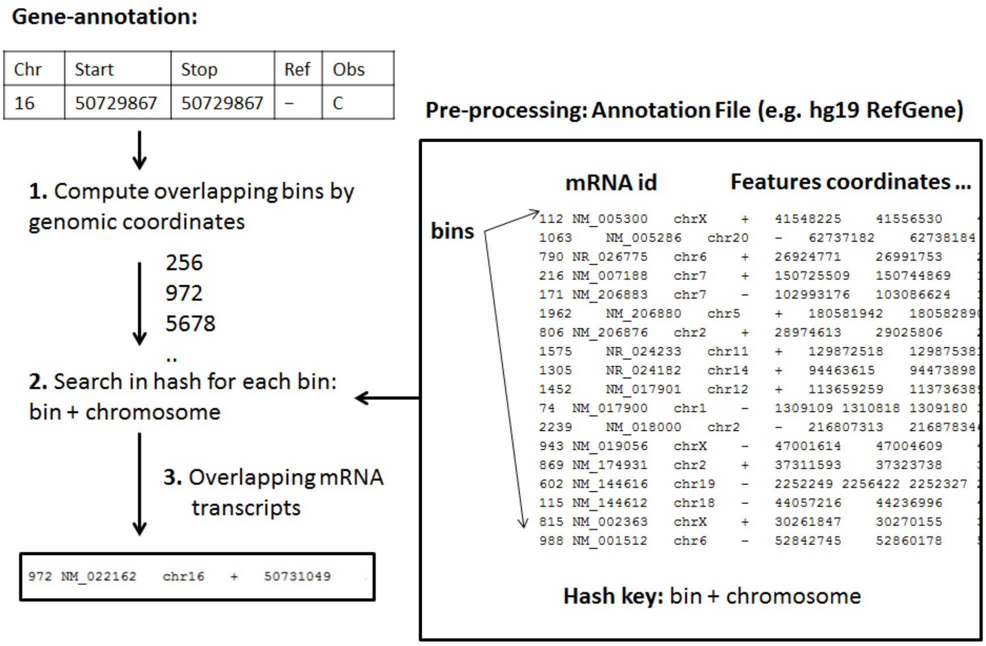
ANNOVAR can be accessed only by command line, making it desirable for programmatic usability.

The workflow consists of two steps: preparation of the input file and variant annotation by genes, regions or other variants.

Variants are represented by genomic coordinates, reference and variant allele. ANNOVAR provides accessory scripts able to convert different variant format (such as VCF) to its predefined one and supports both SNVs and DIVs.

Gene-annotation starts from converted variants and scan annotated mRNA sequences in order to match a variant within an exon, splice site, intron, untranslated region or outside a gene. The search proceeds by genomic interval, in a similar way to UCSC Genome Browser as shown in 2.4.1.1: genomic bin are pre-computed for each mRNA transcript and are stored in the file as a column field. The mRNA file contains genomic coordinates for each transcript, such as exon intervals and relative reading frames.

In the gene-annotation step, the mRNA file (12MB) is loaded in RAM. When a variant has to be annotated, the overlapping bin identifiers for the corresponding chromosome are computed starting from the variant genomic coordinates. The in-memory structure of the mRNA file in the form of key-value (hash) is queried by keys that are chromosome-bins (Figure 12).


Gene-annotation:
1. Compute overlapping bins by genomic coordinates
256
972
5678
..
2. Search in hash for each bin: bin + chromosome
3. Overlapping mRNA transcripts
Pre-processing: Annotation File (e.g. hg19 RefGene)
bins
mRNA id
Features coordinates ...
Hash key: bin + chromosome

Chr	Start	Stop	Ref	Obs
16	50729867	50729867	–	C

```
972 NM_022162    chr16    +    50731049
```

```
112 NM_005300   chrX    +   41548225    41556530
1063    NM_005286   chr20   -   62737182    62738184
790 NR_026775   chr6    +   26924771    26991753
216 NM_007188   chr7    +   150725509   150744869
171 NM_206883   chr7    -   102993176   103086624
1962    NM_206880   chr5    +   180581942   18058289
806 NM_206876   chr2    +   28974613    29025806
1575    NR_024233   chr11   +   129872518   12987538
1305    NR_024182   chr14   +   94463615    94473898
1452    NM_017901   chr12   +   113659259   11373638
74  NM_017900   chr1    -   1309109 1310818 1309180
2239    NM_018000   chr2    -   216807313   21687834
943 NM_019056   chrX    -   47001614    47004609
869 NM_174931   chr2    +   37311593    37323738
602 NM_144616   chr19   -   2252249 2256422 2252327
115 NM_144612   chr18   -   44057216    44236996
815 NM_002363   chrX    +   30261847    30270155
988 NM_001512   chr6    -   52842745    52860178
```

Figure 12.ANNOVAR gene-annotation workflow

Once the mRNA of the corresponding variant has been identified, mRNA features coordinates are checked in order to assign the variant into the coding, intron, splicing or untranslated gene regions.

In case of exonic variants, the annotated mRNA sequences are scanned in order to report the amino acid change given by the variant as well as stop-gain or stop-loss mutations.

In such similar way, ANNOVAR annotate variants against several variant pre-computed resources such as 1000GP and ESP.

2.5. Genomic Data Interpretation in NGS

Human DNA sequencing allows to identify a number of genomic variants that can vary from dozen to up 5 million per sample depending on the sequencing target (several genes, whole exome or the whole genome).

The majority of these variants is frequent in the population because it consists in the natural genetic variation that has been cumulated during centuries, modulated by natural selection and therefore contributed to evolution. These variants are typically called “polymorphisms”. Another part of these variants results to be more rare among a population or private for an individual. The variants belonging to this part are often called “mutations”. Mutations can be distinguished between those that give a selective advantage and therefore will help the carrier organism to survive and will be transmitted to the progenies (becoming therefore common variants in future generations) and those mutations that hinder survival and are under negative selection thus tend to be eliminated from the population.

Polymorphisms consist therefore in the genetic background of a population. A single polymorphism can explain only in part a given human phenotype such as a common (or complex) disease or trait.

Genome wide association studies (GWAS) has been developed in the last ten years in order to exploit common diseases through the genotyping of millions of polymorphisms along the human genome. Genotyping is different from sequencing and it briefly consists into “reading” only a predefined set of genomic regions that are randomly distributed along the human genome.

GWAS have identified genetic risk factors for common diseases such as type II diabetes, schizophrenia and many others including pharmacogenetics traits[101, 102]. However, it has to be noted that, for the largest part, these identified genetic loci collectively accounted for only a small fraction of the observed heritability of the investigated traits [103].

Complex diseases are the results of a combination of genetic and environmental factors and each can contribute to the susceptibility to the phenotype. Polymorphisms are rarely directly associated to the disease, rather they can be a sort of flag indicating the presence of the causal variants through linkage disequilibrium (the non-random allele association of two or more genomic loci given the biased DNA recombination) or synthetic association (see Figure 13).

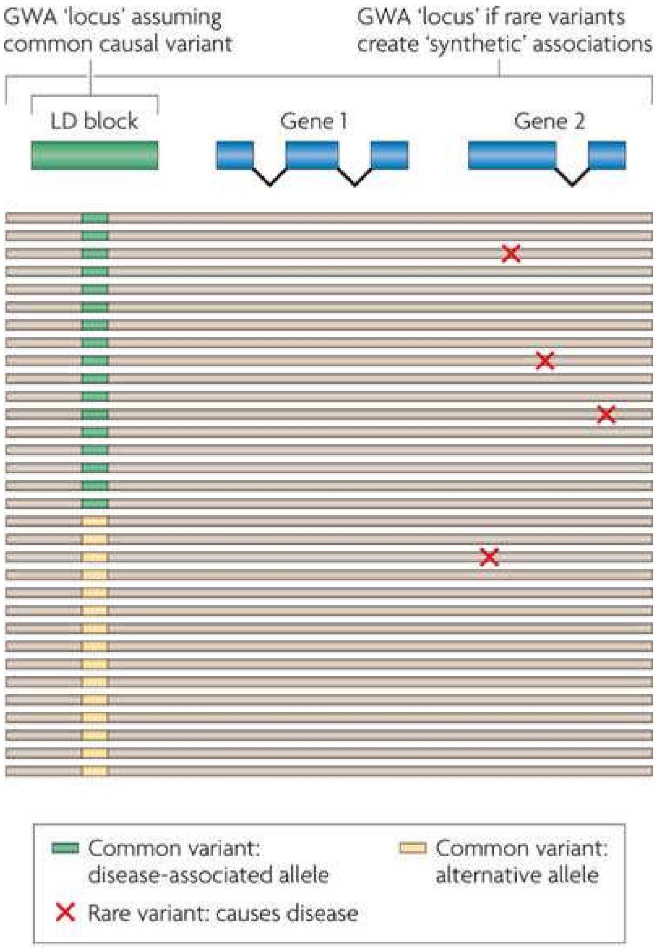

GWA 'locus' assuming common causal variant
GWA 'locus' if rare variants create 'synthetic' associations
LD block
Gene 1
Gene 2
Common variant: disease-associated allele
Common variant: alternative allele
Rare variant: causes disease

Figure 13. Each rare variant shown cause disease and they occur commonly on the haplotype containing a certain allele of the polymorphism. The signal credited to the common variant is weaker than the real effects of the causal variants. In the case shown, moreover, the causal variants do not lie in the Linkage Disequilibrium block of the common variant [104].

In order to identify risk loci for a given trait, GWAS typically need hundreds of genotyped individuals sharing the same phenotype in order to hold the statistical power that leads to significant results in terms of genotype-phenotype association. However sample size for these kind of genetic studies depend upon the expected allele frequency in the population and the expected risk incurred by that allele.

Common Disease Common Variants hypothesis (CDCV) states that common disorders are influenced by common variants in the population with low or moderate penetrance, that is a polymorphism contributes (together with environmental or other genetic factors) to the risk by a small amount, thus the prevalence of the disease and the allele frequency are slightly correlated. Also, in case of complex diseases showing heritability, a polymorphism with a low penetrance must be spread across multiple genetic factors.

The Rare Disease Rare Variants hypothesis (RDRV) instead, applies for those diseases that are typically rare, monogenic, and heritable in Mendelian fashion and caused by rare variants at high penetrance and effect size, thus the allele frequency and the disease prevalence are in high or perfect correlation.

Figure 14 shows the relationship between disease-related variants effect size and their frequency among a population: while GWAS applies on the lower right, sequencing covers a broader range and is suitable to detect both rare variants causing Mendelian diseases and more common variants with moderate effect size.

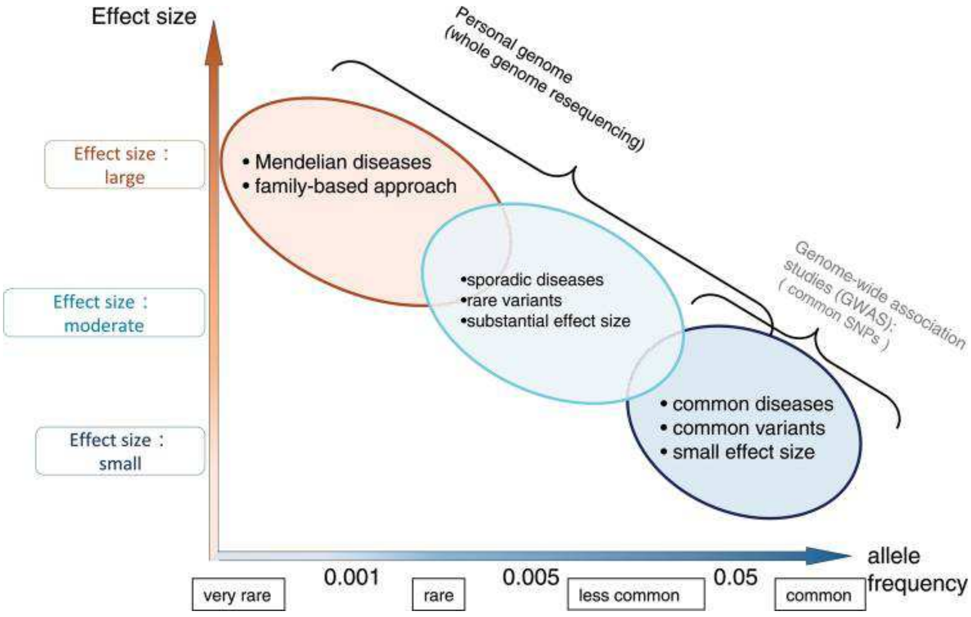


Figure 14. Disease-related variants on the basis of the effect sizes and allele frequencies in the population [105].

Whole genome sequencing is the most comprehensive study to exploit the role of rare and common variants in disease; however, until sequencing costs will be more affordable in order to sequence thousands of samples with a given phenotype and perform whole genome association studies, alternative approaches such as family-based and extreme-trait [104] experiment designs have to be taken into consideration in order to identify genetic causes at the base of complex diseases.

Family-based strategy consists to sequence affected individuals which belong to the same family, possibly the most distantly related ones in order to limit the number of shared and possibly causative variants.

The extreme-trait strategy consists to sequence a relative small set of affected individuals with extreme phenotype traits over the phenotype distribution. In this way, the variants that contribute to the trait will be enriched in frequency in such a population subset. This strategy, however, supposes to have detailed phenotype data for a broad range of individuals in order to generate this extreme sample selection.

Both rare and common disease sample sequencing applications needan a priori weight for the identified variants, in order to further distinguish between putative functional variants that can be related to the disease and those variants that constitute natural genetic background or that can be related to other secondary traits.

Especially in case of whole exome sequencing where the major part of variants belong to the protein-coding region of the genes, variant weighting can be performed by applying several steps [106] (see Figure 15):

- Apply a priori knowledge on gene mapping respect to the reference genome
- Perform discrete-filtering step based on allele frequency among a reference population or control samples (under the assumption that the control set contains no alleles from individuals with the phenotype being studied)
- Rank and prioritize variants based on the involved coding region, qualitative changes to the primary protein sequence, conservation of modified DNA bases (or amino acids in case of protein-coding variants),normal gene variability in order to avoid those genes with highest and noisy mutational rate, protein functional domains and variant matching against well phenotype-associated variants in literature.

Once obtained the reduced variant candidate list, experimental evidence of pathogenicity by functional studies has typically to be assessed.

Gene expression by RT-PCR, in vitro splicing assays or animal models phenotype replication are common examples of such functional studies. Family co-segregation is another important feature for variant interpretation despite there exist cases (e.g. in case of incomplete penetrance) for which establishing the inheritance pattern is far from be straightforward [106].

A common practice requires also to confirm identified candidate mutations with higher accurate experimental methods such as Sanger sequencing.

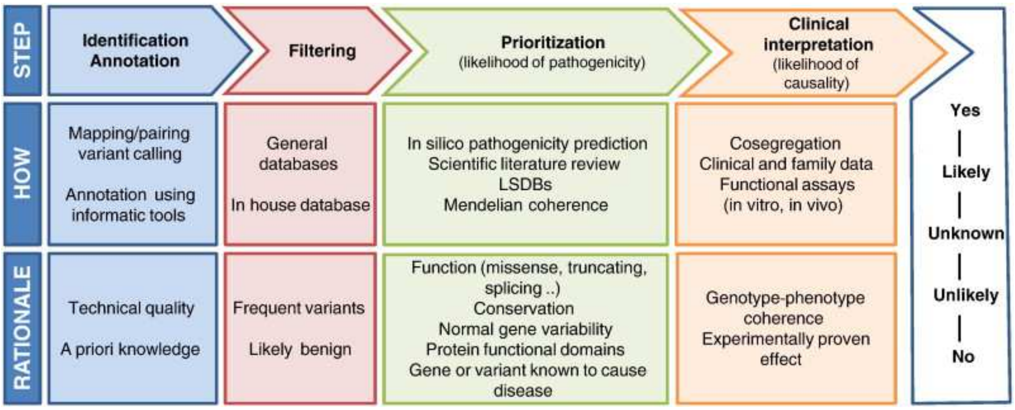

STEP
Identification Annotation
Filtering
Prioritization (likelihood of pathogenicity)
Clinical interpretation (likelihood of causality)
HOW
Mapping/pairing variant calling
Annotation using informatic tools
General databases
In house database
In silico pathogenicity prediction
Scientific literature review
LSDBs
Mendelian coherence
Cosegregation
Clinical and family data
Functional assays (in vitro, in vivo)
RATIONALE
Technical quality
A priori knowledge
Frequent variants
Likely benign
Function (missense, truncating, splicing ..)
Conservation
Normal gene variability
Protein functional domains
Gene or variant known to cause disease
Genotype-phenotype coherence
Experimentally proven effect
Yes
Likely
Unknown
Unlikely
No

Figure 15. Stepwise interpretation for genetic variants[106].

2.5.1. In-silico Prioritization of Genomic Variants

Identifying diseases related variants could be relatively easy in case each of the following rules holds:

- overlapping within a known and well mapped protein-coding region
- rare or unseen in a reference population or control groups
- causing an important and easily interpretable change to encoded protein such as the introduction of a premature stop codon (nonsense) or a translational reading frame shift (frameshift indels)
- related genes have a known function and have been well studied or, better, have been previously related to the same disease

Unfortunately this ideal pattern does not always hold for rare Mendelian diseases and (even more rarely) for complex ones where more common and mild-effect variants can concur to the phenotype as previously discussed.

Additionally, within whole exome or genome sequencing data, there is the need to distinguish between disease-causing or associated variants and the overwhelming amount of potentially functional variants present in any individual genome, but not pathogenic for the disease under study.
MacArthur *et al.*[107]considering only loss of function (LoF) variants (nonsense, frameshift indels, large deletion removing the first gene exon or more than 50% of the protein-coding sequence transcript) estimated that a healthy individual with European ancestry carries ~100 LoF variants with the 20% in an homozygous state. This list was even longer (about double)

previous to filtering variants by several criteria such as annotation errors or frameshift indels close to 5’ end of genes not highly conserved or not known to be functionally important and therefore resulting in likely tolerated truncating variants.

Other insights came up by analyzing properties of the relative genes mapped to this variant list. In fact LoF variants resulted significantly enriched for:

- less evolutionarily conserved genes comprehensive of their promoter region
- genes having more closely related paralogs (gene family members) with a greater sequence identity than other genes
- genes showing lower connectivity in protein-protein interaction and gene interaction networks

while they resulted significantly depleted for genes implicated in protein-binding, transcriptional regulation and anatomical development.

Thus, it is straightforward to integrate both variant-level and gene-level prior knowledge in order to carefully and correctly assess sequence variants in human diseases and to void false positive assessment of pathogenicity that could contaminate published results, and therefore impede the translation of genomic research findings into the clinical diagnostic setting[108].

Informatics evidence for variant pathogenicity assessment can be done at gene level, variant level or both.

Prioritization at gene level leverages existing knowledge on genes, proteins, diseases and phenotypes.

Given a candidate gene list and a disease or phenotype traits of interests the goal is to end up with the ranked gene list according to the phenotype/disease under study.

We can distinguish between two main kinds of gene-prioritization algorithms: those that exclusively use bio-ontologies to mine and report known and new associations between genes and diseases; and those based on “data fusion”, that consist in integrate disparate and heterogeneous data sources and match similarities between a given (training) gene list and a candidate (test)one.

Ontologies are knowledge databases in which the information is represented as a graph: terms are the nodes and respect a precise taxonomy, while edge are relationships between terms. The Gene Ontology (GO)

[109], Disease Ontology (DO)[110] and the Human Phenotype Ontology (HPO) [111]are examples of knowledge databases used in gene prioritization.

Recently, several tools that exploit one or more of these ontologies to prioritize variants have been developed [112-115].

Phevor starts from HPO terms describing the individual traits and comes up with a phenotype-linked gene list that further crosses with other ontologies such as GO and DO. In the ontology search process to each gene is assigned a score depending on relationships with the phenotype, GO terms and diseases: generally, more often a gene or its paralog comes up within the search, the higher it scores.

Other methods [113] makes use of semantic similarity between HPO terms annotated to genes and those used to describe an individual. Similarity between two terms is a function of their specificity and semantic relation: specificity tends to penalize those HPO terms with an high gene connectivity while semantic relation consider the similarity of two terms by the specificity of their most informative common ancestor.

Endeavour [116] and ToppGene [117] are examples of data fusion applications to gene prioritization. Functional annotation (by GO), gene expression, sequence similarity, transcriptional motifs, protein domain, gene network and, in case of ToppGene, mouse phenotype data are integrated to rank a list of candidate (test) genes on the base of similarities with given training genes.

As shown in Figure 16, steps for gene prioritization proceed in a similar way: a training gene set is used to gather information about diseases, pathways and the other kinds of data; a test gene set (corresponding to the candidate gene list) follows the same searching process and for each data record (corresponding to a data source type) genes are ranked accordingly to the data similarity with the training gene list; finally data fusion here consist to merge ranks obtained from the separate data sources into a single ranking. The different tools differ on statistical approaches used to compare similarities between training and test attributes/values (ToppGene uses a combination of Fuzzy Measure and Pearson Correlation, while Endeavour uses Fisher’s omnibus analysis and Pearson Correlation as well),plus data sources used.

However, both ontologies and data fusion based gene prioritization methods suppose that phenotypes and/or disease of the individuals under study have been well defined or that a list of known genes related to the trait of interest exists. If ontologies based methods can help with the second issue ending up with a list of gene similarities on functional annotation (such as Phevor does) the choice of using phenotype terms can be controversial [113]. Moreover, these methods do not care about genomic variants that have led to the candidate gene list building.

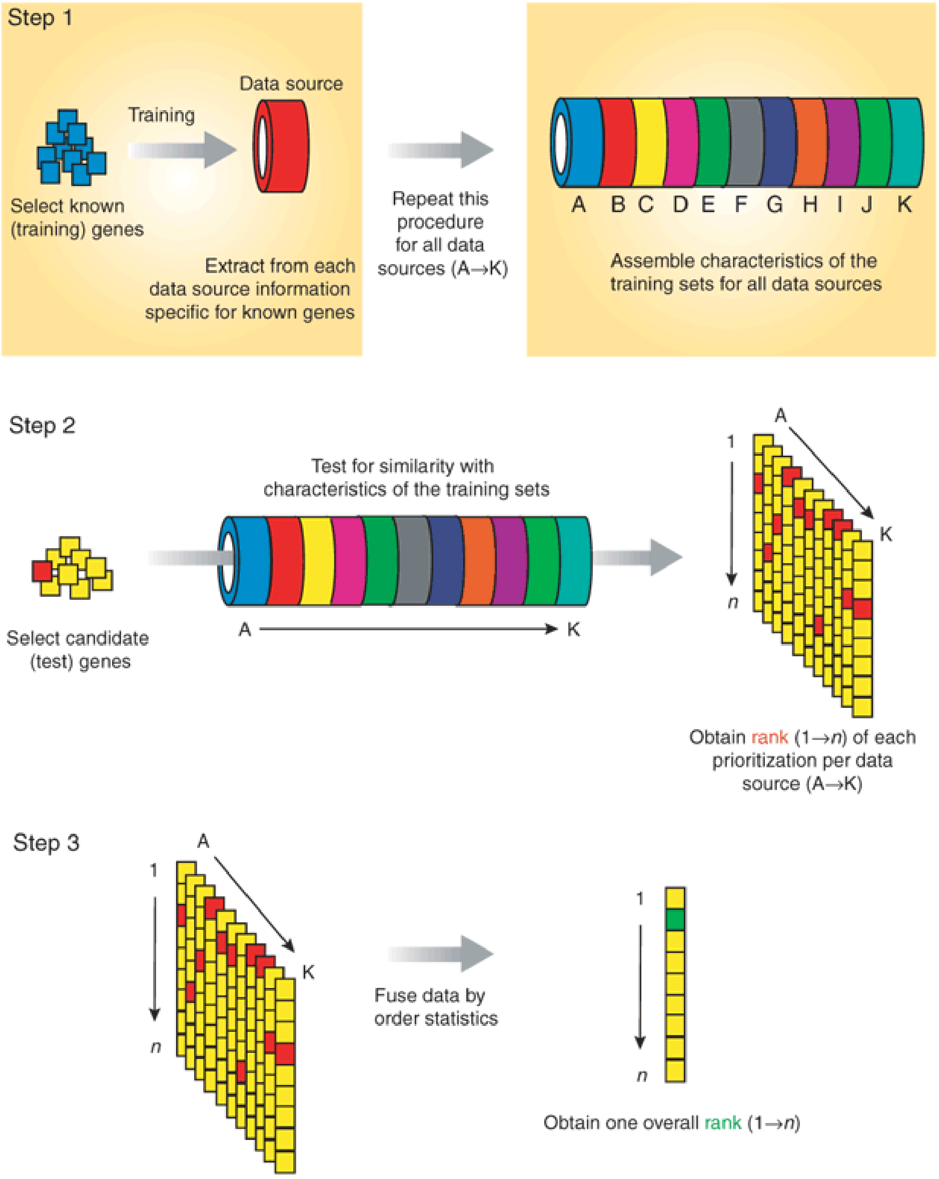

Step 1
Data source
Training
Select known (training) genes
Extract from each data source information specific for known genes
Repeat this procedure for all data sources (A→K)
A B C D E F G H I J K
Assemble characteristics of the training sets for all data sources
Step 2
Test for similarity with characteristics of the training sets
Select candidate (test) genes
A
K
1
n
Obtain rank (1→n) of each prioritization per data source (A→K)
Step 3
Fuse data by order statistics
Obtain one overall rank (1→n)

Figure 16. Typical steps in gene prioritization by data fusion [116]

Variant prioritization consists in assessing the variant pathogenicity by using specific attributes at nucleotide and protein sequence level where a genomic variant occurs.

Recently, methods that combine gene and variant prioritization level approaches has been developed [118].Moreover, since the availability of publicly genome datasets increases with the number of sequenced

individuals, statistical frameworks to prioritize genes basing on their observed/expected mutational rate have shown their utility especially in the identification of rare inherited disease genes [119, 120].
In this thesis we further focus on variant prioritization methods, also called "variant prediction" algorithms.

2.5.2. Variant Prediction Algorithms

Human genomic variants can be first classified on the basis of genomic region to which they overlap.
Coding variants may cause changes to the primary structure of the peptide encoded by the relative gene, therefore are the most amenable to functional interpretation and, as a consequence, are the most studied.
Non-coding variants constitute the major part of the individual human genomic variation [5] and are likely to contribute with low effect size to complex traits as confirmed by GWAS. Moreover, several large-effect regulatory variants (e.g. in promoter and enhancer regions) have been confirmed to be the cause of several Mendelian diseases.

Predicting the deleteriousness of coding variants is associated to predicting whether they alter protein stability, structure and/or protein function. We can distinguish between two main coding variant predictor classes: structure-based and sequence-based methods [121].
Structure-based methods [122, 123] rely on energy function-based approaches and require protein three-dimensional structure as input to end up with accurate results. Because of their high computational demand, their application to the big amount of sequencing data is unfeasible.
Sequence-based methods typically use sequence homology, sequence evolutionary conservation, structural information(e.g. surface accessibility, hydrogen bonding). They can be further divided into those adopting a "first-principle" approach and trained classifiers [76]. First-principle approaches make predictions basing on a defined biological property (e.g. evolutionary conservation) while trained classifiers are based on heuristic associations of many potentially relevant attributes that significantly can discriminate between true positive and negative instances. Trained classifiers methods are generally more accurate but can be biased by the training data; however, they have the advantage to be tunable. First principle approaches are more interpretable, but are limited to their assumption and do not model all the possibly relevant factors.

A gold standard algorithm in variant prediction does not exist yet and the scientific community is in agreement that the combination of the diverse approaches is actually the most accurate solution [124], despite the optimal ensemble model has not been univocally identified.

Many variant prediction methods basing on homology sequence alignment [8, 125-127], sequence conservation [124, 128-131] and protein structural/functional parameters [2, 3, 132, 133] have been developed.
Four algorithms, encompassing aforementioned categories are hereby discussed in more detail, selected for their strategic role in this thesis.

2.5.2.1. PolyPhen-2

PolyPhen-2 is an algorithm that aims to predict weather a single nucleotide variants leading to an amino acid substitution (missense or non-synonymous variant) can affect or not the encoded protein functionality.

It is based on the multiple sequence alignment (MSA) paradigm and makes use of protein 3D structure attributes too.

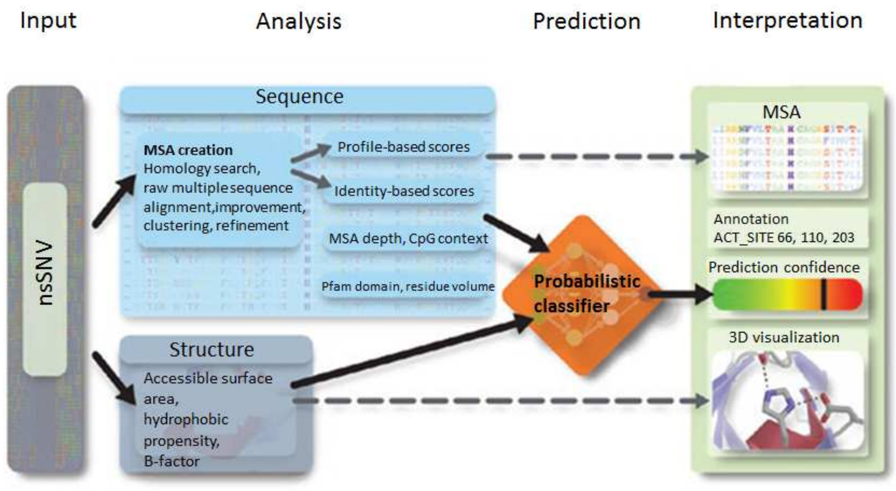


Figure 17. PolyPhen-2 workflow[2]

Given a single nucleotide variant (SNV), it is mapped onto available mapped mRNA transcripts: PolyPhen2 makes use of UCSC mRNA transcripts tracks by UCSC Table browser for variant mapping.
Depending on the reading frame of the mRNA transcript, the nucleotide sequence flanking the SNV (25 base pair for each snippet) [134] is translated into amino acids.
The amino acid sequences are search via BLAST+ [135] in a database of sequence protein (the UniProt UniRef100 and SwissProt) in order to match homologous sequences(both orthologs and paralogs with an identity match between 10-94%) and are aligned by MAFFT [136], a multiple sequence aligner software. The obtained MSA is then improved by alignment

refining (LEON [137])and clusters of similar sequences are identified by Secator algorithm [138]in order to distinguish between subfamilies. Protein subfamilies are frequently representative of sets of protein with related functions and/or domain organization and therefore only the compact cluster, which includes the analyzed sequence, is further processed.

Position-specific independent counts (PSIC) software [139] is used to assign weights to the amino acids respect to the MSA and obtain the so-called profile matrix, which elements represent the logarithmic ratio of the likelihood of a given amino acid occurring at a particular site to the likelihood of this amino acid occurring at any site, computed by using prior probabilities from the amino acid substitution matrix BLOSUM62 [140]. PSIC for the wild, mutated amino acid and their PSIC difference are computed.

At the protein and nucleotide sequence level, PolyPhen-2 considers also: whether the nucleotide variant overlaps to CpG islands and is a transition (A<->G, C<->T) or a transversion (A<->C, G<->T,A<->T,C<->G); whether the variant is inside a Pfam [141]domain.

PolyPhen2 takes into account protein 3D structure parameters related to the amino acid change. It first maps with BLAST the analyzed sequence to a database of protein structure, in particular PDB [142], considering at least sequences with 50% identity. Then, it obtains structural parameters by mapping the amino acid residue of the PDB record into DSSP database [143] and calculates many parameters, including the normalized accessible surface area of amino acid residue, the change in accessible surface area propensity for buried residues, the change in residue side chain volume and B-factors (a measure of the local mobility resulting from crystallography).

A complete list of PolyPhen-2 features is reported in Table 3.

Features	Type	Values
PSIC score for the wild type amino acid	sequence	(-1,1)
PSIC difference between wild and mutated	sequence	(-3.27,4.57)
Number of residues observed at the position of the MSA	sequence	(1,432)
Congruency of the mutant allele to the multiple alignment	sequence	(0,95.5)
Sequence identity with the closest homologue deviating from wild type	sequence	(1.56,95.5)
Pfam domain hit	sequence	Yes,No
Variant transition/transversion	sequence	No,

in CpG island		Transition, Transversion
Change in residue side chain volume	structure	(-167,167)
Normalized accessible surface area of amino acid residue	structure	(0,1.55)
Crystallographic beta-factor	structure	(-1.85,5.17)
Change in accessible surface area propensity for buried residues	structure	(-1.83,2.89)

Table 3. PolyPhen-2 features and relative range values.

Sequence and structure related features are used to train a Naïve Bayes classifier coupled with entropy-based discretization[144], chosen because of the heterogeneous feature set (discrete and continuous values as shown in Table 3) and the presence of missing values (e.g. when PDB structure is not available or lack of homologous sequences).

Being the Naïve Bayes classifier a supervised approach, it requires data for training and for testing.

PolyPhen-2 has been trained on UniProt database for the positive class while negative variants (that is supposed neutral) were compiled from differences in homologous protein sequences of closely related mammalian species. The Naïve Bayes classifier has been trained and tested by 5-fold-cross-validation consisting in split the dataset in five parts, four for training and one for test, repeating it 5 times with different parts used for test.

Actually, two versions of PolyPhen-2 exists, based on the learnt classification model on two different filtered data set: HumDiv, with 3155 Mendelian disease related variants extracted from UniProt database as positive instances and 6321 differences between human proteins and their closely related mammalian homologs that were considered neutral (negative instances); HumVar, with 13032 human disease causing variants (comprehensive of Mendelian diseases but not only) and 8946 human missense variants without annotated disease data.

Results on test set showed that for a false positive rate of 20%, PolyPhen-2 achieved true positive prediction rates of 92% for the HumDiv and 73% for the HumVar dataset (see Figure 18).

A reason for the lower accuracy on HumVar is that the relative variant database may contain mildly deleterious alleles that have been classified as non-damaging. Therefore, it is recommended to use HumVar to predict variants in Mendelian diseases in order to clearly separate high from low or null effect size variants.

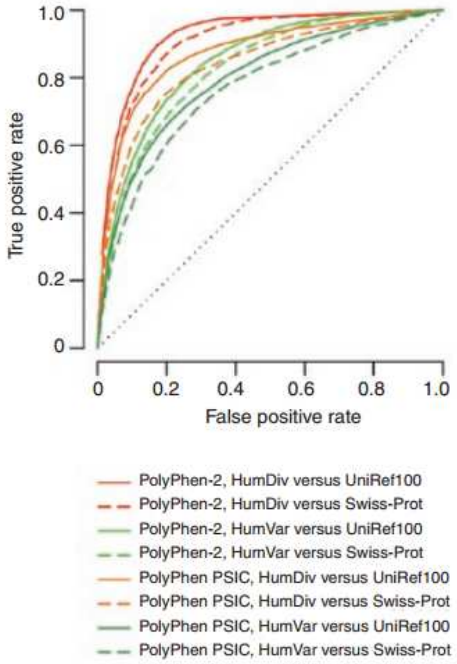


Figure 18.ROC curves of PolyPhen-2 on HumVar and HumDiv datasets. Comparison with the first version of PolyPhen is shown as well[2].

2.5.2.2. SIFT

Sorting Tolerant From Intolerant (SIFT) is a multi-step algorithm based on protein homology sequence similarities like PolyPhen-2 but it holds critical differences on how it retrieves homologous sequences and calculates the weights associated to amino acid substitutions in the resulting MSA, moreover it does not make use of structural properties and uses empirical cutoff for classification rather than learning.

Given a sequence query representing the mutated protein, SIFT searches a protein database (SwissProt) using PSI-BLAST and selects similar sequences iteratively until conservation in the conserved regions decreases. PSI-BLAST, in fact, performs the multiple sequence alignment and SIFT clusters aligned regions in case of sequence identity greater than 90%. Then, a consensus sequence is made for each group by choosing the most frequent residue for each position. The MOTIF algorithm [145] is used to search conserved regions which are then grouped together if they are >90% identical and a consensus sequence is made for each conserved group. Conserved regions of the query sequence and sequences with >90% identity constitutes the “seed” to which additional sequences will be added. The seed is given again to PSI-BLAST to search among the consensus sequences that were excluded from the seed. The best hit is added to the

MSA and conservation for each conserved sequences position is computed by the following formula:

$$R_c = \log_2 20 - \sum_{a=1}^{20} p_{ca} \log p_{ca}$$

wherep_{ca} is the frequency at which amino acid a appears in position c. The total conservation is calculated as $\sum_C R_c$. If this conservation score is greater than or equal to the conservation of the seed, the best hit is added to the MSA and the seed is rebuilt. This step repeats until the conservation score does not decrease.

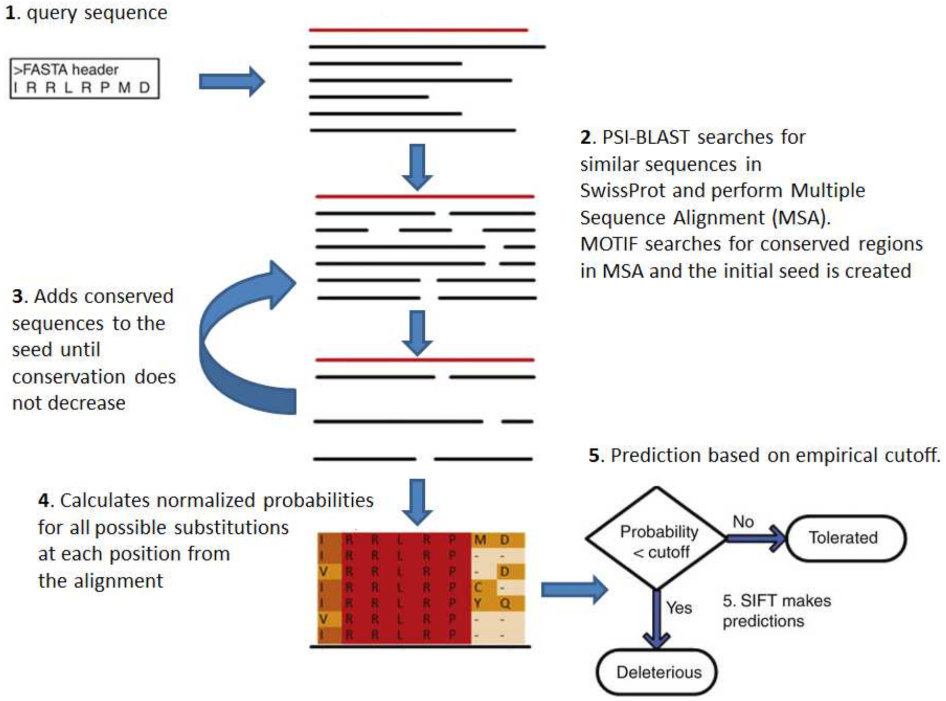

1. query sequence
>FASTA header
I R R L R P M D
2. PSI-BLAST searches for similar sequences in SwissProt and perform Multiple Sequence Alignment (MSA).
MOTIF searches for conserved regions in MSA and the initial seed is created
3. Adds conserved sequences to the seed until conservation does not decrease
4. Calculates normalized probabilities for all possible substitutions at each position from the alignment
5. Prediction based on empirical cutoff.
Probability < cutoff
No
Tolerated
Yes
Deleterious
5. SIFT makes predictions

Figure 19. Sift workflow. Image adapted from [146].

After the selection of the most conserved sequences into the MSA, the alignment is converted into a position-specific matrix (PSSM)[147], a Lx20 matrix where L is the length of the protein sequence. Each element of the matrix p_{ca} is the probability of amino acid a at position c of the protein.$p_{ca}$is a function of the residue frequency into the MSA in that position and the *pseudo-counts* [148]for the same residue, a method used to correct profile scores taking into account the fact that the observed sequences are an incomplete sample of the full set of related sequences.

Finally, each p_{ca} is normalized with respect to the max(p_{ca}) for that position and an empirical cutoff of 0.05 has been chosen to discriminate between damaging (<0.05) and tolerated (>0.05) depending on the observed amino acid substitution.

2.5.2.3. MutationTaster

MutationTaster is a variant prediction algorithm able to score both human protein-coding and non-coding variants.
Basing on different genomic variant databases (OMIM, HGMD, ClinVar, HapMap, 1000GP) and using results from different algorithms, it collects features on nucleotide/protein sequence conservation, protein structural properties, splicing sites, polyadenylation signals, regulatory regions and Kozak consensus sequences to train a Naïve Bayes classifier on known datasets of damaging and tolerated variants of different types.

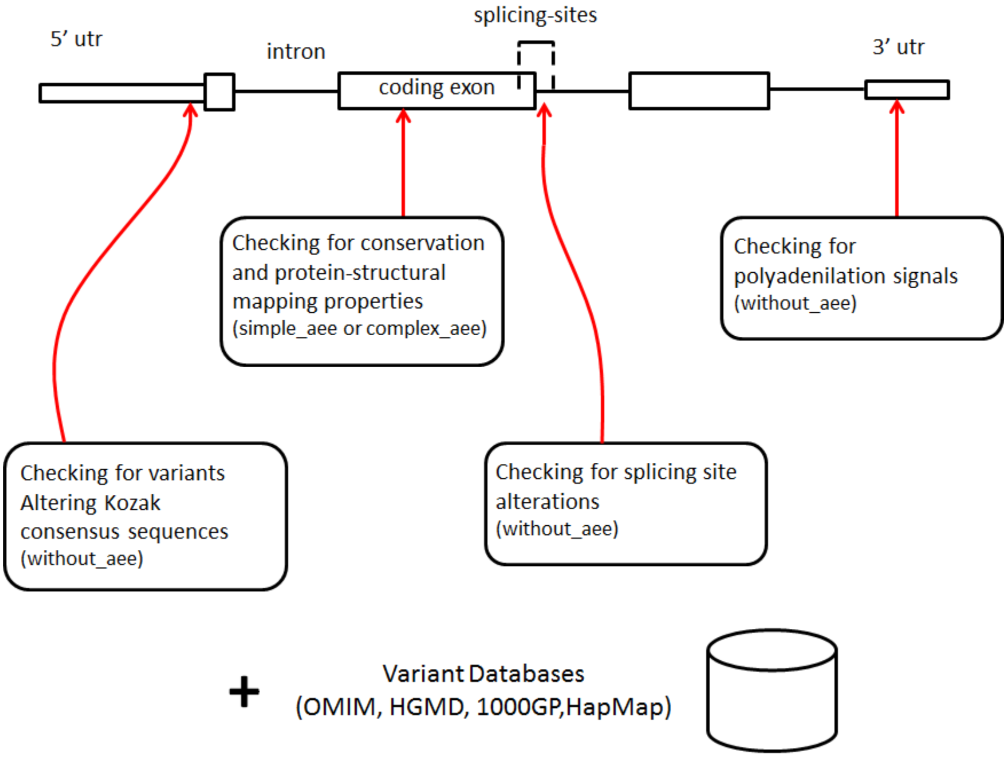

splicing-sites
5' utr
intron
3' utr
coding exon
Checking for conservation and protein-structural mapping properties (simple_aee or complex_aee)
Checking for polyadenilation signals (without_aee)
Checking for variants Altering Kozak consensus sequences (without_aee)
Checking for splicing site alterations (without_aee)
+
Variant Databases (OMIM, HGMD, 1000GP,HapMap)

Figure 20. MutationTaster capabilities based on diverse variant types. On the top a gene is represented by its introns (lines), coding exons (larger rectangles) and untranslated regions (smaller rectangles).

Conservation of residues or nucleotide sequences is computed analyzing the MSA of the sequence query and homologous sequences of ten different species both at nucleotide (by *bl2seq*[149]) and protein level (by *blastp*). MutationTaster then classifies the conservation into three classes (identical, partly conserved or not conserved) on the basis of amino acid sequence similarity or in two classes (conserved or not conserved) for the nucleotide sequence one. Moreover, MutationTaster also uses two evolutionary conservation scores computed on the multiple alignments of 46 vertebrate species, phyloP [129] and phastCons [150].

Protein structure features are retrieved by searching in SwissProt database the mapped properties to the protein of interest and check whether the analyzed variant overlaps some (directly affect) or may influence others in case e.g., of frameshift or splicing-site alterations (indirectly affect). Despite it has not been exactly clarified by the authors which protein features have been used to train the Naïve Bayes classifier, MutationTaster reports the directly or indirectly affected features such as helix and beta strands, domains, binding sites, active sites etc.

Variants overlapping splice sites (intron-exon borders) are processed by *NNSplice* [151] a splice site predictor algorithm that uses neural networks on dinucleotide frequencies to identify gene structures. 60 bases around the variant are used to compare wild-type and mutated sequences. Upon *NNSplice* results the variant is classified on the basis of its probability to alter existing splice site in positive/negative way, if an additional splice site is activated or the splice site completely lost.

MutationTaster also analyze consensus sequences in untranslated gene regions (5’utr and 3’utr), respectively checking whether the variant overlaps a Kozak consensus sequence (gccRccAUGG; R=purine) typically positioned upstream the start codon (AUG) and polyadenylation signal (PAS) regions consisting of two type of examers (AATAAA or ATTAAA) by *polyadq* algorithm. Both features play an important role into the corresponding mRNA expression.

Finally, MutationTaster implements several roles in order to limit false positives and negative rates by checking if variants have been already known to be disease-causing or a potential polymorphisms by querying variant disease databases such as HGMD, ClinVar, OMIM and natural background variant databases such as 1000GP and HapMap respectively.

The implemented Naïve Bayes classifier has been trained on diverse datasets, known disease and neutral variants from HGMD and 1000GP (with allele frequency threshold to assume neutrality) filtered upon three variant types: intronic or synonymous variants (*without_aee* model), amino acid substitutions (*simple_aee* model), coding variants such as frameshifts, introducing or disrupting a stop codon (*complex_aee* model).

2.5.2.4. GERP++

Non-coding variants are difficult to interpret, especially when they overlap with not-annotated genomic regions such as splice sites or regulatory elements. The effort to annotate all functional elements in the human genome such as the ENCODE project[152] is under continuous development and evolutionary conservation by comparative genomics is a central component in the pursuit of this goal. Sequence conservation, in fact, is a peculiarity of regions under negative selection that are reasonably supposed to have a biological function.

Genomic Evolutionary Rate Profiling (GERP++) [128, 153]is an MSA based algorithm that estimates evolutionary rates of each single alignment column and compares the inferred rates with a tree describing the null model, in order to define significance thresholds against a neutral background of substitution rate of the species under consideration. The identified constrained elements are then scored according to the “rejected substitutions” (RS) deficit.

MSA is built up by the multiple alignment of 34 mammalian species by the use of TBA algorithm [154].

Giving the MSA and a phylogenetic tree of the species in the alignment (see Figure 21B), GERP++ estimates the neutral rate for the entire tree in terms of neutral divergence among closely related species and extrapolates rate estimates over the entire branch length tree. During this step, if an alignment contains gap in a given position, the corresponding species is not considered into the computation of the neutral rate.

Subsequently, constrained elements at each position of the MSA are calculated in terms of RS score, that is the difference between the expected and observed evolutionary rate at each position. The observed value is the maximum likelihood estimate of the alignment column expected substitution count, and likelihood is maximized with respect to all branch lengths in the topology of the tree. The expected evolutionary rate for each column is obtained by pruning the tree in order to eliminate gaps and summing the residual branch lengths. Finally, constrained elements are identified by a threshold on the observed/expected rate allowing the merging of few diverse positions exhibiting a ratio lower than threshold (Figure 21A). The RS is the sum of the individual site differences between observed and expected rates of these merged elements. Finally, a p-value is assigned to each RS score, representing the probability of a random neutral segment of equal length having an equal or higher RS score.

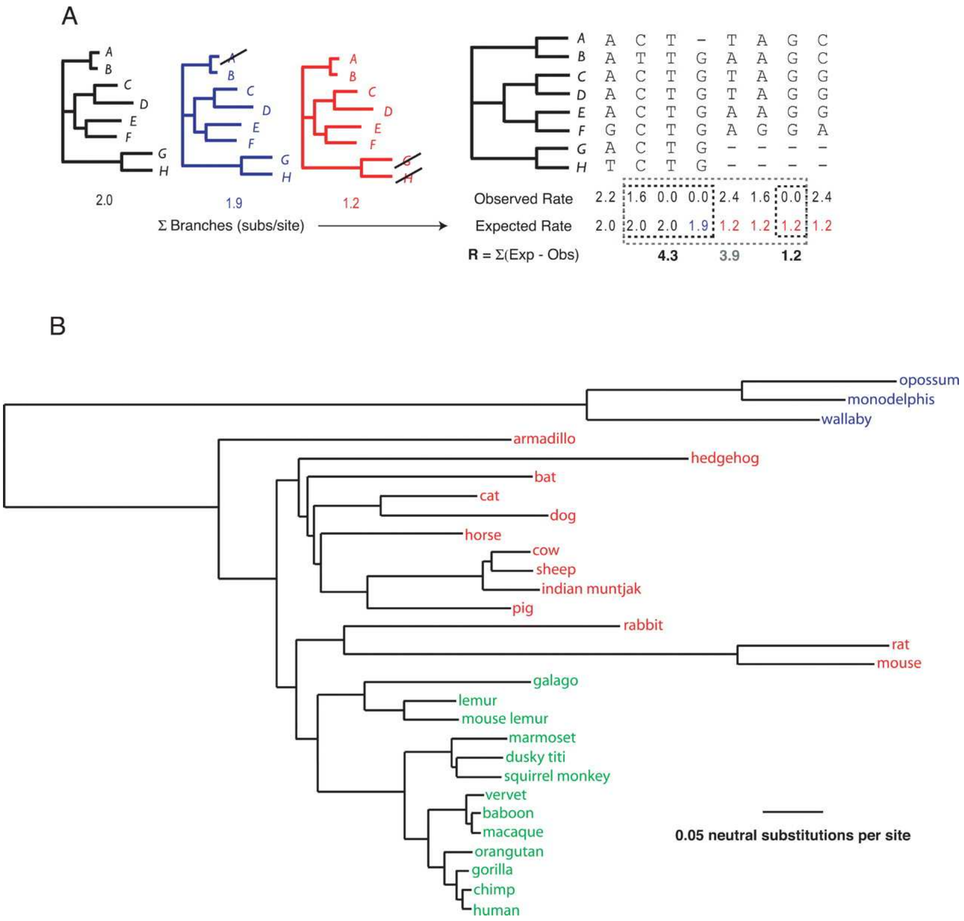


Figure 21. GERP++ workflow overview[153]

Chapter 3

Sequencing data management

This Chapter describes the Variant Management Systems (VMS) developed in the thesis. VMS has the aim to organize sequenced samples along with related genomic variants resulting from sequencing pipelines.
In particular, paragraph 3.1presents the system based on a relational database approach developed for such purpose, describes its technological components and discusses about results and limitations. Application results of this system are shown in Chapter 5.
Paragraph 3.2deals with the system based on NoSQL database, describes its paradigm, related technologies, performances and future directions.

3.1. A Relational Database for Genomic Variants and Annotations

In Chapter 2 the issue of management and interpretation of sequencing data has been introduced, focusing on genomic variants. The plethora of data produced by NGS is not straightforwardly interpretable and needs to be integrated with genomic and disease knowledge in order to correctly link genotype to phenotype features.

Genomic databases constitute this knowledge, but their data are not ready to use in the most part of cases: indeed they have to be accessed, processed and linked with target data objects, i.e. genomic variants. Organizing genomic variants along with their annotations is a requirement in order to learn from collected data and to proceed to variants interpretation.

We have therefore developed a VMS with the aim to store genomic variants collected from sequenced samples in several experiments, using different targeting enrichment platforms.

Variants come in VCF format, are imported into the system and are annotated for genomic knowledge such as mRNA transcripts, genes, proteins, amino acid consequences, polymorphisms and disease variant databases (see Figure 22). Once sample variants have been annotated, the system allows choosing the cohort of samples to use as the target (cases) and the cohort of samples to use as controls, for which aggregate data on variants are computed on the fly. The system queries data by applying user-filtering criteria on variant annotations such as amino acidic change type (non-synonymous, stop-causing), population variant frequency and/or variant attributes such as reads coverage and quality. Resulting variants can be further processed by variant prediction tools, such as PolyPhen-2 and MutationTaster, and results are stored without the needs to re-process the same variant entities resulting from other analysis.

The system relies on the MySQL Relational Database Management System (RDBMS) database for variant and genomic annotation storage and J2EE technologies for business logic and user interface. We therefore name the system as RDBVMS (Relational Database Variant Management System).

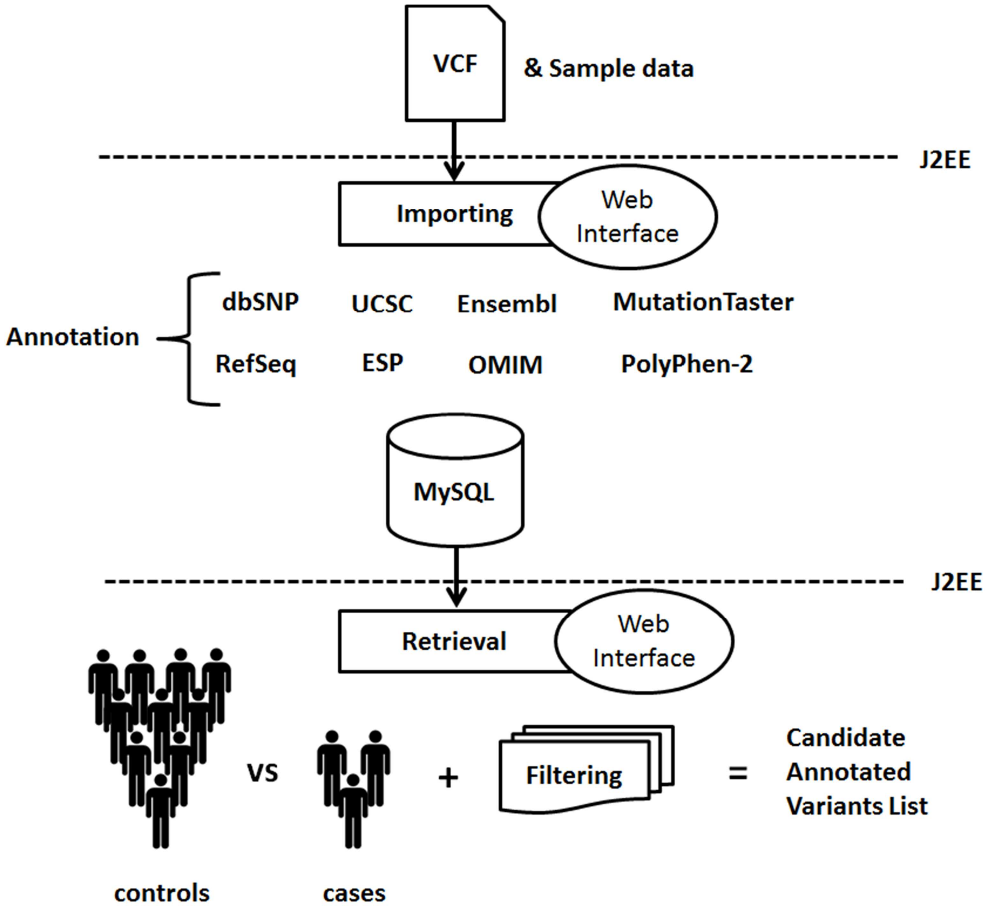


Figure 22. The RDBVMS general workflow.

3.1.1. Data Tier

The MySQL relational database management system (v. 5.5) has been chosen in order to store genomic variants and annotations. The choice has been leaded by the flexibility, scalability, transaction and indexes support that MySQL offers. MyISAM and InnoDB storage engines have been used both depending on reads/writes expected ratio for tables and the needs of constrains. Figure 23 represents a simplified version of the database schema.

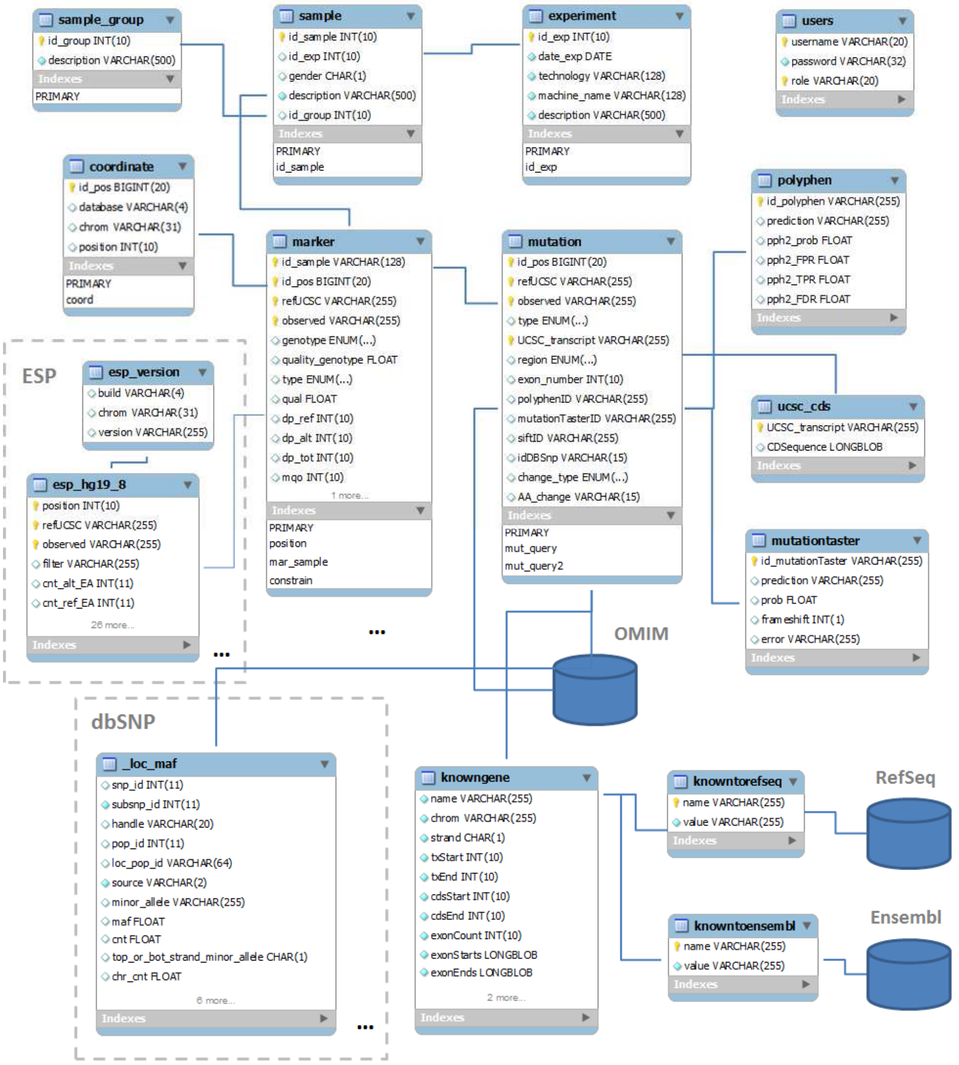


Figure 23.RDBVMS: the simplified Database Schema

The core of the database concerning genomic variants is made of four main tables: *sample*, *coordinate*, *marker* and *mutation*.

- *sample* contains data about the sequenced sample and in particular the internal id, the sample code, gender and the sample group which describe the NGS application used (e.g. exome or a specific gene panel). The sample id univocally identifies the row of the *sample* table.
- *coordinate* represents genomic variant by their reference assembly, chromosome and absolute 1-based chromosome position corresponding to the variant starting point. The coordinate id univocally identifies a row of the *coordinate* table.
- *marker* table reports genomic variants for each sample. A variant is identified by its coordinate id, reference and altered nucleotide

bases in a VCF-like format. Each marker holds data on that particular variant for the specific sample in terms of genotype, total reads coverage, coverage of the reference/altered bases, quality, filter annotations and other data coming from the VCF file. The coordinate id, reference/altered bases and the sample id univocally identify a row of the *marker* table.

- *mutation* instead, is a variant abstraction and store all those annotation data that depend on the variant within its transcript and not on the specific sample. Therefore mRNA transcript, coding region, amino acid change, pathogenic probabilities and other genomic annotation data are memorized within each row. The coordinate id, reference/altered bases and the mRNA transcript id univocally identify a row of the *mutation* table.

Coordinates have been separated from marker (and mutation) in order to make these tables independent from genome assembly and, in case of genomic reference update, only coordinates have to be changed by using a mapping algorithm such as liftOver [84].

The *mutation* table is filled during data import: business tier components annotate each variant using both data stored in the database and requests to web services over HTTP.

B-tree indexes have been set on those tables and fields that are generally queried, in order to speed up data retrieval.

3.1.2. Business and Presentation Tiers

The VMS has been developed as a J2EE-compliant application. The Java servlet technology has been chosen: it runs inside the Web Server (in our case Apache Tomcat), receives the HTTP request from the browser and generates dynamic content through Java Server Pages (JSP) components providing HTTP response back to the browser (see Figure 24).

Servlet technology is convenient in case of server intensive applications, such as those accessing a database, that is our case. Servlets are managed by the Servlet Container that is Tomcat and their flow and mapping can be configured through a specific xml file, the web application deployment descriptor (web.xml). A Main Servlet has been configured in order to manage every request from the Servlet Container, i.e. the user HTTP requests. In order to manage authentication for requested resources, a Filter has been configured: it preprocesses the request for the Main Servlet and check whether a specific session attribute has been initialized (i.e. the username) and it eventually address to the login page for authentication.

Main Servlet is used in combination with JSPs and Plain Old Java Objects (POJOs). The first are used to manage the dynamic content of web pages, the seconds belonging to the business tier, to perform operations such as querying database and manipulate data.

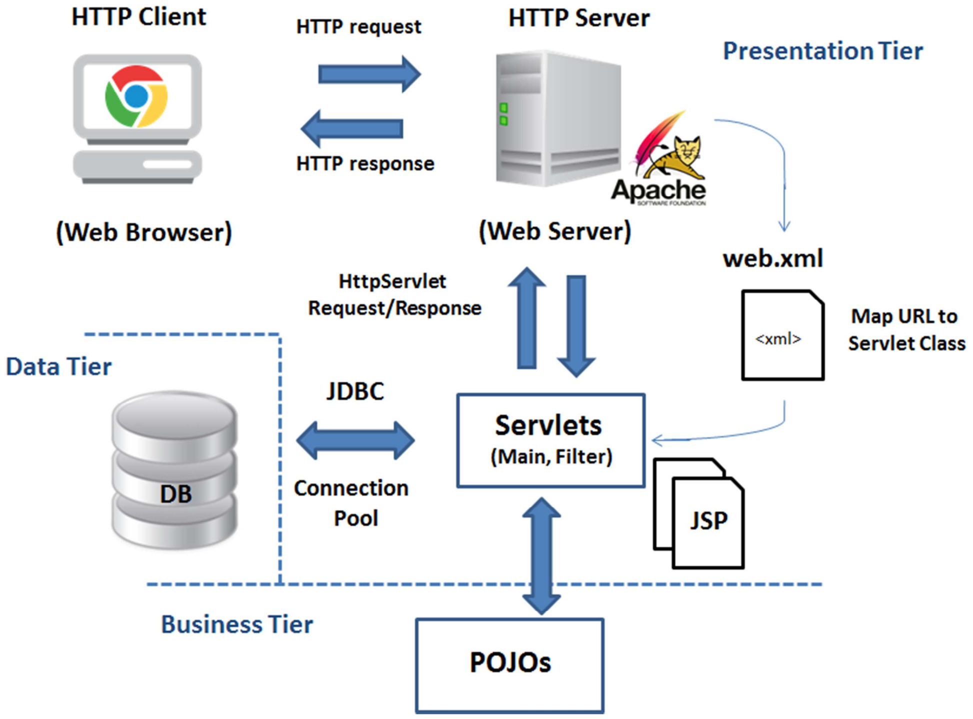


Figure 24.RDBVMS architecture

The Main Servlet has been coded in order to:

- load configuration parameters: a configuration file has been used in order to store those parameters needed to the system and that may change in time, e.g. URL and credentials of the MySQL database, local directories used as file repositories for database dumps.

- dispatch the user requests to POJOs and JSPs: the Servlet interacts with the Servlet Container through predefined objects for request and response (HttpServletRequest and HttpServletResponse) and methods to manage HTTP GET and POST requests. Each request hold a parameter (command) that is the code used by the Servlet to perform an action, such as instantiate a POJO, call its methods and, finally, invoke JSPs (see AppendixA.2, *ServletNGS.java*).

- initialize a connection pool to the database: in order to limit database connections creation and manage them, a connection pool logic has been implemented. The connection pool is initialized within the Main Servlet and a new connection is created or pulled from the pool when a POJO asks for a connection to the database in order to perform a query (see AppendixA.2, *ConnectionPool.java*).

POJOs have been used in different ways such as representing data objects grabbed from the database, processing data during data importing and presentation phases, performing requests over HTTP and updating data tables. Figure 25 depicts the simplified UML class diagram of the application: *LoginFilter* is between the Servlet Container and the Main Servlet (called *ServletNGS*) and it manages authentication to the requested resources (e.g. JSPs). *DBAnalize* is the Java class used to perform the major part of CRUD (Create, Read, Update and Delete) operations on the database and makes use of POJOs representing database entities, such as *Sample*, *Coordinate*, *Marker* and *Mutation*.

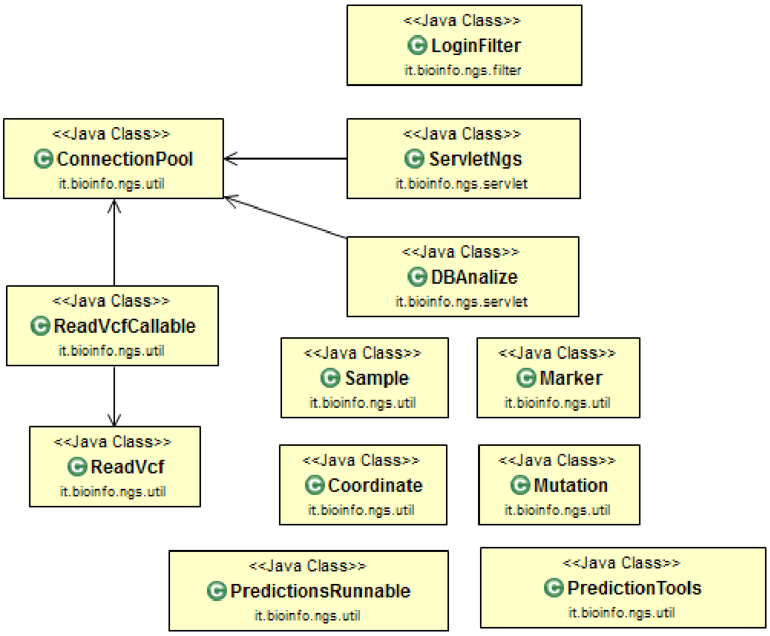
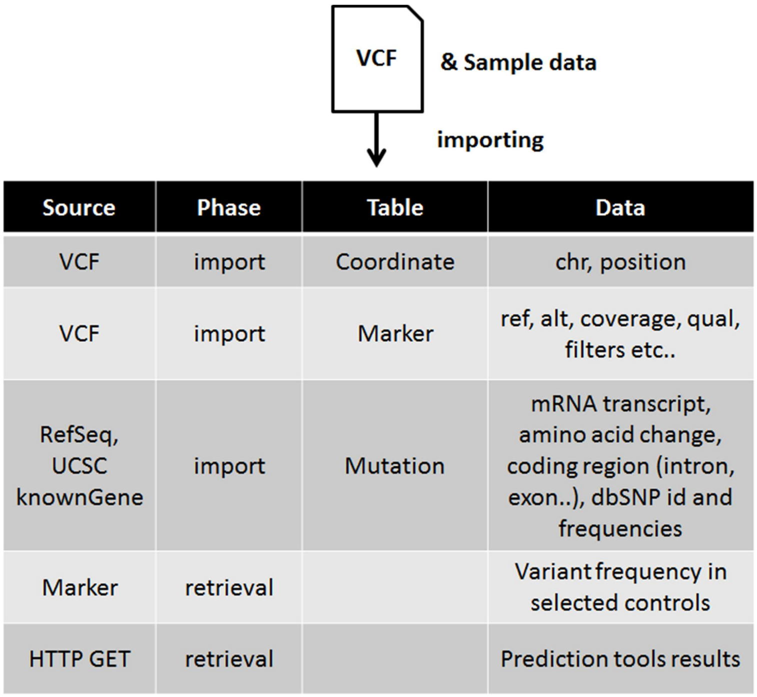


Figure 25.RDBVMS: the simplified UML diagram of classes

3.1.3. Data Workflow

Genomic variants follow a specific workflow. The user imports and retrieves sample variants interacting with the web interface. In Figure 26the main VMS variant annotations are reported, distinguishing between those that are performed on the fly during data import and data retrieval.



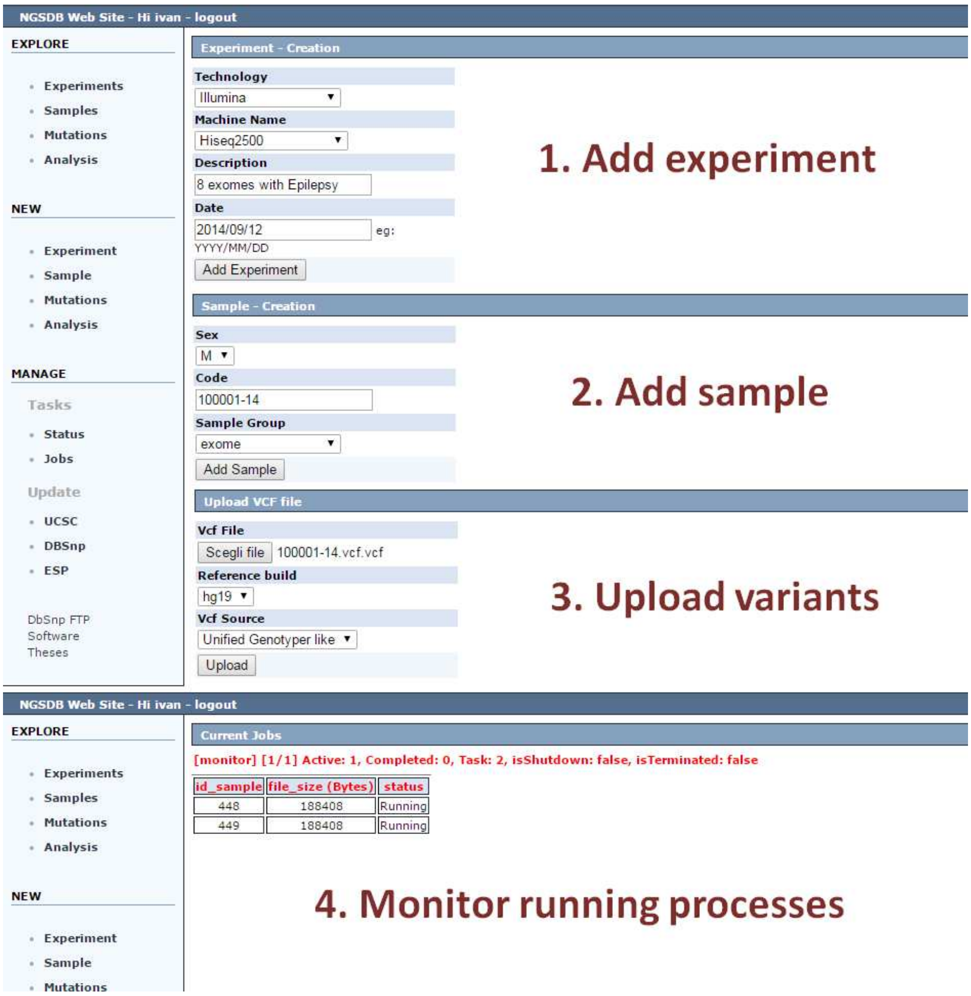
Source	Phase	Table	Data
VCF	import	Coordinate	chr, position
VCF	import	Marker	ref, alt, coverage, qual, filters etc..
RefSeq, UCSC knownGene	import	Mutation	mRNA transcript, amino acid change, coding region (intron, exon..), dbSNP id and frequencies
Marker	retrieval		Variant frequency in selected controls
HTTP GET	retrieval		Prediction tools results

Figure 26. Overview of the RDBVMS annotations distinguishing those computed during data import and retrieval.

3.1.3.1. Data Import

Variants are imported into the system through the web interface (see Figure 27). VCF file format is the accepted standard and can process the specific fields produced by both GATK Unified Genotyper [49] and another variant caller, MuTect [155]. The user fills the form relative to the sequenced sample and the NGS experiment to which it belongs and uploads the related VCF file.

On back end the request is managed by the *ServletNGS* which instantiates the objects and calls the related methods to parse the file (*ReadVCF*, see Figure 25). Each variant in the VCF file is then annotated with several genomic data by querying and processing specific tables content: the UCSC table of known genes is queried by genomic intervals related to the transcriptional regions. Exons coordinates are scanned in order to classify the variant as an “exonic”, “intronic”, “splice-site” or untranslated “utr” one. In case of an exonic variant, the amino acid sequence of the related mRNA transcript is pulled (*ucsc_cds* table, see Figure 23)and the amino acid change is calculated on the basis of the variant allele. The variant is therefore associated to the relative dbSNP-id, in case it exists, and relative frequencies. Some dbSNP database tables have been dumped into the VMS database in order to link the right id to each variant depending on the reference and alternate allele reported in the VCF file, while views regarding dbSNP allele frequencies have been elaborated in order to associate the variant allele to the minor allele frequencies in the general population. Notably, in case of G/C and A/T allele ambiguities, the Illumina TOP/BOT strand convention[92] has been implemented in order to be sure to refer to the same variant allele strand reported in dbSNP. Not always the TOP/BOT information is available, and in such a case, the warning of possible ambiguity is reported as additionally fields in the final report.



Experiment - Creation

Technology: Illumina
Machine Name: Hiseq2500
Description: 8 exomes with Epilepsy
Date: 2014/09/12 eg: YYYY/MM/DD
Add Experiment

1. Add experiment

Sample - Creation

Sex: M
Code: 100001-14
Sample Group: exome
Add Sample

2. Add sample

Upload VCF file

Vcf File: Scegli file 100001-14.vcf.vcf
Reference build: hg19
Vcf Source: Unified Genotyper like
Upload

3. Upload variants

Current Jobs

[monitor] [1/1] Active: 1, Completed: 0, Task: 2, isShutdown: false, isTerminated: false

id_sample	file_size (Bytes)	status
448	188408	Running
449	188408	Running

4. Monitor running processes

Figure 27.RDBVMS - Data import web interface

While VCF related data are stored in the *Marker* table, genomic annotations are stored into the *Mutation* one. The latter contains variant abstractions, therefore rows in this table do not depend on samples: a variant is processed for annotation only if its correspondent variant object does not exist in the *Mutation* table, in order to speed up the import phase and avoid SQL exceptions of duplicates keys.

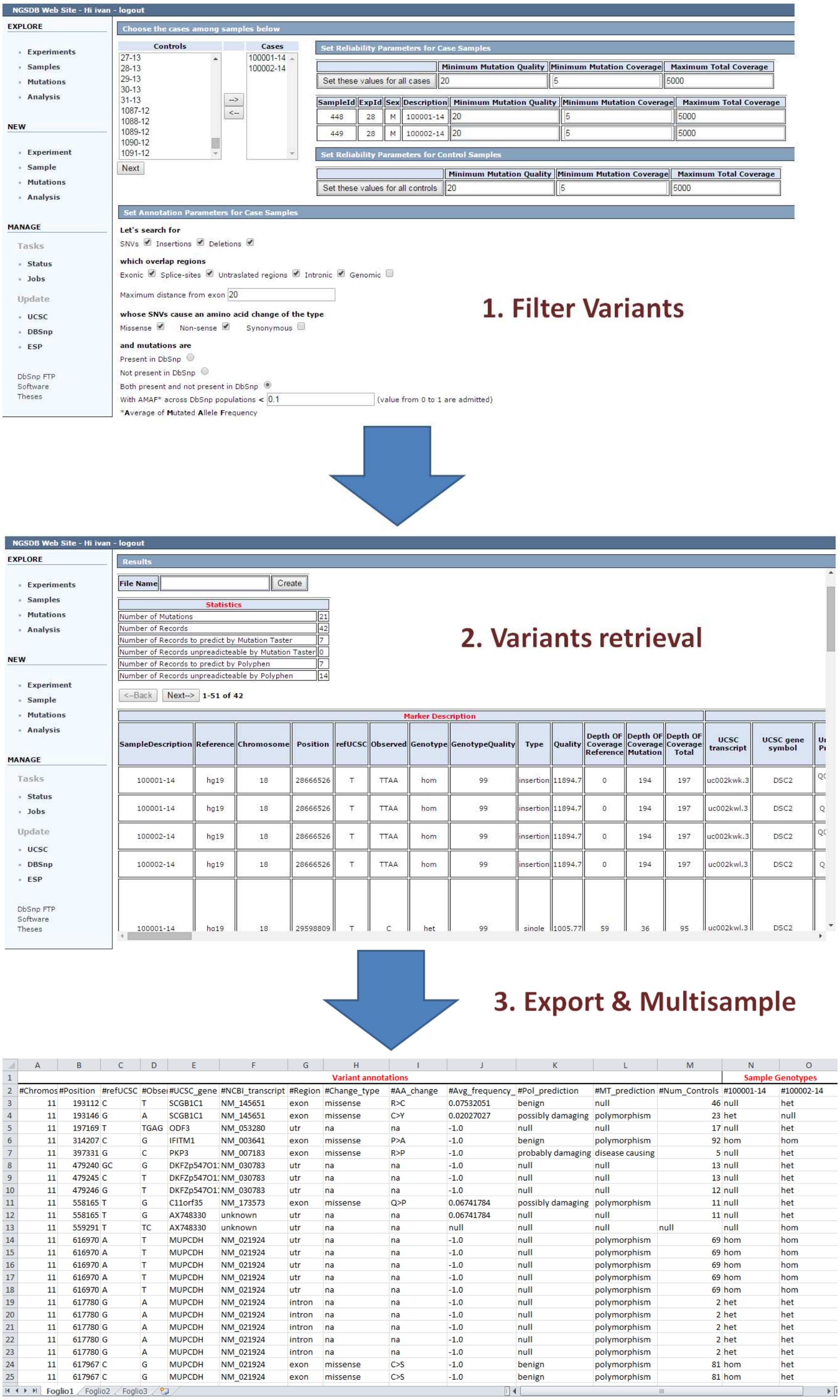
Data import has been coded in an asynchronous way implementing the Java *Callable* interface (see Figure 25 and *ReadVCFCallable.java in* AppendixA.2) and pulling the object into an Executor with a single thread. Therefore, the user can upload several VCF files in cascade and monitoring the importing process for each sample.

3.1.3.2. Data Retrieval

Data retrieval allows the user to download genomic variants belonging to a selected sample cohort and respecting several optional filtering criteria. The user first chooses all the samples to include in the analysis by selecting them on the assigned sample group or individually. Then, the samples cohort to be used as cases (for which the genomic variants are retrieved) and as controls can be chosen. Subsequently, filtering criteria based on genomic annotation (e.g. exonic region and dbSNP frequency) and per sample variant attributes (e.g. coverage, quality) can be selected in order to reduce the space of genomic variants to retrieve and further analyze. During the retrieval of genomic variants, two main operations are performed on the fly:

- Variant frequencies calculated among the controls cohort: depending on the samples control cohort for the specific analysis, dedicated Java classes retrieve, for each sample-case variant, the number of controls samples that share the same variant (rows of *Marker* table) along with data related to the matched sample-variants (markers) such as the genotype and coverage. In the final report, for each variant reported in at least one sample-case, aggregate values computed on the sample cohort are available, e.g. variant frequency, heterozygosity/homozygosity rates.

- Variant functional predictions: PolyPhen-2 and MutationTaster (see 2.5.2) are run in order to assess the variant pathogenicity of the resulting variants. A predefined Java class (*PredictionTools*, see Figure 25)has been coded in order to run variant predictions by directly making requests over HTTP (POST method) to the correspondent web services, using the *Apache commons HttpClient*[156] libraries. Data are first processed in order to respect the requested format (e.g. mRNA transcript conversions), sent over HTTP to the web services and retrieved using HTTP GET method. Results are saved into the database and linked to the *Mutation* table: in fact, only those variants lacking a stored prediction follow this procedure. Java classes in multithread mode have been coded in order to run requests in a parallel fashion and speed up the retrieval of prediction results (seeFigure 25 and*PredictionsRunnable.java* inAppendixA.2).



Choose the cases among samples below

Controls		Cases
27-13 28-13 29-13 30-13 31-13 1087-12 1088-12 1089-12 1090-12 1091-12	--> <--	100001-14 100002-14

Next

Set Reliability Parameters for Case Samples

	Minimum Mutation Quality	Minimum Mutation Coverage	Maximum Total Coverage
Set these values for all cases	20	5	5000

SampleId	ExpId	Sex	Description	Minimum Mutation Quality	Minimum Mutation Coverage	Maximum Total Coverage
448	28	M	100001-14	20	5	5000
449	28	M	100002-14	20	5	5000

Set Reliability Parameters for Control Samples

	Minimum Mutation Quality	Minimum Mutation Coverage	Maximum Total Coverage
Set these values for all controls	20	5	5000

Set Annotation Parameters for Case Samples

Let's search for

SNVs ☑ Insertions ☑ Deletions ☑

which overlap regions

Exonic ☑ Splice-sites ☑ Untraslated regions ☑ Intronic ☑ Genomic ☐

Maximum distance from exon 20

whose SNVs cause an amino acid change of the type

Missense ☑ Non-sense ☑ Synonymous ☐

and mutations are

Present in DbSnp ○

Not present in DbSnp ○

Both present and not present in DbSnp ●

With AMAF* across DbSnp populations < 0.1 (value from 0 to 1 are admitted)

*Average of Mutated Allele Frequency

1. Filter Variants

Results

File Name: Create

Statistics	
Number of Mutations	21
Number of Records	42
Number of Records to predict by Mutation Taster	7
Number of Records unpreadicteable by Mutation Taster	0
Number of Records to predict by Polyphen	7
Number of Records unpreadicteable by Polyphen	14

<--Back Next--> 1-51 of 42

2. Variants retrieval

Marker Description															
SampleDescription	Reference	Chromosome	Position	refUCSC	Observed	Genotype	GenotypeQuality	Type	Quality	Depth OF Coverage Reference	Depth OF Coverage Mutation	Depth OF Coverage Total	UCSC transcript	UCSC gene symbol	Ui Pi
100001-14	hg19	18	28666526	T	TTAA	hom	99	insertion	11894.7	0	194	197	uc002kwk.3	DSC2	QC
100001-14	hg19	18	28666526	T	TTAA	hom	99	insertion	11894.7	0	194	197	uc002kwl.3	DSC2	Q
100002-14	hg19	18	28666526	T	TTAA	hom	99	insertion	11894.7	0	194	197	uc002kwk.3	DSC2	QC
100002-14	hg19	18	28666526	T	TTAA	hom	99	insertion	11894.7	0	194	197	uc002kwl.3	DSC2	Q
100001-14	hg19	18	29598809	T	C	het	99	single	1005.77	59	36	95	uc002kwl.3	DSC2	

3. Export & Multisample

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1								Variant annotations						Sample Genotypes	
2	#Chromos	#Position	#refUCSC	#Obse	#UCSC_gene	#NCBI_transcript	#Region	#Change_type	#AA_change	#Avg_frequency_	#Pol_prediction	#MT_prediction	#Num_Controls	#100001-14	#100002-14
3	11	193112	C	T	SCGB1C1	NM_145651	exon	missense	R>C	0.07532051	benign	null	46	null	het
4	11	193146	G	A	SCGB1C1	NM_145651	exon	missense	C>Y	0.02027027	possibly damaging	polymorphism	23	het	null
5	11	197169	T	TGAG	ODF3	NM_053280	utr	na	na	-1.0	null	null	17	null	het
6	11	314207	C	G	IFITM1	NM_003641	exon	missense	P>A	-1.0	benign	polymorphism	92	hom	hom
7	11	397331	G	C	PKP3	NM_007183	exon	missense	R>P	-1.0	probably damaging	disease causing	5	null	het
8	11	479240	GC	G	DKFZp547O1	NM_030783	utr	na	na	-1.0	null	null	13	null	het
9	11	479245	C	T	DKFZp547O1	NM_030783	utr	na	na	-1.0	null	null	13	null	het
10	11	479246	G	T	DKFZp547O1	NM_030783	utr	na	na	-1.0	null	null	12	null	het
11	11	558165	T	G	C11orf35	NM_173573	exon	missense	Q>P	0.06741784	possibly damaging	polymorphism	11	null	het
12	11	558165	T	G	AX748330	unknown	utr	na	na	0.06741784	null	null	11	null	het
13	11	559291	T	TC	AX748330	unknown	utr	na	na	null	null	null	null	null	hom
14	11	616970	A	T	MUPCDH	NM_021924	utr	na	na	-1.0	null	polymorphism	69	hom	hom
15	11	616970	A	T	MUPCDH	NM_021924	utr	na	na	-1.0	null	polymorphism	69	hom	hom
16	11	616970	A	T	MUPCDH	NM_021924	utr	na	na	-1.0	null	polymorphism	69	hom	hom
17	11	616970	A	T	MUPCDH	NM_021924	utr	na	na	-1.0	null	polymorphism	69	hom	hom
18	11	616970	A	T	MUPCDH	NM_021924	utr	na	na	-1.0	null	polymorphism	69	hom	hom
19	11	617780	G	A	MUPCDH	NM_021924	intron	na	na	-1.0	null	polymorphism	2	het	het
20	11	617780	G	A	MUPCDH	NM_021924	intron	na	na	-1.0	null	polymorphism	2	het	het
21	11	617780	G	A	MUPCDH	NM_021924	intron	na	na	-1.0	null	polymorphism	2	het	het
22	11	617780	G	A	MUPCDH	NM_021924	intron	na	na	-1.0	null	polymorphism	2	het	het
23	11	617780	G	A	MUPCDH	NM_021924	intron	na	na	-1.0	null	polymorphism	2	het	het
24	11	617967	C	G	MUPCDH	NM_021924	exon	missense	C>S	-1.0	benign	polymorphism	81	hom	het
25	11	617967	C	G	MUPCDH	NM_021924	exon	missense	C>S	-1.0	benign	polymorphism	81	hom	het

Figure 28.RDBVMS - Data Retrieval Workflow by web interface

The final report consists in a list of genomic variants extracted from the database and enriched by the aforementioned data. Each row is univocally identified by the sample code, variant coordinates and the mRNA transcript to which it overlaps. The report can be exported as a tab-delimited text file along with genomic annotations and variant frequencies on dbSNP, ESP and the control cohort. Finally, the exported file can be optionally processed in order to aggregate sample genotypes on variants, resulting in a multi-sample variant file, easy to be managed and further analyzed by a spreadsheet software such as Microsoft Excel (see Figure 28).

3.1.4. Results and Discussion

The developed RDBVMS has been used to perform several case-control studies by extracting the related filtered list of genomic variants. This procedure allowed the researchers to test their hypothesis on the basis of the annotated variants in the final report. In particular, the genetic causes of different rare and complex diseases have been exploited starting from the RDBVMS output. Some successful case studies have been reported in Chapter 5.

The system allowed to store genomic variant data of 437 sequenced samples divided between whole-exome (123) and gene-panel applications (314).A total of 33,799,523 genomic variants have been collected along with data on genotypes and metrics for each sample. Figure 29reports row counts for each main table of the database respect to the schema in Figure 23.

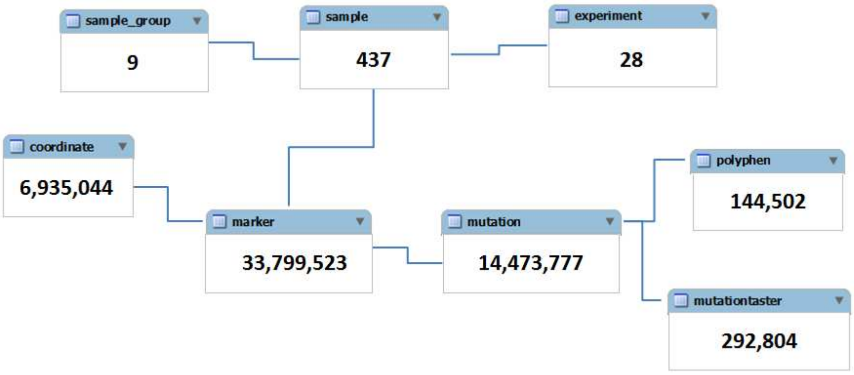


Figure 29.RDBVMS - Row counts for each genomic-variant related table

In terms of memory, the database occupies 293GB on hard disk drive and it has been configured within both the presentation and business tiers on a single workstation with an Intel i3 CPU and 4GB RAM.

The system, in fact, has been originally developed in order to be light in terms of CPU and RAM consuming, by limiting the number of processes to be managed in multi-threading mode both during data import and data retrieval.

Despite the efficacy showed by the RDBVMS to give reliable results in terms of variant annotation and retrieval, the system drawbacks rely on performances in terms of computational times and flexibility. Actually, the system imports, in average, a single sample variant in 88.6 ms: considering that a whole exome sample holds up to 60K raw genomic variants, that is without applying any filter, the system needs, in average, about 1 hour to import an entire whole exome variant set as well as to retrieve its filtered controls-matched genomic variants with the actual data amount on the database.

For flexibility we refer, in our case, to the capacity of the system to adapt data and data structure to a changeable environment in terms of data updates and new requisites, respectively. Public genomic data are in continuous evolution and for certain databases (such as UCSC genome browser) updates are done daily. New insights on genomic variants, genes and diseases come up at an impressive rate, especially nowadays, with high throughput technologies widespread all over the world. Therefore it is not uncommon that researchers ask for the most update genomic variant annotations in order to increase their power in discovery and check the existing literature for newly disease-related genes and variants.

The developed RDBVMS is not much flexible in this sense, thus its genomic data sources and the elaborated genomic variant annotations are integrated into the database itself (e.g. the *Mutation* table). Therefore, efforts are needed to update the database and to maintain data consistency; moreover, the insertion of new data fields is challenging due to the intrinsic rigidity of the relational database schema.

Another limitation of the developed system is the lack of phenotype data related to the individuals to which the samples belong. While managing case-control studies can be relatively easy in family-based approaches where only few samples have to be analyzed and the presence of few individuals can allow to manually select samples to be included in the analysis, it is not the case with large sample cohorts, where a selection criteria based on a standardized phenotype terminology would be strategic.

3.2. A NoSQL Database for Genomic Variants and Annotations

The knowhow on genomic data annotation and integration acquired during the RDBVMS development, joined with the experienced issues with its use, the development of efficient genomic annotation algorithms [97, 157] and the spread of a new generation of flexible databases, leaded us to a radical change of strategy in genomic variant management.

We developed, therefore, a new VMS with the main goal of high performances in terms of computational time, flexibility of the data structure and, last but not least, the integration of phenotype and genotype data.

The newly VMS consists of different interacting modules (see Figure 30).

- The *genomic annotator*. We chose ANNOVAR (see 2.4.3.2) to annotate genomic variants coming from VCF files.

- The *genomic variants database*. We chose CouchDB [158], a NoSQL database that allows for data structure flexibility.

- The *phenotype database*. We chose i2b2 platform[4, 159] in order to query patients phenotype data and in the meanwhile to interface with genomic variants database.

Briefly, genomic variants in VCF format are first processed by ANNOVAR and, subsequently, a plain text file with variants and genomic annotations is produced. The file is further processed in order to be imported into the CouchDB NoSQL database, where genomic variants are stored as documents within all relative annotations. An ad-hoc i2b2 software module has been developed in order to communicate both with CouchDB and the i2b2 core. Moreover another plugin has been developed to provide an interface to build genotype query on a selected cohort of patients (see Figure 30).

My contribution to this project has been limited to the conception of the idea, the design of the software and the genomic data annotation.

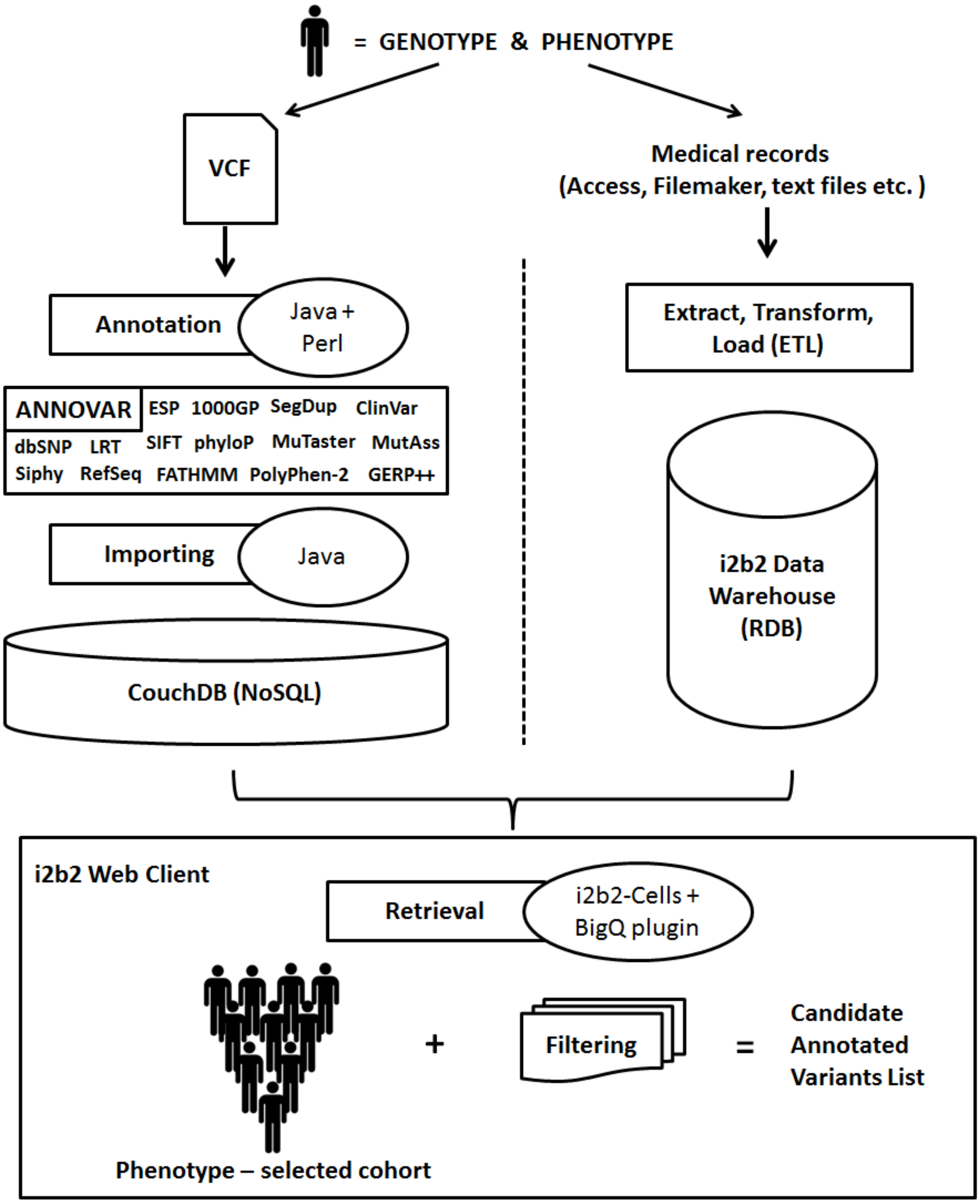

= GENOTYPE & PHENOTYPE
VCF
Medical records
(Access, Filemaker, text files etc.)
Annotation
Java + Perl
Extract, Transform, Load (ETL)
ANNOVAR
ESP 1000GP SegDup ClinVar
dbSNP LRT SIFT phyloP MuTaster MutAss
Siphy RefSeq FATHMM PolyPhen-2 GERP++
Importing
Java
i2b2 Data Warehouse (RDB)
CouchDB (NoSQL)
i2b2 Web Client
Retrieval
i2b2-Cells + BigQ plugin
+
Filtering
=
Candidate Annotated Variants List
Phenotype – selected cohort

Figure 30. NoSQL-VMS + i2b2 workflow overview

3.2.1. NoSQL databases

NoSQL stands for “Not only SQL” and it is a movement (begun early 2009) promoting a new generation of open-source databases especially suited for Big Data and therefore used by big companies, such as Google, Amazon, Facebook and many others, to manage their overwhelming amount of data. NoSQL solutions are all characterized by the following peculiarities:

- *non-relational*: a NoSQL database is not based on the traditional structure of data tables where all the related fields (columns) is constrained to a specific type and, in the meanwhile, tables have to be normalized in order to reduce data duplication. NoSQL database tends to be schema-less or to adopt semi-structured schema, allowing data duplication and heterogeneity.

- *distributed*: NoSQL databases are designed to workon distributed environments. Therefore, the database does not rely on a single machine, but conversely, it holds data duplicates and/or replicates among a net of interconnected machines (or nodes) in a cluster environment.

- *horizontally scalable:* relational databases scale “vertically”, by adding hardware resources to the high performance server when needed. Conversely, NoSQL scales “horizontally”, by adding a new node in the distributed database and with no impact ondata availability.

- *BASE*: differently from relational database, adopting the ACID (Atomicity, Consistency, Isolation, Durability) set of properties in transactions, NoSQL database are Basically Available, Soft state and Eventually consistent (BASE) in order to achieve much higher performance and scalability with the tradeoff of consistency, guaranteed only with a reasonable delay.

In the last years many NoSQL databases (about 150) have been developed[160], each one with its peculiarities, but generally classifiable under several categories. Among them:

- *Document store*: the principal component is the “document”, an object that encapsulates data in some standard format (e.g. XML, YAML, JSON). Two examples: CouchDB, MongoDB.

- *Column store*: basing on rows and columns with limited constrains, their scalability model is splitting both rows and columns over multiple nodes. Rows are similar to documents and can be collected into groups (table or families). Two examples: HBase, Cassandra.

- *Key Value/Tuble store*: based on data dictionary, where to each unique and indexed key is associated only one value, typically consisting of a collection of elements (bins)[161]. Two examples: DynamoDB, Riak.

- *Graph database*: based on graph models, data consist of nodes and relationship between nodes. Data are not queried but "navigated" along the graph in order to match data queries. Two examples: OrientDB, Neo4j.

3.2.1.1. CouchDBOverview

CouchDB [158] is an open source Apache project since 2008. Written in Erlang [162], is a NoSQL document-based database, using JavaScript Object Notation (JSON) as the standard format for documents, a RESTful programming interface and Javascript as the query language combined to the MapReduce paradigm.

JSON is an open standard format used to transmit data objects consisting of attribute-value pairs. A value can be one of the traditional data type (number, Boolean) or an object such as a String, an array or another document.
In CouchDB, a univocal id must be assigned to each JSON document. Generally the best choice is to use the ones generated by CouchDB itself, that is the Universally (or Globally) Unique Identifier (UUID) consisting in randomly assigned numbers with a low collision probability. Hereby an example of JSON document is reported.

```
{
  "couchdb": "Welcome",
  "uuid": "dca7f93eb1b5d7998ac468a002bcde44",
  "version": "1.6.1",
  "vendor": {
    "version": "1.6.1",
    "name": "The Apache Software Foundation"
  }
}
```

CouchDB items are associated to a Uniform Resource Identifier (URI) accessible via HTTP. CRUD operations on the database are performed using a RESTful API, therefore all HTTP methods (POST, GET, PUT, DELETE) can be used. This allows using programming languages such as cURL to interact with CouchDB, but also REST clients are available for many coding languages such as Java, Python and Ruby.

The simplest query to CouchDB is to request a single document by its UUID and associated URI. E.g. to request by cURL the previous document stored into the "couch" database accessible at localhost on 5984 port:

```
curl –X GET
http://localhost:5984/couch/dca7f93eb1b5d7998ac468a002bc
de44
```

and the relative JSON file is sent back to client. However it assumes to know exactly the document id; therefore it would be impractical to use.
Documents in CouchDB can be organized into groups, called "views". A view is designed by JavaScript to specify attribute-value constraints corresponding to the query requirement and by implementing the *map* and *reduce* functions in the MapReduce style. *Map* functions are called once on each document: the document can be skipped (if it does not respect the constrain) or can be transformed (*emit*) into one or more view rows as key/value pairs. All the constraints of the *map* functions have to refer exclusively to the document attributes. View rows are indexed, that is inserted into a B-tree storage engine and sorted by key: to look up by key or key range is therefore extremely efficient, performing in logarithmic time. E.g. to create a view of CouchDB documents respecting the previous example format, one can code a *map* function on the document "version" attribute by the following JavaScript lines:

```
function(doc) {
  var version;
  if(doc.version){
       version= doc.version;
  }
  emit(version,null);
}
```

The *map* function takes the document as the argument and *emits* the view row for each document consisting in the version as the key and a null value in this case. The document UUID comes by default with the key-value pair and a name has to be assigned to the view (e.g. *versionView*), which rows will be sorted by version values, ready to be queried. Note that keys can be a combination of two or more document attributes, resulting in sorted view rows following the order of attributes-values established into the *emit* function. Once the view has been created one can modify it by accessing to its design document (a JSON file as well), modify the JavaScript code corresponding to the view and re-upload it to CouchDB, resulting in the view re-computation. For example, one can retrieve every raw of the *versionView*, saved into the "*versionD*" design document, by the following command:

```
curl –X GET
http://localhost:5984/couch/_design/versionD/_view/versi
onView
```

and a JSON document containing the list of documents UUIDs with associated key-values are sent back to the client. To query the view for a certain value, instead, should be passed an additional parameter, e.g. adding to the URI the "?key=(value)" string:

```
curl –X GET
http://localhost:5984/couch/_design/versionD/_view/versi
onView?key="1.6.1"
```

and only documents respecting this constraint are sent back.
Range queries, by using the "?startkey=(valueStart)&endkey=(valueEnd)" format are supported as well.

Reduce functions can be optionally used in combination with *map* functions in order to report data aggregates grouping by row keys (it navigates the relative B-tree), such as counting the number of rows within a viewor to calculate averages on related values. Hereby the *reduce* function reporting the number of view rows:

```
function(keys,values) {
  return(values.length)
}
```

If combined with the *map* function of the previous example, the following query

```
curl –X GET
http://localhost:5984/couch/_design/versionD/_view/versi
onView?startkey="1.6.1"&endkey="1.6.2"
```

reports the number of documents having version between 1.6.1 and 1.6.2.
Because *map* functions are applied to each document in isolation, computation can be high parallelized within and across nodes where the database is distributed.

3.2.2. The i2b2 Platform

Informatics for Integrating Biology and Beside (i2b2) is an NIH-funded National Center for Biomedical Computing based at Partners HealthCare System in Boston, MA USA.
The i2b2 Center effort has been focused on developing an open source scalable informatics framework designed to bridge clinical research data and the data bases arising from basic research (such as genomics):in such way the understanding of genetics underlie diseases and the design of

targeted therapies for individual patients would be speeded up by a solid, fast and easy query able integrated data resource.
The i2b2 platform nowadays is used worldwide in many hospitals and it is supported by an active developers community as well.

Built on multiple (hive)server-side software modules (cells) that communicate through their integrated XML-based web services (see Figure 31), the i2b2 platform consists of several core and optional cells. Each cell either hold data or business tier. For the purpose of this thesis, only two i2b2 core cells are briefly discussed below.

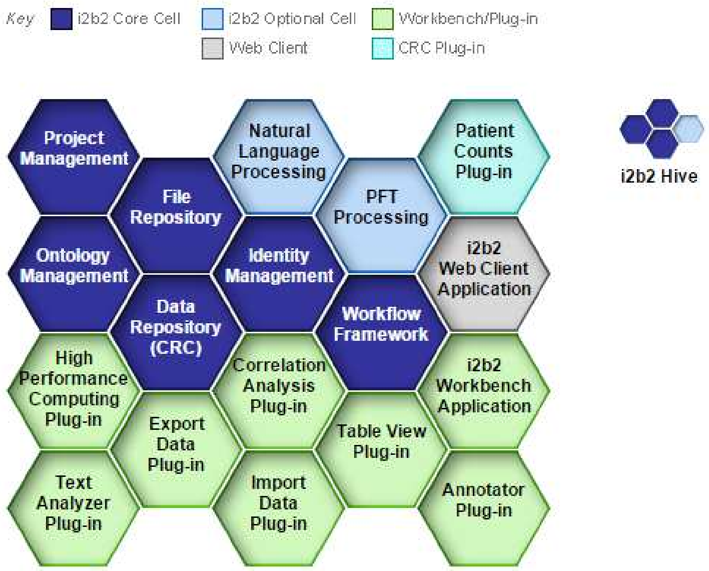


Figure 31.i2b2 Hive and accessory modules

The Clinical Research Chart (CRC) is the i2b2 cell through which clinical data of patients are accessed and relies on a “star schema” data warehouse [163] with a central “fact” table representing observations about patients, recorded by a specific observer regarding a specific concept (e.g. a lab test outcome or a reported phenotype) [164]. The concepts are coded and can be related to heterogeneous data. In other words, a concept is represented as a row rather than a column into the data model, following the entity-attribute-value (EAV) model [165]. Because concepts of all patients are stored into the fact table, indexing on the latter results into efficient cross-patients queries.

The Ontology Management (ONT) cell manages i2b2 vocabulary definitions (ontologies) and contains information about relationships between concepts for the entire hive. The controlled vocabulary holds categories organized as a hierarchical structure with relationships between terms. The other i2b2 cells accesses to the ONT in order to give semantic meaning to the data, e.g. the CRC associates each row of the main fact table to an ontology term in order to support data retrieval through the ONT.

The i2b2 web client (see Figure 32) allows performing ad hoc queries in order to find patient set respecting constraints based on ontology terms. The ontology is represented as a tree structure and its terms can be dragged and dropped to the query tool. The latter implements a Venn-diagram-like logic: terms into the same panel are OR’d together, conversely, terms in different panels are logically AND’d. When query is executed, web client send the request to the CRC and query results are shown into the Query History panel. The query result is the patient set corresponding to defined constraints and can be filtered again (by dragging it from the Query History panel to the Query Tool again) or passed to i2b2 plug-ins for further analysis. Plug-ins are software modules, accessible by the i2b2 Analysis Tool web client interface, especially dedicated to analysis methods.

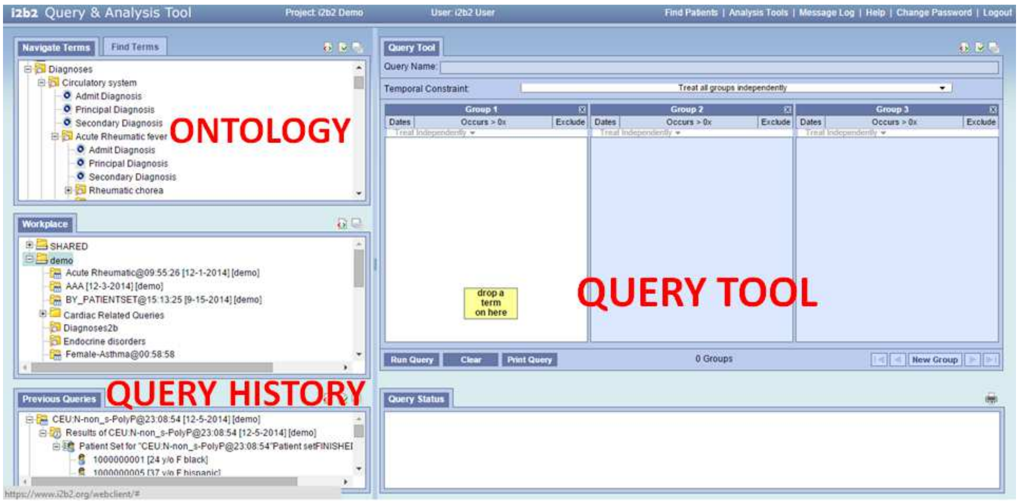


Figure 32.i2b2 web client

3.2.3. Data Annotation

ANNOVAR, described previously in the thesis, has been chosen for its portability and performances in computational time.

Annotation procedure, as well as the other steps, has been implemented by creating a Java application able to manage the events flow.

First VCF files are converted to the ANNOVAR format and subsequently the annotation step is launched by using the *table_annovar* Perl script which requires several input parameters in order to tune data output format

(CSV files) and annotation steps. ANNOVAR uses genomic indexed files within the pointed resource path where it expects to find the mapped resource files within its scripts: the resource path is passed as an input. Figure 33depicts the simplified UML class diagram of the Java application that manages the annotation step.

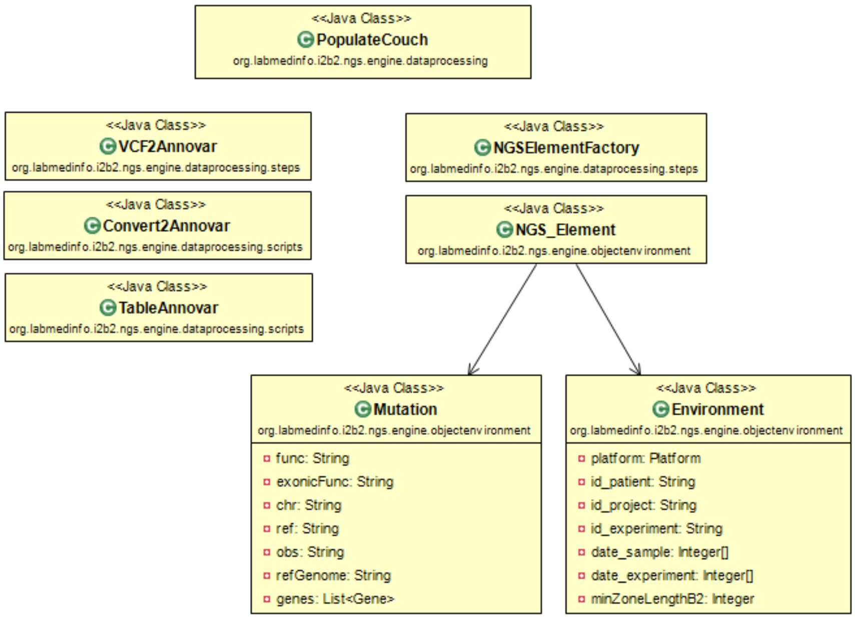


Figure 33. Simplified UML class diagram of the Java application that manages annotation and import steps. (1) Annotation: PopulateCouch class calls VCF2Annovar methods to both convert the VCF file into the ANNOVAR format and run the annotation script table_annovar wrapped into the TableAnnovar class. (2) Import: PopulateCouch class call NGS_Element methods to parse the ANNOVAR output, generate the JSON files and load them into the database. The Mutation and Environment classes hold the attributes of the JSON documents (partially showed).

In order to speed up the overall process, VCF files are split in batches, running an ANNOVAR instances for each. The import step follows the same schema. This procedure has been implemented in the Amazon AWSCloud [166] allowing the use of different virtual machines (EC2 instances)running in a parallel fashion (seeFigure 34).

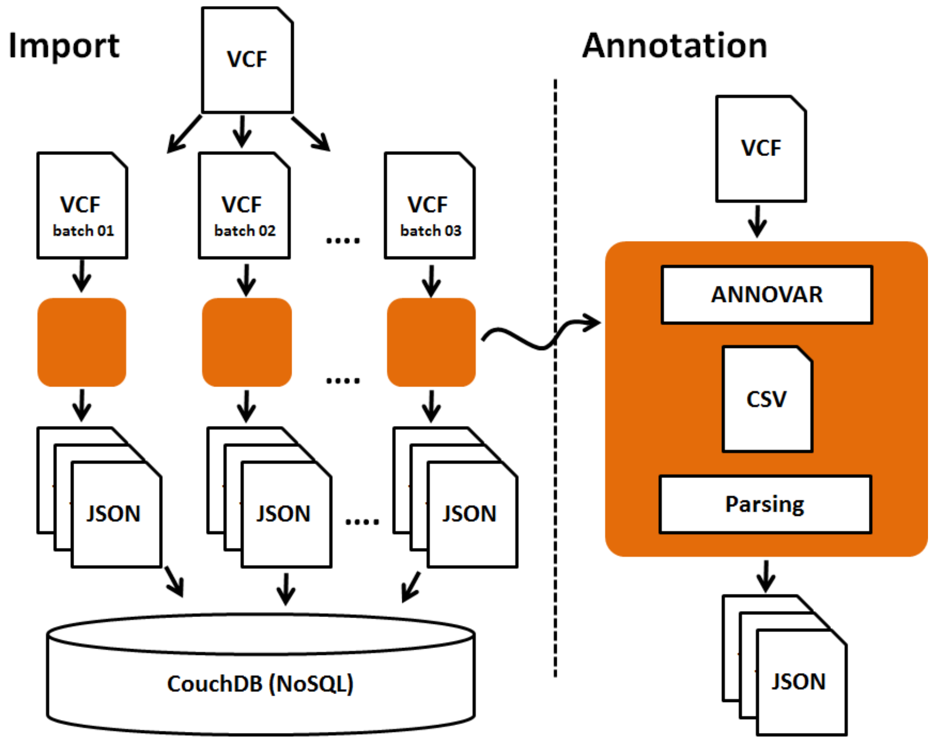


Figure 34. The import within annotation steps to CouchDB database. Orange squares represents EC2 Amazon instances.

3.2.4. Data Import

In order to be imported in CouchDB, first data have to be converted in JSON format. In the import step, each row of the annotated variant file (ANNOVAR output) is therefore transformed into a JSON having predefined attributes (see Table 4) and identified by an UUID. Data are sent to CouchDB database through the Java application that makes use of the LightCouch Java API [167] to convert the Java objects into CouchDB REST methods.

Field Name	Type	Description
chr	String	Chromosome
ref	String	Reference
obs	String	Variant
function	String	Variant function
refGenome	String	Genome assembly
start	Integer	Mutation start position
end	Integer	Mutation end position
gene name	String	Gene symbol (refSeq)
segDup	Double	Sequence identity score for the

		segmental duplication region where variant is located in
AVSIFT	Double	Whole-exome SIFT scores for non-synonymous variants
exonicFunc	String	Exonic variant function
exonic_hgvs_transcript	String	Variant in hgvs format on transcript
exonic_hgvs_protein	String	Variant in hgvs format on protein
exonic_exon	Integer	Exon number where variant is located in
gt	String	Genotype
vcf line	String	Original VCF line
1kgp_freq	Double	Variant frequency for 1KGP
1kgp_version	String	1KGP version
dbSNP_id	String	dbSNP identification id
dbSNP_version	String	dbSNP version
dbESP_freq	Double	Variant frequency for ESP
dbESP_version	String	ESP version
dbClinVar	String	ClinVar accession number
dbClinVar_version	String	ClinVar version
LJB_phyloP_score	Double	Evolutionary conservational score by phyloP
LJB_SIFT_score	Double	SIFT scores for non-synonymous variants
LJB_PolyPhen2_HDIV_score	Double	PolyPhen2 scores for non-synonymous variants (hdiv model)
LJB_PolyPhen2_HDIV_pred	String	PolyPhen2 class for non-synonymous variants (hdiv model)
LJB_mutationTaster_score	Double	MutationTaster scores for non-synonymous variants
LJB_mutationTaster_pred	String	MutationTaster class for non-synonymous variants
LJB_GERP	Double	Evolutionary conservational score by GERP
LJB_PolyPhen2_HVAR_score	Double	PolyPhen2 scores for non-synonymous variants (hvar model)
LJB_PolyPhen2_HVAR_pred	String	PolyPhen2 class for non-synonymous variants (hvar model)
LJB_mutationAssessor_score	Double	MutationAssessor scores for non-synonymous variants

LJB_mutationAssessor_pred	String	MutationAssessor class for non-synonymous variants
LJB_fathm_score	Double	FATHM scores for non-synonymous variants
LJB_siPhy_score	Double	Evolutionary conservational score by Siphy
LJB_ LRT_score	Double	LRT scores for non-synonymous variants
LJB_LRT_pred	String	LRT class for non-synonymous variants
LJB_version	String	dbNSFP version

Table 4. Genomic variant attributes stored in CouchDB

Along with genomic variants, an identification sample code has to be inserted into the JSON as well, corresponding to the i2b2 patient code: generally, sample codes are within the VCF file. Actually, both single and multi-sample VCF files are supported.

Together to JSON files (one for each sample-variant), *designsdocuments* defining *views* are imported into CouchDB as well. We previously described how CouchDB stores data, and how queries are pre-computed as lists of key-values pointing documents by the *map* and *reduce* functions resulting in *views*. In our case we chose to pre-compute views for each JSON attribute(i.e. genomic annotation) in order to have lists of documents grouped and indexed (B-tree storage engine) by the corresponding values.
The rationale is that the user can choose the desired combination of annotation fields to filter patients' genomic variants. The logical AND/OR operations are managed by the application software that communicates with CouchDB and works on UUIDS of the returned lists of documents.

3.2.4.1. A View for Fast Genomic Interval Queries

One of the most useful and stressed query in Genomics consists in retrieve the genomic features overlapping a given chromosome interval. In section 2.4.1.1 we have showed how the UCSC Genome Browser binning scheme allows combining pre-defined genomic regions (bins) as attributes of the genomic features and B-tree indexes to guarantee a fast retrieval by range queries.

We wanted to replicate a similar approach in CouchDB, enabling genomic interval queries across all the stored genomic sample variants.

Each chromosome has been divided into a predefined set of hierarchical bins (tree) depending on the specific chromosome length. The value 0 or 1 has been assigned to each bin, depending on being the left or right child bin within the tree, respectively (with the exception of the root bin, coded by 0). The ordered combination of 0s and 1s is then assigned to each genomic features, thus allowing navigating the binning tree from the root to the smallest bin entirely containing the genomic feature. In Figure 35, for example, for the genomic feature A, the smallest containing bin is the one reached by navigating the tree in the following way (red numbers): 0,0,1,0,0. Feature B, instead, is contained into the 0,1 bin and despite it could be contained, giving its dimension, into a smaller bin, it overlaps to lower level bin borders, therefore it is assigned to the higher level one.

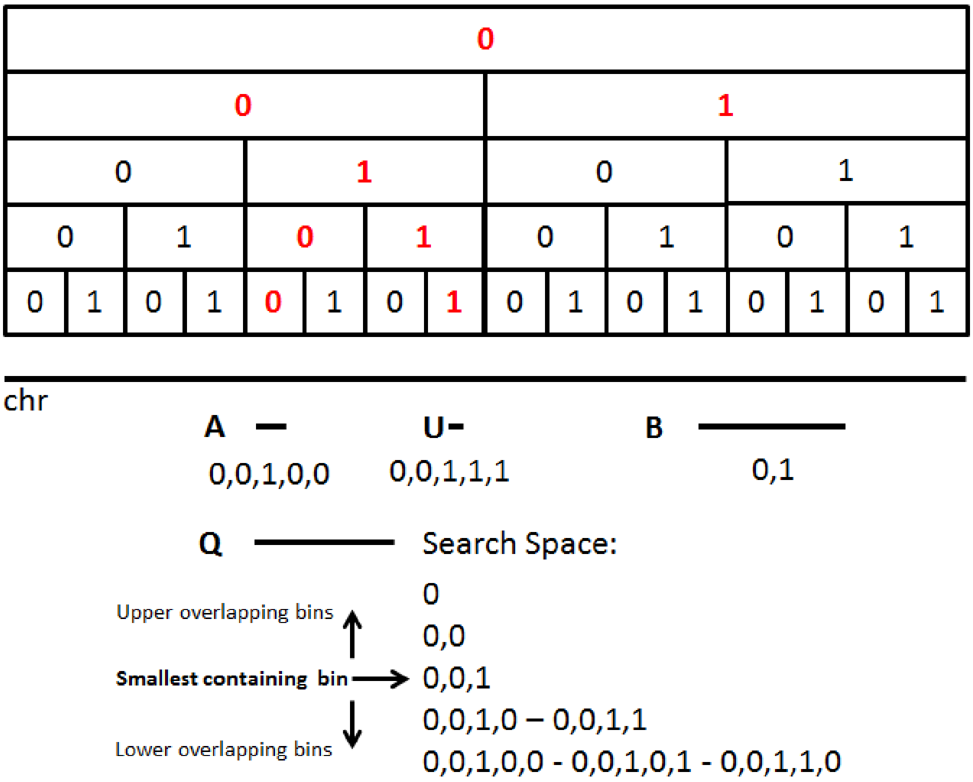


Figure 35.Binning scheme representation implemented in CouchDB

The smallest bin containing each genomic variant is computed, giving its genomic coordinates, and stored as an attribute in the JSON file as an array of 0s and 1s.

It is therefore possible to write a *map* function in CouchDB to pre-compute a view having as row keys the following elements in this order: *sample*, *chromosome* and *bin*. CouchDB will store the sorted keys into the B-tree for the logarithmic search.

Given an interval query, the smallest containing bin is calculated and range queries on the created view retrieve all the genomic features mapping on the computed bin and its pre-calculated overlapping ones. For example, in Figure 35 the interval query Q is entirely contained into the 0,0,1 bin. We can pre-calculate the search space consisting of the overlapping bins both for the upper part of the binning tree (parent bins) and those in the lower one (child bins). The view is therefore searched for the genomic features (documents associated to the view keys) within these bins.

Actually, this query would give back genomic features overlapping to the search binning space, but not necessarily to the interval search of interest (that is a sub-segment of its smallest containing bin). For example, as reported in Figure 35, the result of the query Q would report also the unwanted genomic feature U: in fact, despite it overlaps to one of the searched bin, it does not overlap to Q. Therefore other two queries (views) are performed in order to remove the non-overlapping elements: the first add the *start* position of the genomic feature to the view keys (*sample*, *chromosome*, *bin*, *start*)while the second add the *stop* position. In the showed example, genomic feature U would be removed from the query result set because its start position is greater than Q stop one.

3.2.5. Data Retrieval

In order to build, perform and show query results, two software components have been developed and integrated into the i2b2 framework: an i2b2 cell, called NoSQL-NGS, and an i2b2 plug-in, called BigQ-NGS.

Figure 36 shows the overall system and the aforementioned interacting components.

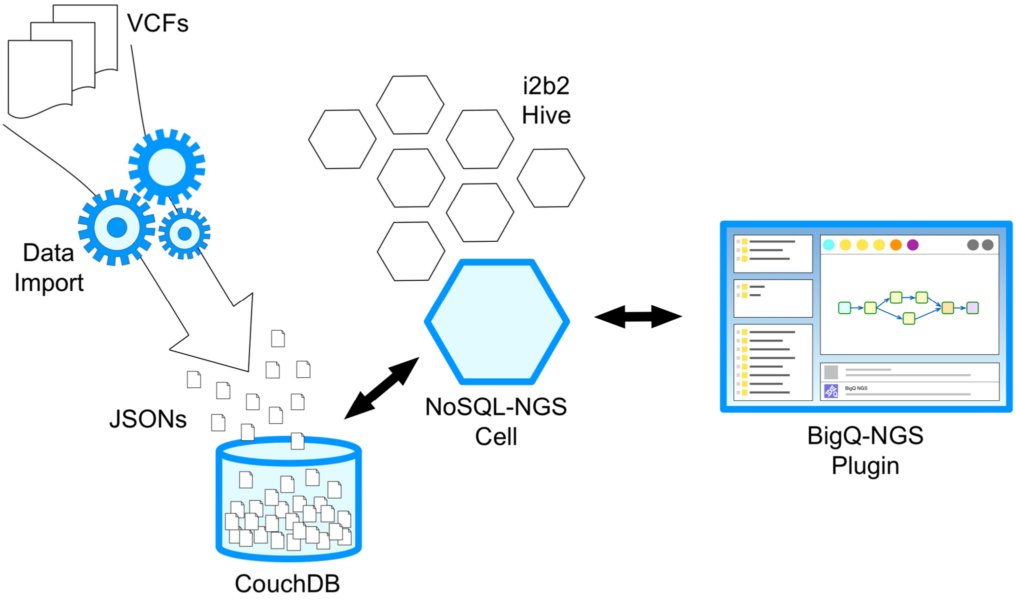
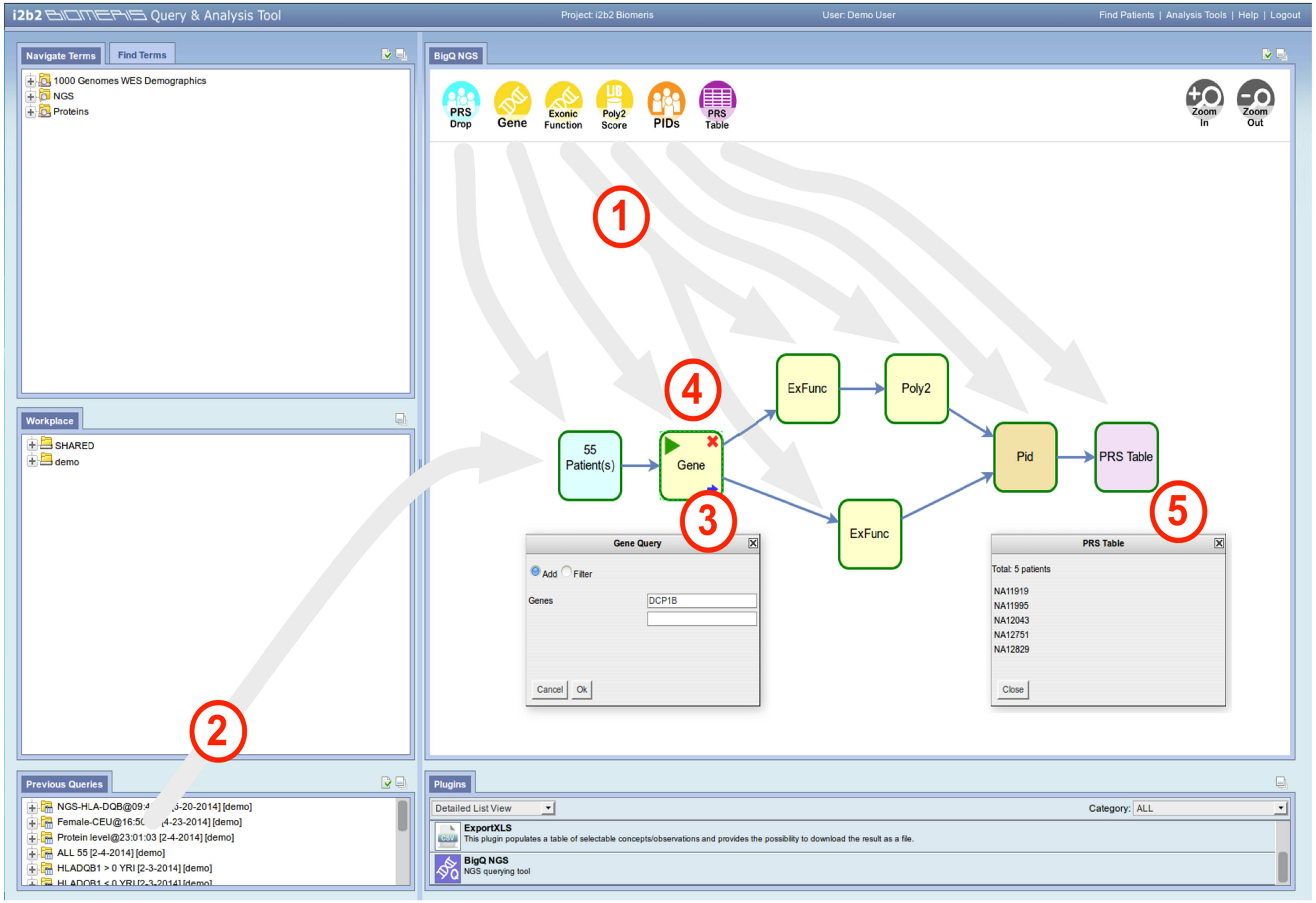


Figure 36. System overview showing interacting components, from data import to data query.

First, the user, thanks to the i2b2 webclient, performs a query on the i2b2 CRC by setting constraints on Ontology terms and using the Query Tool (see 3.2.2). Once the patient cohort has been retrieved, the user selects the BigQ-NGS plug-in and builds up the query in the Plugin Viewer, basing on available genomic data annotations respect to the selected patient cohort. The BigQ-NGS plugin interface is based on visual programming (mxGraph Javascript libraries [168]) and the user can graphically build queries with drag and drop interactions. Genomic annotation features are represented as graphical blocks and can be linked together in order to create the filtering procedure used to query genomic variants: AND and OR operations are implemented by linking blocks in series and in parallel, respectively.

Referring to Figure 37, query is made up by the following operations: (1) the user drags and drops blocks inside the plugin's workspace. Blocks are connected to each other to define the query. (2) The patient set is dragged and dropped on the Patient Result Set Drop (PRS Drop) block. (3) each query block (in yellow) holds parameters to set query logic and attribute value constraints. (4) when run, each block executes its query sequentially, calling the NoSQL-NGS cell. (5) when all blocks have performed their query, the user can visualize the results by double-clicking the Patient Result Set Table (PRS Table) block, consisting, in the reported example, of patient codes having at least one genomic variant matching the query.



NGS-HLA-DQB@09:4 -20-2014] [demo]
Female-CEU@16:5 4-23-2014] [demo]
Protein level@23:01:03 [2-4-2014] [demo]
ALL 55 [2-4-2014] [demo]
HLADQB1 > 0 YRI [2-3-2014] [demo]
HLADQB1 < 0 YRI [2-3-2014] [demo]

Gene Query

● Add ○ Filter

Genes: DCP1B

PRS Table

Total: 5 patients

NA11919
NA11995
NA12043
NA12751
NA12829

Category: ALL

ExportXLS: This plugin populates a table of selectable concepts/observations and provides the possibility to download the result as a file.

BigQ NGS: NGS querying tool

Figure 37. The BigQ-NGS plug-in with user interactions highlighted.

The i2b2 cell has been developed to communicate with CouchDB and execute queries on genomic variants passed by the BigQ-NGS plug-in. The i2b2 cell has been written in Java and uses LightCouch API as well as the import application.

Each block of the BigQ-NGS plugin performs a query to the NoSQL-NGS cell by sending it an i2b2 XML-based message. The cell extracts all parameters required to run the query:

- *dataIn:* the basic object exchanged between the plug-in and the cell consisting in a set of documents UUIDs grouped by patient code

- *query logic:* two mutually exclusively operations are allowed, *add* and *filter*. The first represents the operation of union, therefore if *add* is chosen, the UUIDs returned by the CouchDB are added to *dataIn* and sent back in the cell response to the plug-in; the second in an intersection operation; therefore when *filter* is chosen, only the UUIDs belonging to both sets are returned.

- *query type*: that identifies the variant fields (see Table 1) on which the query should be executed. Examples of allowed query types are: gene for gene names, exonicFunc for exonic functions and PolyPhenScore for LJB PolyPhen2 score.

- *query details*:the set of values required to perform the specific type of query. For example: the list of gene names for the gene query orthe PolyPhen-2 score interval endpoints for the PolyPhen query.

Once these parameters are extracted, the NoSQL-NGS cell accesses the CouchDB *view* associated with the specific query type according to the query details; this operation is performed for each patient in the *dataIn* set. Results from database, consisting in a new set of UUIDs (genomic variants) grouped by patient, are combined with *dataIn* according to the query logic (*add* or *filter*) to build the output of the cell, called *dataOut*.
Finally, the NoSQL-NGS cell builds the response XML message encoding the *dataOut* object and sends it back to the client.

3.2.6. Results and Discussion

To test our approach for integrating genetic queries within the i2b2 framework, we have performed a “stress test” on the system by submitting increasingly large Whole Exome Sequencing (WES) datasets on which we performed the same two queries: a simple one (retrieving patients with a given variant in dbSNP) and a more structured one (querying by the combination of a given gene name, exonic functions values and PolyPhen2 score). For each test we have measured the size (in GB) of the database with its views, and the average time necessary to run the two queries in particular (see Table 5).

All tests were performed on a single AWS EC2machine: in particular we have used a *c3.2xlarge* machine[169], a medium-high level server with 8 virtual CPUs and 15GB of memory. Regarding import phase (ANNOVAR + JSON conversion),which is the most computational demanding one, we used 6 *m1.large* EC2 machines.

WES data were retrieved from the 1000 Genomes Project phase1 integrated release[170]. We have tested our system on sets containing variants coming from 10, 20, 50, 100, 200 and 500 exomes. The average number of variants of the sequenced exomes, and thereby of the JSON documents added to the database for each case, is about 23,000.
A single exome, in average, has been imported in about 4 minutes and 50 seconds, i.e. less than 13 ms per single genomic variant.

The first query aimed at extracting those patients having a particular mutation (*rs1805009*) associated with red hair and poor tanning ability[171]. The goal of the second query was to identify patients that have nonsense or probably damaging (according to PolyPhen2) missense mutation in the DCP1B gene, which is known to be related to pancreatic cancer[172].

Table 5 and Figure 38show the results obtained, indicating that the query time is independent of the size of the database in the case of the simple query 1, while it linearly scales with the size of the database in query 2. It is interesting to note that, with the proposed computational infrastructure, the query time is almost instantaneous for the user in the case of query 1 (about 0.06 seconds), while the time to query 500 genomes and more than 11 million variants is only about 34 seconds.

# Exomes	Size (GB)	Tquery1 (ms)	Tquery2 (ms)
10	1,1	669	3440,2
20	2,5	678,8	3647,2
50	7,9	554,4	6462
100	19	680,4	9753,2
200	48	691	15595,8
500	160	678,4	34836,8

Table 5. Results obtained on sets containing variants coming from 10, 20, 50, 100, 200 and 500 exomes.

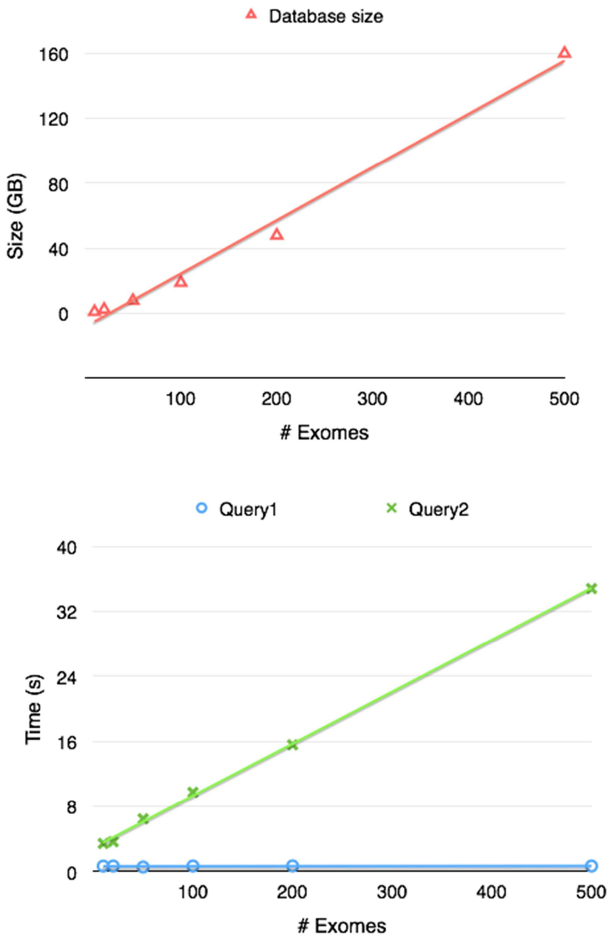


Figure 38. Results obtained on sets containing variants coming from 10, 20, 50, 100, 200 and 500 exomes.

Combining NoSQL document database and the i2b2 platform holds all the premises to be a winning strategy both in the management of genomic annotated variants and in being the ideal knowledge data source to exploit new correlations between patients phenotypes and genotypes.

The system has been conceived to deal with variants of unknown clinical meaning. For this reason, the data model is flexible, and reflects the contents of ANNOVAR documents; the database can thus be easily updated with new versions of the variant annotations. In fact, because of the high performances showed, it results more practical to re-annotate and re-import all genomic sample variants, even if a single genomic annotation field had to be updated or added, rather than try to update JSON documents for a single or few attributes. Moreover, the computational time during import phase can be further reduced by horizontally scaling: being genomic variants treated independently both by annotation and importing including view pre-computation (at least for *map* functions), parallelization is straightforward.

Indeed, the query system has very promising performance, showing to scale well with the database volume, making it feasible to jointly query clinical and genetic data. We note that the choice of CouchDB allows naturally relying on cloud-based implementations on elastic clusters, such as the BigCouch system [173].

In the future would be interesting to compare our implementation to other state of the art extensions of i2b2 and TRANSMART[174]developed to deal with NGS data.

Finally, several pending points have still to be addressed in order to complete the whole picture:

- Update process (versioning): any document change in CouchDB is tracked by the intrinsic document versioning, which is an additional default attribute to the JSON. Imagine that the new version of a genomic database of interest comes up and we would desire to update JSON documents for it. To completely re-annotate every variant into the database for every annotation field is the most practicable solution. Two ways may be followed in order to track data updates: relies on CouchDB versioning by updating the same document; create new documents with an updated and custom version number as JSON attribute. The latter would result in a duplication of the total number of documents in the database for each update and in the inclusion of the versioning attribute within each view (queries should point, by default, to the last update version). Nonetheless, this should be straightforward to implement. The second would use CouchDB versioning system, but is expected to be more difficult, because it requires tracking the JSON UUID during re-annotation process in order to be able to associate the very same re-annotated genomic variant to the original document.

- Controls cohort with aggregate results: right now there is no possibility yet to perform a case control matching at genomic variant level as we have seen in the RDBVMS. Reduce functions in CouchDB allows us to aggregate data such as averages on views etc. but to pre-compute all the possible combination of patient cohorts to be used as controls would results in a combinatorial explosion; therefore, these kinds of data aggregations need necessarily to be performed on the fly. We can therefore imagine to have a Patient Result Set Drop block for a patient controls cohort as well, whose genomic variants follow the same filtering procedure of the patient case cohort: each document resulting from the final list of the case cohort would be searched within a *view* given by a *map* function emitting (*sample*, *chromosome*, *start*, *stop*, *ref*, *obs*), which univocally identifies a genomic sample variant, and where *sample* is set for each sample of the controls cohort.

- Patient code mapping: VCF provides the patient code coming from the sequencing analysis pipeline. Within the presented performed tests, the i2b2 and the VCF codes were the same. However, we expect this correspondence to lack in the major part of cases; therefore an internal mapping between VCF patient codes and i2b2 ones should be present and converted codes should be passed to the BigQ-NGS plugin.

- Plug-in enhanced functionalities: BigQ-NGS plugin could be improved by adding new features such as :i) the possibility to save the performed queries, ii) an expanded genomic annotation attributes set through which query variants, iii) the possibility to build query exploiting external resources (such as list of genes coming from an user file or specific pathway-related gene list coming from KEGG database).

Chapter 4

Sequencing data interpretation

In this Chapter a new algorithm for the prediction of genomic variants pathogenicity is presented.
In 2.5.2 and Chapter 3 we have discussed the variant prediction tools principles and how their results are utilized as genomic annotation in order to prioritize variants according to their prediction and score.
Hereby we present PaPI, our developed algorithm that makes use of pseudo amino acid composition to score human-protein coding variants.

The Chapter is based on the following paper, at the moment under review:

Limongelli I, Marini S, Bellazzi R, PaPI: pseudo amino acid composition to score human protein-coding variants. *BMC Bioinformatics, under review.*

4.1. Variant prediction by pseudo amino acid composition: PaPI

PaPI is a new machine-learning approach to classify and score human coding variants by estimating the probability to damage their protein-related function. The novelty of this approach consists in using pseudo amino acid composition through which wild and mutated protein sequences are represented in a discrete model. A machine learning classifier has been trained on a set of known deleterious and benign coding variants with the aim to score unobserved variants by taking into account hidden sequence patterns in human genome potentially leading to diseases. We show how the combination of amphiphilic pseudo amino acid composition, evolutionary conservation and homologous proteins based methods outperforms several prediction algorithms and it is also able to score complex variants such as deletions, insertions and indels.

A freely available web application (http://papi.unipv.it) has been developed with the presented method, able to score up to thousands variants in a single run.

4.2. Background and Rationale

Variant predictors are generally based on four different approaches: multiple sequence alignment (MSA) methods of homologous proteins[125, 175]such as SIFT [8], protein structure information such as PolyPhen2[2], comparative evolutionary data[128-130]and structural or sequence pattern encoding[176]. Since each algorithm has some limitations, one approach to detect a deleterious variant consists of testing several independent methods and checking if at least one assesses its pathogenicity[177, 178]. This strategy has high sensitivity, but poor specificity, thus leading to low accuracy. Therefore, a number of algorithms that combine the outputs of several predictors and optimize accuracy on known variant sets have been developed[179-181].Moreover, methods that use prior knowledge (e.g. Human Phenotype Ontology, Gene Ontology) in combination with functional predictions in order to rank variants on the basis of a given phenotype[112, 118]have been successfully implemented, as well.

In this scenario, due to the importance of having more accurate and exhaustive variant functional predictors, we developed a new phenotype-free method based on pseudo amino acid composition (PseAAC)[6] and evolutionary conservation in combination with other two well-established and commonly-used approaches (PolyPhen2 and SIFT). We believe that our approach may provide a valuable addition to the worldwide research efforts devoted to predicting the role of uncharacterized variants.

PseAAC is a feature encoding method allowing both compositional and positional amino acid pattern representation of peptide primary sequence in a discrete model. Given a peptide sequence, PseAAC is computed by modeling pairwise relationships between amino acids using residues chemical properties. In particular, we used amphiphilic PseAAC, based on normalized hydrophobicity and hydrophilicity: the arrangement of these two indices along a protein chain play an important role in protein folding, catalytic mechanism and protein interaction with other molecules and environment[182].For example, hydrophobicity is often a major contributor of binding affinity between a protein and its ligand[183], hydrophilic residues such as Arg, Asp, Lys, and Glu have the highest protein-surface frequencies[184], and intrinsically disordered regions (IDRs) usually have few large hydrophobic residues but favor polar and charged amino acids[185].

Previous studies[186, 187]analyzed human coding variants in terms of amino acid substitution both in disease and natural background variant datasets, such as Human Gene Mutation Database [188] and 1000 Genomes Project (1TGP). Such studies showed that disease-associated variant distributions are radically different from neutral amino acid ones and that disease-associated variants exhibit more extreme differences in terms of physicochemical properties such as amino acid volume, charge and hydrophobicity.

We therefore coupled hydrophobicity and hydrophilicity PseAAC feature encoding with machine learning to develop a model able to learn pseudo amino acid composition substitution patterns following coding variants that can alter protein function and/or structure, leading to disease.

The difference in terms of PseAAC between wild and mutated protein sequences together with evolutionary conservation scores of the altered bases have been used as features to train a Random Forest (RF)[7]with the aim to score coding variants into protein-damaging or tolerated class. Since PseAACs model amino acid relationships in terms of hydrophobicity and hydrophilicity arrangements within the wild and mutated sequences, RF is supposed to learn from substitution patterns occurring at amino acid composition level in terms of frequency and order. A variant is therefore implicitly evaluated within its sequence context.

We finally combined the RF output with PolyPhen2 and SIFT by a voting strategy. Despite the advantages of combining PolyPhen2 and SIFT have been previously reported[181], we show the RF inclusion is able to further increase prediction performances.

The overall algorithm, called PaPI, provides predictions even for those variants that the other tools cannot process (e.g. because of lack of data) and it is able to deal with any variant type, including single nucleotide variants and insertion or deletion of several nucleotides.

While RF classifiers have been already used in Genomics, from GWAS to RNA-protein binding prediction [189], to our knowledge, this is the first time that PseAAC is applied to protein variant prediction.

4.3. Methods

Hereby we denote with the term *indel* the following variants: insertions, deletions, insertions followed by deletions (or vice-versa) and multi-nucleotide variants. We refer to single nucleotide variants (*SNVs*) in case of non-synonymous single nucleotide variants that lead to a single amino acid change. Finally, we denote as *in-frame* and *frameshift indels* those variants causing the insertion/deletion of one or more amino acids and those altering the open reading frame of the coding sequence, respectively.

PaPI is an ensemble classifier consisting of a voting scheme that includes a RF classifier, PolyPhen2 and SIFT. The RF model have been trained on PseAAC differences of mutated and wild protein sequences, evolutionary conservation scores and several full-length protein attributes. Figure 39depicts the workflow through which a new variant is classified and assigned a score, representing its risk of being protein damaging.

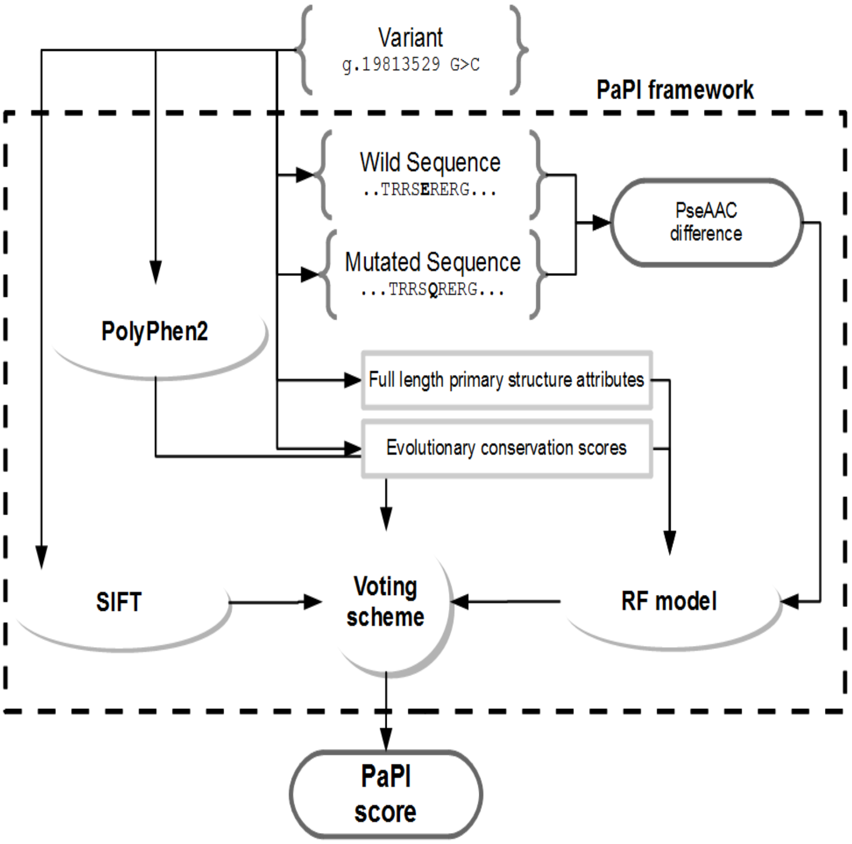


Figure 39. Feature encoding scheme. A genomic variant is translated into wild and mutated amino acid sequences. The difference in terms of PseAAC features is computed and is given as input to the trained RF model along with evolutionary conservation scores and several full-length protein attributes. PolyPhen2, SIFT and RF results are finally combined together to obtain the final PaPI class and score.

4.3.1. Psuedo Amino Acid Composition

An amino acid sequence can be represented by a set of discrete numbers mapping the patterns of its amino acid physico-chemical properties into a fixed number of features.

Traditional amino acid composition approach has been widely used in predicting protein structural class[190, 191]and it merely records amino acids frequencies in a protein sequence.

PseAAC adds a number of position-related features and therefore it reflects both compositional and sequential order. We utilized, in particular, amphiphilic PseAAC, based on normalized hydrophobicity and hydrophilicity[182].

In brief, given a protein sequence *P* with *L* amino acid residues:

$$P=A_1\,A_2\,A_3\,A_4\,A_5\,A_6\ldots A_L$$

it is possible to convert it into a finite set of number *P*

$$P'=\{p_1, p_2, p_3, p_4, p_5,\ldots, p_{20}, p_{20+1},\ldots, p_{20+2\lambda}\}$$

where the first 20 numbers are functions of the frequencies of the 20 amino acids within *P* and the remaining 2λ are a set of correlation factors that reflect different hydrophobicity and hydrophilicity distribution patterns along a protein chain. Correlation factors are given by coupling the most contiguous residues whose contiguity condition varies according the considered tier (see Figure 40).

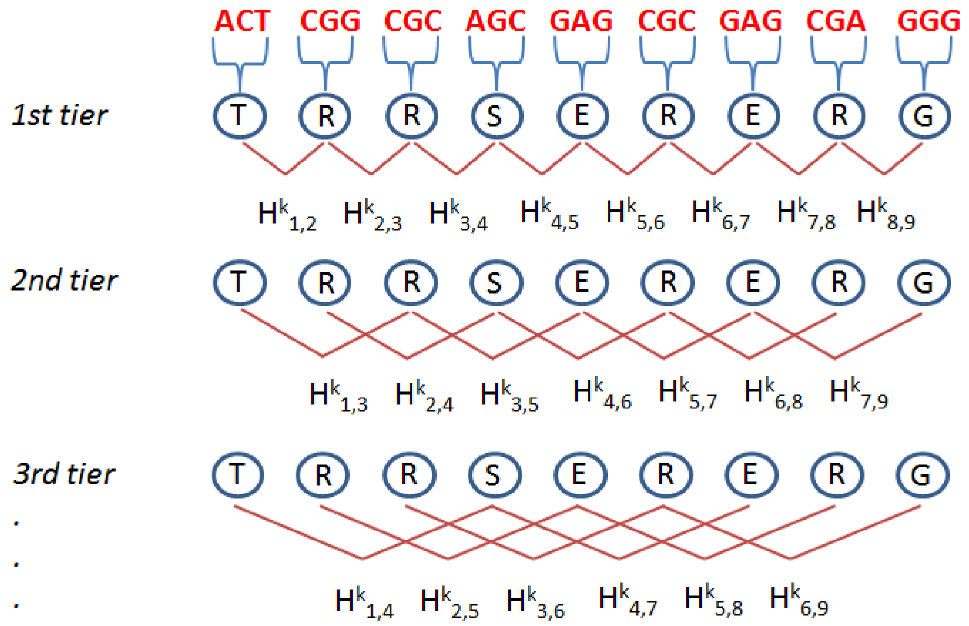

ACT CGG CGC AGC GAG CGC GAG CGA GGG
1st tier
T R R S E R E R G
$H^k_{1,2}$ $H^k_{2,3}$ $H^k_{3,4}$ $H^k_{4,5}$ $H^k_{5,6}$ $H^k_{6,7}$ $H^k_{7,8}$ $H^k_{8,9}$
2nd tier
T R R S E R E R G
$H^k_{1,3}$ $H^k_{2,4}$ $H^k_{3,5}$ $H^k_{4,6}$ $H^k_{5,7}$ $H^k_{6,8}$ $H^k_{7,9}$
3rd tier
T R R S E R E R G
$H^k_{1,4}$ $H^k_{2,5}$ $H^k_{3,6}$ $H^k_{4,7}$ $H^k_{5,8}$ $H^k_{6,9}$

Figure 40. Amphiphilic PseAAC representation. This is a diagram shows how the correlation factors Hk , based on amino acid hydrophobicity (k=1) and hydrophilicity (k=2), vary in each tier by coupling residues at different distances.

The maximum number of tiers corresponds to λ. Coupling is then given by the hydrophobicity and hydrophilicity correlation functions.

$$H^1_{i,j} = h^1(A_i) \cdot h^1(A_j)\ ,\ H^2_{i,j} = h^2(A_i) \cdot h^2(A_j) \qquad (1)$$

where h1(Ai) and h2(Ai) are, respectively, the hydrophobicity and hydrophilicity values for the ith (i=1,2,…, L) amino acid in P. Correlation functions are summed over each λ-tier and the 20+2λ coupling factors are given

$$p_u = \begin{cases} \dfrac{f_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{2\lambda} \tau_j}, 1 \le u \ \le 20 \\ \dfrac{w\tau_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{2\lambda} \tau_j}, 20 + 1 \ \le u \ \le 20 + 2\lambda \end{cases} \qquad (2)$$

where f_i are the normalized frequencies of the possible 20 amino acids in *P*, τ_j is the sum of the j-tier correlation functions and *w* is a weight factor.

4.3.2. Feature Set

The features utilized for RF and LR training can be divided into three groups: (a) PseAAC, (b) full-length primary sequence attributes, and (c) evolutionary conservation scores. The three feature groups are described as follows.

4.3.2.1. PseAAC

Given a genomic variant overlapping a protein, we first generated the altered protein sequence in according to the coding frame, we then considered the 20 amino acid residues upstream and downstream the first mutated amino acid forming a snippet of 41 amino acid residues. The same procedure is followed in the case of the corresponding wild type protein sequence. Amphiphilic PseAAC is then computed by PseAAC-builder[192]for both wild and mutated snippets. The variant-sequence features are finally encoded as the element-wise difference of wild and mutated PseAAC vectors (see Figure 41).

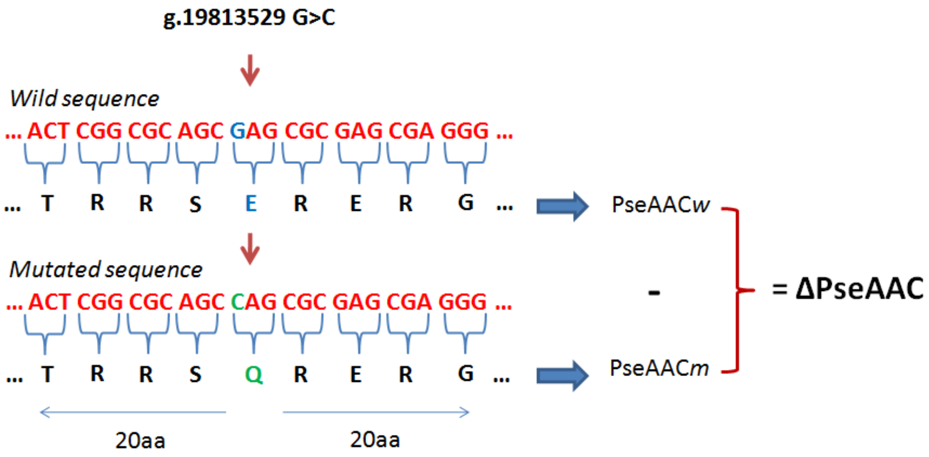

g.19813529 G>C
Wild sequence
... ACT CGG CGC AGC GAG CGC GAG CGA GGG ...
... T R R S E R E R G ...
PseAACw
Mutated sequence
... ACT CGG CGC AGC CAG CGC GAG CGA GGG ...
... T R R S Q R E R G ...
PseAACm
-
= ΔPseAAC
20aa
20aa

Figure 41. Example of PseAAC variant feature encoding. A genomic variant is translated into the relative wild and mutated amino acid sequences. PseAAC for both wild and mutated protein snippets are computed and the differences between each PseAAC term makes the PseAAC feature set.

Note that even if the method allows theoretically dealing with amino acid sequence changes of any length, only insertions/deletions up to 20 amino acids (60 nucleotides) were considered for PseAAC model training.

We chose two 20 amino acid flanking regions for two reasons. First, since the features are encoded by PseAAC differences, considering large sequence portions (e.g. the whole primary structure) could introduce noise and dilute the PseAAC difference information content, especially in case of single amino acid substitution, where both positional and composition change would be minimal. Second, we considered that protein short functional regions, such as short linear motifs, which play a pivotal role in protein interactions, range from 3 to 11 amino acids in length [193].Changes in their flanking regions could severely alter the protein function as well[194, 195].We therefore assumed that 20 amino acids constitute a reasonable window size to encompass possible short functional motifs and their flanking regions.

4.3.2.2. Full Length Primary Sequence Attributes

We included in the RF model three features related to variant position and protein length. First, we considered the difference and the ratio between mutated and wild amino acid sequence lengths. In other words, we measured the number of possible lost/inserted amino acids caused by the variant. Second, we considered the position of the variant in the amino acid sequence normalized by the protein length (e.g. 0.9 for a 100 amino acids long protein and its mutated amino acid at the 90th position). This feature reflects the fact that some kind of variants affecting the initial part of the primary sequence may have a huge damaging potential for the whole protein (e.g. a stop-causing variant).

4.3.2.3. Evolutionary Conservation Scores

GERP++[128], PhyloP [129] and Siphy[130]were chosen because they apply different and complementary methods to weight nucleotide conservation among different species. In case of indels, the following policy was adopted: in case of deletion we took the highest score among the deleted nucleotide bases; in case of insertion we took the highest score between the two reference bases where insertion occurs.

4.3.3. Data Set

We obtained positive (damaging) variants from the HGMD (updated to May 2013). Variants were annotated by ANNOVAR using the RefSeq gene model. All non-coding variants, as well as variants reported with a frequency higher than 5% in the total of 1092 samples from 1TGP (April 2012 release) were filtered out. Negative (tolerated) variants were extracted from the aforementioned release of 1TGP and from ESP (6500si release) retaining only variant at frequency higher than 0.05. Non-coding and synonymous variants were filtered out by ANNOVAR. Each variant was then processed by the PaPI annotation framework in order to build the relative feature set. The original variant dataset consisted of 204021 coding SNVs and indels, distinguished by transcript and filtered out for synonymous SNVs. Stratification for descriptive protein alteration to the primary structure unveiled a significant proportion (about 44%) of frameshift indels or stop-causing/disrupting variants in the damaging set in comparison to the tolerated one (about 3%), as reported inTable 6.

	Damaging (HGMD)	**Tolerated** (1TGP + ESP)
Initial variants	176523	65377
- SYN	1333	36546
- FR/SC/SD	77627	929
= Final variants	**97653**	**27902**

Table 6. Damaging and Tolerated variant sets. Damaging and tolerated sets after synonymous-SNVs and frameshift, stop-causing and stop-disrupting variants removal. An instance of the data

set is the coding variant relative to the transcript to which overlaps. SYN = synonymous, FR = frameshift, SC = stop-causing, SD = stop-disrupting.

One can suppose these variants should be treated as deleterious a priori: the proportion showed above is in accordance with this hypothesis. Including these variants in our data set would introduce a severe classification bias, due to the aforementioned disproportion. Therefore we randomly assembled three quasi-balanced training (70%) and test (30%) sets (see Table 7) without considering these types of variants for the evaluation and comparison steps, but we trained the final RF model (available online) on the whole unfiltered dataset.

Set	Training		Test	
	Damaging	Tolerated	Damaging	Tolerated
1	25291	19570	10729	8332
2	25318	19570	10861	8332
3	17838	13763	7616	5879

Table 7. Three random variant sets. Three quasi-balanced variant sets were generated randomly and divided by training (70%) and test (30%) sets.

Indeed, PaPI is capable to score stop-causing/disrupting and frameshift variants as well. The three test sets have been used to measure the performances of the RF and LR (see4.4).

In order to compare RF, PolyPhen2, SIFT and PaPI (RF + PolyPhen2 + SIFT) on the three test sets we further filtered out the variants that PolyPhen2 and/or SIFT were not able to classify(see Table 8).

Set	Damaging	Tolerated	All
1	5316	5153	10469
2	5323	5153	10476
3	3763	3642	7405

Table 8. The three filtered variant set used to measure performances of RF, PolyPhen2, SIFT and PaPI. Test sets have been divided by Tolerated and Damaging set. Variants on multiple transcripts have been counted once.

4.3.3.1. Preparing the comparison

PolyPhen2, SIFT and the other predictors that have been compared to PaPI (see 4.4.2)only classify SNVs. Furthermore, these tools may be unable to provide any prediction for lack of information (e.g. when only few homologous sequences exist or remain after their filtering). To avoid any bias that could favor PaPI, we removed from the aforementioned test sets all the variants that other algorithms were unable to score (Table 9).

Set	Damaging	Tolerated	All
1	5189	3631	8820
2	5105	3631	8736
3	3618	2553	6171

Table 9. Filtered test sets. The three filtered-variant set used for comparison with PolyPhen2, SIFT, Carol, PROVEAN, FATHMM, MutationAssessor and LRT. Test sets are divided by Tolerated and Damaging set.

The whole data set filtering and processing workflow is shown inFigure 42. Note that we grouped all the different transcript-variants for each variant in the same set, i.e. all the mutated protein isoforms for a variant were either all in the training or in the test set. This procedure assured that very similar instances were not present in both training and test sets.

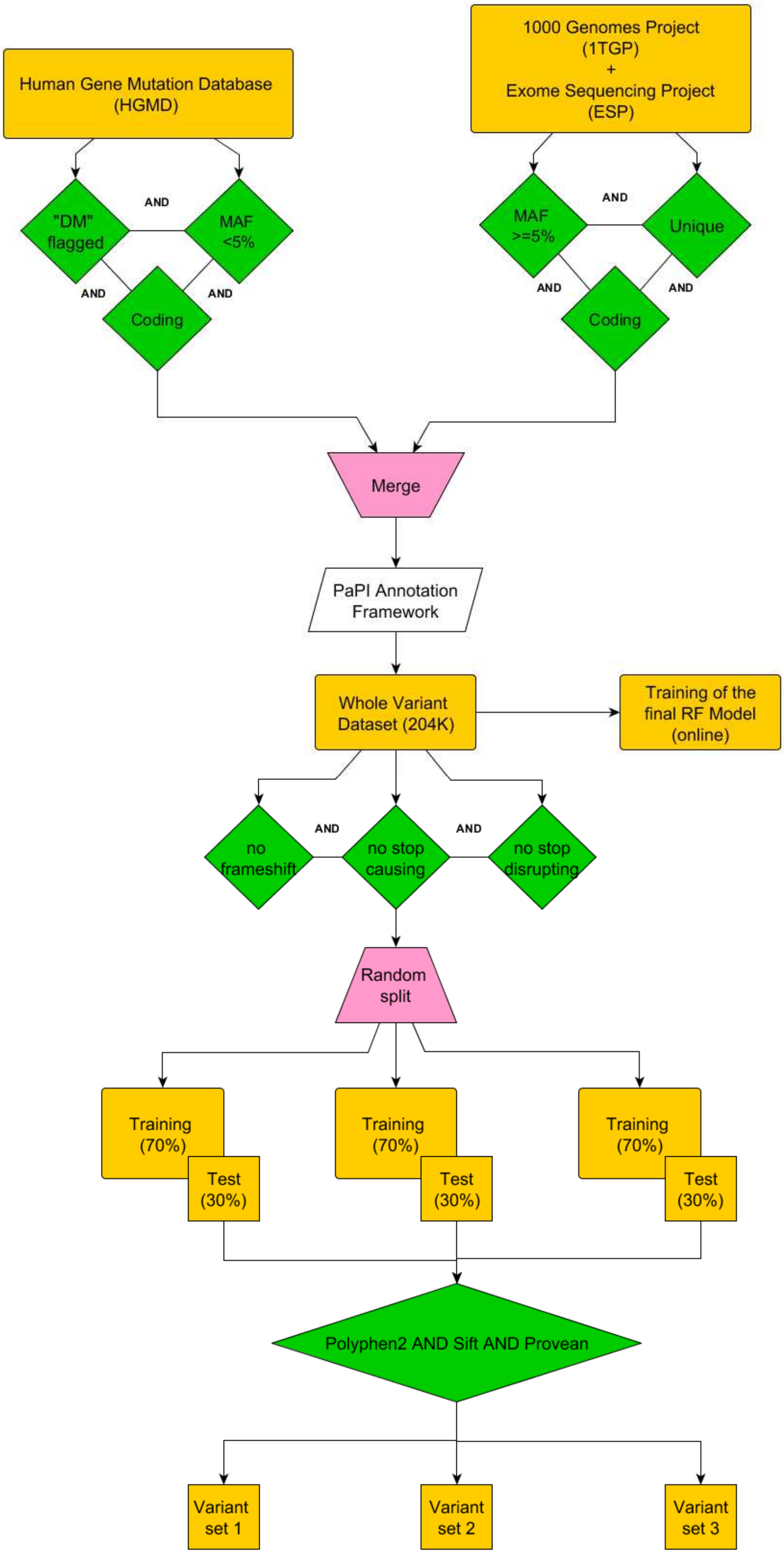


Figure 42. Data set workflow. Workflow representing the data set selection. Variants from HGMD, 1TGP and ESP were filtered basing on coding, frequency and non-overlapping (unique) variants

among the different data sources. In order to evaluate and compare the performances of the variant predictor tools, variants were further filtered for frameshift, stop-disrupting, stop-causing and variants not predictable for the other algorithms.

4.3.4. Voting Scheme

The RF model score is computed as the posterior probability of the class. For each instance, the RF model will provide a probability score for damaging class and its complement to one for the tolerated class. The instance is thus considered damaging if the related score is equal or larger than 0.5, and a tolerated variant otherwise.

SIFT and PolyPhen2 provide scores in the (0,1) interval, and the thresholds t_s to separate damaging and tolerated variants are 0.447 and 0.05 respectively. We needed to standardize both SIFT and PolyPhen2 scores in order to compare them with our RF model score. We thus remapped SIFT and PolyPhen2 results by forcing scores $<t_s$ in the (0, 0.5) interval, and scores $>t_s$ in the (0.5, 1) interval according the following standardization

$$A=((A'-\min(A'))/(\max(A')-\min(A')))*(\max(A)-\min(A))+\min(A) \qquad (3)$$

Where A’ is the score in the original interval and A is the score mapped to the new interval, while min/max(A) and min/max(A’) are the minimum and maximum scores of the new and original interval, respectively.

A majority voting scheme is then applied when each of the three models provides a prediction. That is, in case of conflict between two tools, the vote on class prediction of the third is determinant for the final class assignment (damaging or tolerated). The normalized score of the most confident tool (distance from decision threshold) is taken as the final score. If PolyPhen2 or SIFT are not able to provide a prediction, the most confident normalized score between the remaining two algorithms leads class and score assignment. Finally, in case both PolyPhen2 and SIFT are not able to provide a prediction, only the RF model is used.

Usually the more tools are combined, the smaller is the number of the cases that all of them can predict [196]. However, PaPI is not affected by this limitation since the RF model and the policy used allow obtaining a prediction even when PolyPhen2 and/or SIFT do not.

4.3.5. PaPI Annotation Framework

Each genomic variant (SNVs and indel) is annotated by one of the available gene models (RefSeq or GENCODE). Non coding RNAs and

ORF genes were excluded. Selenoproteins, for which the UGA-stop-codon in the middle of their coding regions codifies for selenocysteine were included: notably, we observed that no one of the other mentioned prediction tools cited in this paper is able to correctly deal with these particular genes, even if several disorders involving changes in selenoprotein structure, activity or expression have been reported[197]. Therefore, only variants that overlap the identified coding regions of the above gene models are considered for downstream analysis. All possible transcripts for which the variant is coding are retrieved, and features are computed for each transcript. In particular, for PhyloP and Gerp++ positional scores that can involve more than one base change, (a) in case of deletion/indel the maximum score between the deleted bases is taken, while (b) in case of insertion the maximum score between the two neighbour genomic positions is taken. Siphy score is included only in case of missense SNVs using dbNSFP (v2.1) data source[198]. DbNSFP database was used to retrieve SIFT and PolyPhen2 pre-computed prediction scores as well.

The PaPI annotation framework has been written in Java and customized Tabix [157]libraries have been used to perform a fast genomic interval search on compressed data files.

4.3.6. Parameter Tuning

The implemented RF model is based on Weka libraries[9].We tuned RF model parameters by running an independent 10-fold cross validation on each of the generated training sets. The considered parameters were four, two related to the RF (number of trees and number of features per node) and two related to the PseAAC (λ and w). Parameter details are shown inTable 10. Note that the 41 amino acid snippet length used to compute PseAAC is fixed and it was not included in the optimization parameters. For each training set, we obtained the same optimal set of parameters, w = 0.1, λ = 12, number of trees = 250 and number of features per node = 2.According to the PseAAC representation, λ determines the number of positional features (if λ=0, we have the traditional amino acid composition representation). In the amphiphilic PseAAC, features are 20 (frequency related) + 2*λ (positional). As a consequence, the total number of features varies according to λ, from a minimum of 30 (24 for PseAAC, 3 for quantitative attributes and 3 evolutionary conservation scores) to a maximum of 66 (60 for PseAAC + 3 for quantitative attributes + 3 evolutionary conservation scores). Thus, with λ = 12, our RF model uses 50 features (44 + 3 for quantitative attributes + 3 evolutionary conservation scores). Being λ responsible for 4 to 60 features in the RF model, the feature selection process stands implicitly in the λ parameter tuning. The selected model includes 50 features and it is trained on datasets including tens of thousands variants (from 31601 to 44888, as shown in Table 7).

Since the number of samples is much greater than the number of features, we did not proceed with a further feature selection.

Amino acid sequences shorter than λ+1 cannot be represented with PseAAC. This issue, nevertheless, can happen only in the case of a coding mutation that introduces a premature stop-codon at the beginning of the protein: this is the case of stop-gain variants; these mutations are automatically labeled as deleterious. It has to be noted that, considering λ = 12, only 438 mutated sequences (out of about 204K of the overall dataset) were too short to be represented by this PseAAC model.

Parameter	**Used in**	**Values**
Num trees	RF	5, 10, 50, 100, 150, 200, 250, 300, 350
Num features per node	RF	int(log(# trees) +1), 2, 4
λ	PseAAC	4, 8, 12, 16, 20
w	PseAAC	0.5, 0.1

Table 10. Parameter values used for RF model tuning. List of parameters and relative values used for the optimization of the RF model on training sets.

4.4. Results and Discussion

Known coding disease-related variants (damaging) were retrieved from HGMD, including SNVs and indels. We assumed that frequent genomic variants are less suitable of being deleterious, therefore, tolerated variants were retrieved by combining 1TGP and ESP selecting only polymorphic (frequency higher than 0.05) and unique variants. Due to the unbalancing between damaging versus tolerated variants of the resulting dataset (Table 6), we randomly split it into three quasi-balanced sets. We further split each into a training (70%) and test (30%) set (Table 7). The whole process has been explained in4.3.3.

For each variant, the difference in PseAAC between wild (reference genome) and mutated amino acid sequence was computed, resulting in a set of quantitative features, used to train and test a machine learning classifier.

Three evolutionary conservation scores and three full length protein attributes were included in the feature set as well (see 4.3.2).

An RF and a Logistic Regression (LR) models were built upon the resulting training sets, while performances were measured on each relative test set. The RF achieved an average area under the curve (AUC) of 0.897 and an average accuracy of 0.832 on the three sets, resulting in performances higher than the LR ones (AUC=0.878, accuracy= 0.813, see Table 11 and Figure 43). The gap between the two classifiers can be explained by the complexity of the feature set: given its non-linear nature, RF is more suitable to detect hidden structures in data with respect to the LR.

Test Set	**M**	**AUC**	**Accuracy (IC95%)**	**Sens**	**Spec**	**PPV**	**NPV**	**F-m**	**MCC**
# 1	RF	.898	.8314 (.8381-.8246)	.835	.827	.829	.833	.832	.662
	LR	.877	.8118 (.8188-.8047)	.841	.782	.795	.830	.817	.624
# 2	RF	.90	.8310 (.8377-.8242)	.837	.825	.828	.834	.832	.662
	LR	.875	.8121 (.8190-8049)	.846	.777	.793	.834	.818	.625
# 3	RF	.903	.8344 (.8422-8262)	.840	.828	.831	.837	.835	.668
	LR	.883	.8168 (.8250-.8083)	.845	.787	.800	.835	.822	.634

Table 11. Performances of RF and LR Models (M) on the three test sets.Performances of the Random Forest (RF) and Logistic Regression (LR) on the three test sets. Area under the curve (AUC), accuracy with 95% confidence interval, sensitivity (Sens), specificity (Spec), Positive Predictive Value (PPV), Negative Predictive Value (NPV), F-measure (F-m) and Matthews correlation coefficient (MCC) are reported for each method.

In order to quantify the contribution of PseAAC features in classification we measured the performance of the RF trained on the aforementioned training sets without evolutionary conservation scores and full length protein features (see Table 12). In other words, we assessed a RF model based on PseAAC only. Notably, the RF trained solely on PseAAC reached, on average, an AUC and accuracy of 0.88 and

0.82,respectively. This is only about one percent less than the RF holding the complete feature set.

Tool	Test Set	AUC	Balanced accuracy	F-m	MCC
RF – PseAAC only	# 1	.885	.816	.816	.628
	# 2	.888	.821	.821	.637
	# 3	.892	.827	.827	.65

Table 12. Performances of the RF trained only using PseAAC features, measured on the three unfiltered test sets. Area under the curve (AUC), balanced accuracy (sensitivity/2 + specificity/2), F-measure (F-m) and Matthews correlation coefficient (MCC) are shown.

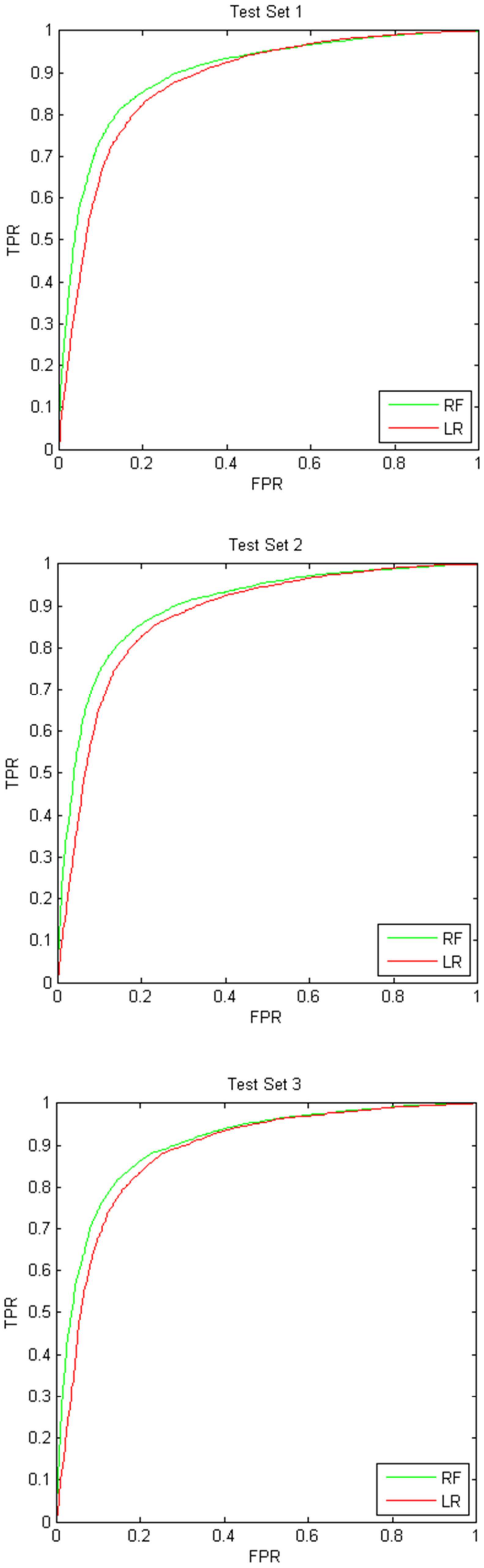


Figure 43. ROC curves of Logistic Regression (LR) and Random Forest (RF) on the three unfiltered variant test sets.

We finally combined the RF model with PolyPhen2 and SIFT scores through the implemented voting scheme. In order to independently measure performances of the three algorithms on the same data, test sets were filtered out for variants that PolyPhen2 and/or SIFT were not able to predict. The combined approach, which we called PaPI, increased the overall performances: AUC, accuracy and Matthews correlation coefficient (MCC) are increased in average by 2, 3 and 7 percentage points respectively when compared to the RF model alone. Sensitivity, specificity and other performance metrics of the RF, PolyPhen2, SIFT and PaPI on the three test set are reported in Table 13,while receiver operating characteristic (ROC) curves are reported in Figure 45.

Test Set	**Tool**	**AUC**	**Acc (IC95%)**	**Sens**	**Spec**	**PPV**	**NPV**	**F-m**	**Mcc**
# 1	PaPI	.9207	.8621 (.8553-.8685)	.858	.866	.868	.855	.863	.724
	RF	.8941	.8262 (.8189-.8334)	.828	.823	.829	.823	.828	.652
	PP2	.9137	.8425 (.8354-.8493)	.853	.831	.839	.846	.846	.684
	SIFT	.8682	.8045 (.7968-.812)	.772	.837	.830	.781	.800	.610
# 2	PaPI	.9196	.8618 (.8550-.8683)	.857	.866	.869	.854	.863	.723
	RF	.8960	.8275 (.8202-.8346)	.831	.823	.829	.825	.830	.654
	PP2	.9121	.8401 (.8330-.847)	.848	.831	.838	.841	.843	.680
	SIFT	.8677	.7994 (.7917-.807)	.762	.837	.829	.773	.794	.601
# 3	PaPI	.9239	.8648 (.8568-.8724)	.857	.872	.874	.855	.865	.729
	RF	.8999	.8289 (.8202-.8373)	.835	.821	.828	.828	.832	.657
	PP2	.9185	.8416 (.8331-.8497)	.850	.832	.840	.843	.845	.683
	SIFT	.8688	.7999	.755	.845	.834	.770	.793	.603

			(.7906- .8088)						

Table 13. Performances of RF, PolyPhen2, SIFT and PaPI on the three test sets. Performances of the Random Forest (RF), PolyPhen2 (PP2), SIFT and PaPI (RF + PolyPhen2 + SIFT) on the three test. Area under the curve (AUC), accuracy (Acc) with 95% confidence interval, sensitivity (Sens), specificity (Spec), Positive Predictive Value (PPV), Negative Predictive Value (NPV), F-measure (F-m) and Matthews correlation coefficient (Mcc) are reported for each method. Test sets were filtered in order to retain only those variants that both PolyPhen2 and SIFT were able to predict.

Being PaPI an ensemble method based on three classifiers, we also analyzed the prediction consistency among the three tools. The great majority of the correct predictions (over 75%) finds RF, PolyPhen2 and SIFT in agreement. More details are summarized by the Venn diagrams reported in Figure 44.

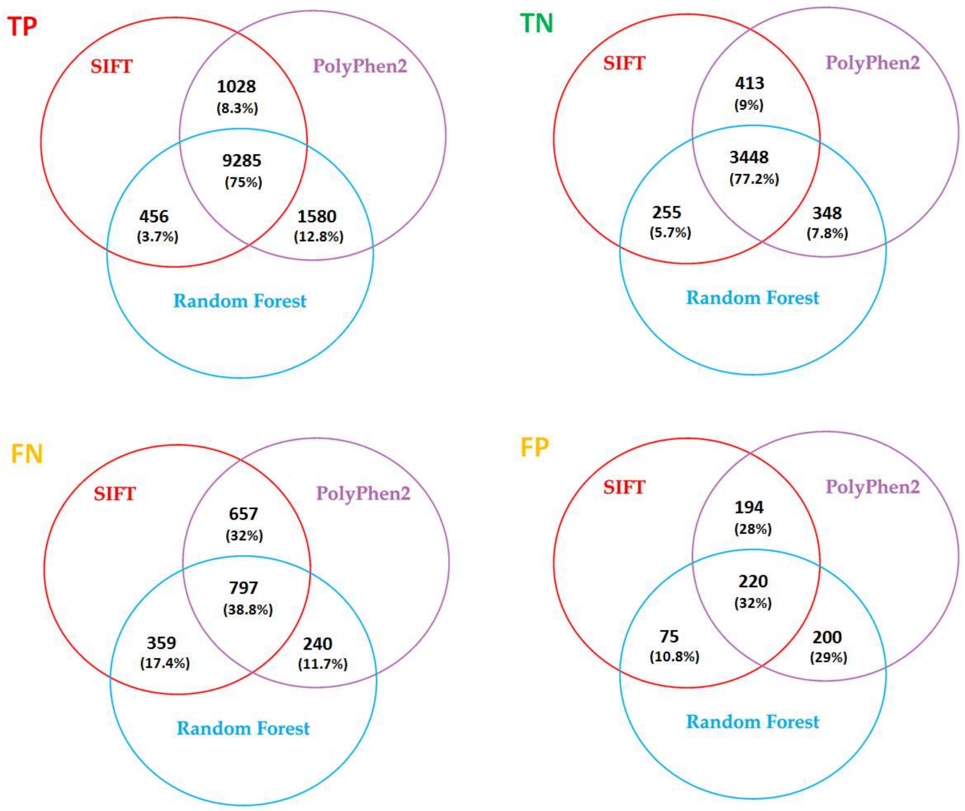


Figure 44. Venn diagrams showing contingencies in terms of prediction agreement between the Random Forest, SIFT and PolyPhen2 on the whole variant test set where both PolyPhen2 and SIFT hold a prediction. P=Positive (Damaging), N=Negative (Tolerated).

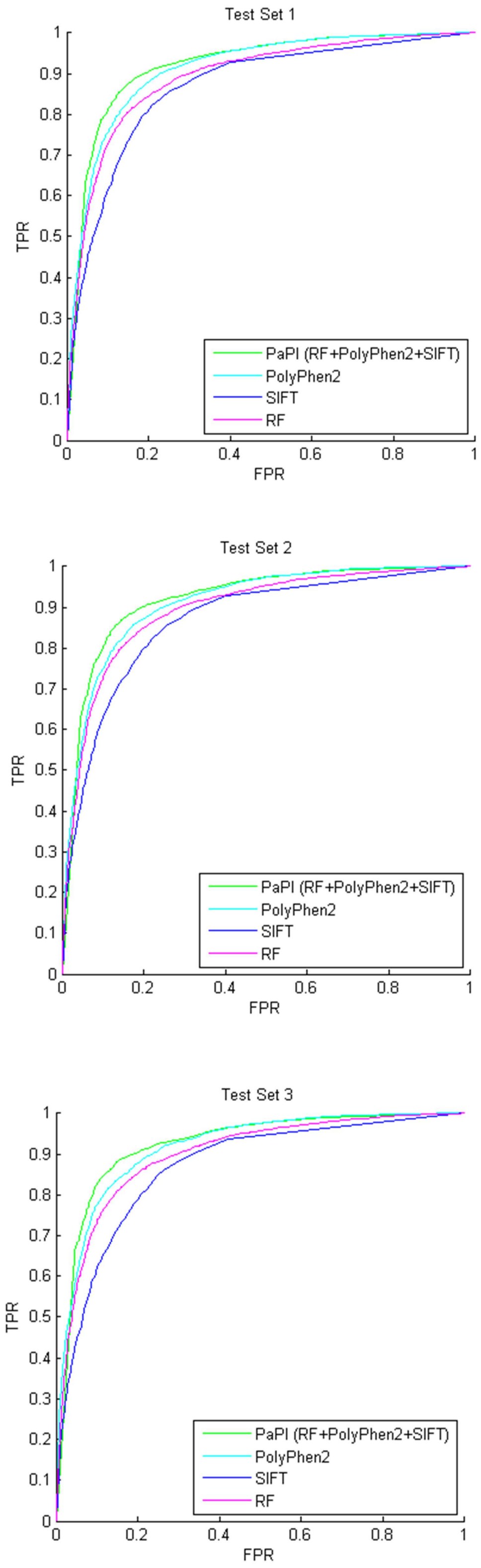


Figure 45. ROC curves of the RF, PolyPhen2, SIFT and PaPI (RF+PolyPhen2+SIFT). ROC curves of Random Forest (RF), PolyPhen2, SIFT and their ensemble (PaPI) on the three test sets. Variants that PolyPhen2 and/or SIFT were not able to predict were filtered out.

4.4.1. Performances on Unpredictable Variants for PolyPhen2 and SIFT

We further proceeded to evaluate PaPI performances on those variants of the test sets for which both PolyPhen2 and SIFT were unable to give a prediction, resulting in a total of 416 tolerated and 974 damaging missed variants. In these cases, PaPI classes and scores coincide with the RF predictor ones. The average area under the curve (AUC) of the RF was equal to 0.94 while the average accuracy on the three variant sets was equal to 0.87 (see Table 14for the performance metrics and Figure 46 for ROC curves).

Test Set	**Tool**	**AUC**	**Acc(IC95%)**	**Sens**	**Spec**	**PPV**	**NPV**	**F-m**	**Mcc**
# 1	PaPI (RF)	.9368	.8676 (.8420-.8896)	.9171	.8245	.8198	.9196	.8657	.7405
#2	PaPI (RF)	.9418	.8611 (.8352-8836)	.9214	.8077	.8095	.9205	.8619	.7296
# 3	PaPI (RF)	.942	.8830 (.8523-9080)	.9256	.845	.8421	.9271	.8819	.7699

Table 14. PaPI performances on the unpredictable variants by PolyPhen2 and SIFT. PaPI performances on the three test retaining only those variants unpredictable both for PolyPhen2 and SIFT. In this case, PaPI coincides with RF. Area under the curve (AUC), accuracy (Acc) with 95% confidence intervals, sensitivity (Sens), specificity (Spec), Positive Predictive Value (PPV), Negative Predictive Value (NPV), F-measure (F-m) and Matthews correlation coefficient (Mcc) are reported for each method.

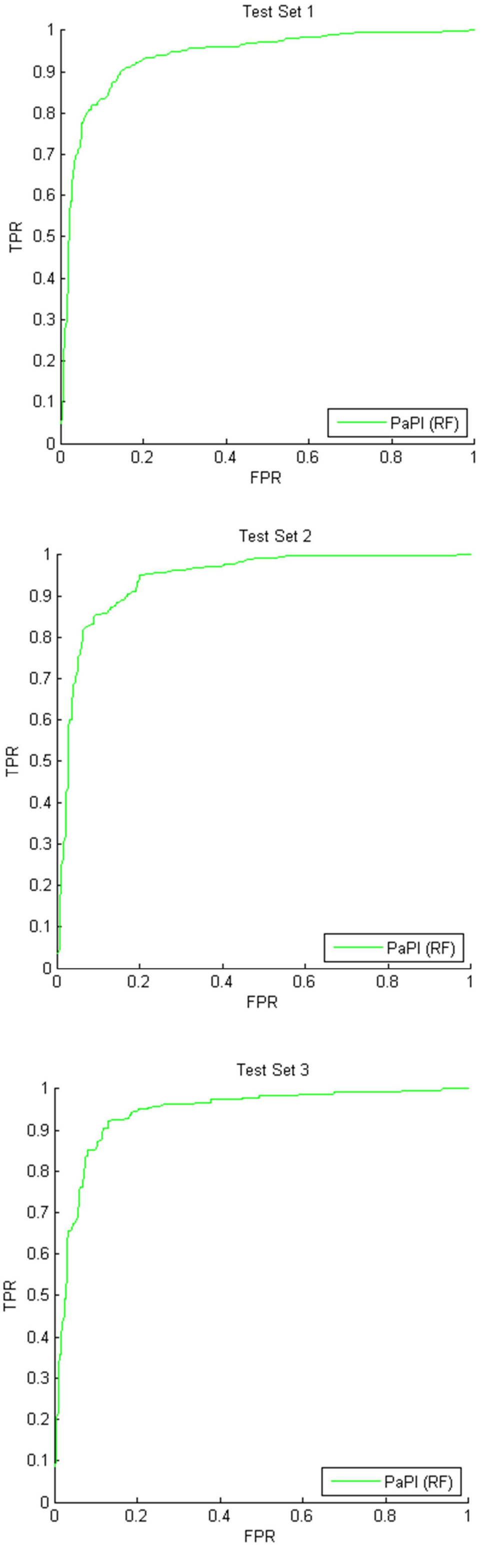


Figure 46.ROC curves of PaPI on the three variant test sets after retaining those variants unpredictable for both PolyPhen2 and SIFT. For these cases PaPI coincides with RF.

4.4.2. Comparison with Other Variant Prediction Tools

PaPI's performances were compared to the following variant predictors: Carol[181], PROVEAN[127], FATHMM[175], MutationAssessor[125], LRT [124], PolyPhen2 and SIFT. Thanks to the RF model, PaPI is capable to score variants of any kind up to 60 nucleotides (see 4.3.2.1). PolyPhen2, SIFT, FATHMM, MutationAssessor and LRT only classify SNVs, while PROVEAN deals with in-frame but not frameshift indels. Furthermore, these tools may be unable to provide some predictions due to lack of information (e.g. when only few homologous sequences exist or remain after their filtering). Therefore, in order to obtain a fair comparison, we removed from the aforementioned test sets those variants that other algorithms were unable to score (see Table 9).While PaPI scored every variant, missing rates of the other prediction tools on the three sets ranged from 7.34% to 23%, as reported inTable 15.

	% Missing rate		
	Set #1	**Set #2**	**Set #3**
PolyPhen2	7.66	7.68	7.34
SIFT	10.41	10.39	9.8
Carol	11.05	11.44	10.98
PROVEAN	9.85	9.89	9.34
FATHMM	7.64	7.72	7.09
MutationAssessor	9.78	9.72	9.58
LRT	22.99	22.82	22.23
PaPI	**0**	**0**	**0**

Table 15. Missing rates on the three unfiltered test sets. Missing rates (i.e. algorithm unable to provide prediction) of considered algorithms on the three unfiltered test sets.

The average AUC and balanced accuracy of PaPI were of 0.926 and 0.864, respectively, reporting an average increase of 1.5 and 3.3 percentage points in balanced accuracy and MCC when compared to the second best predictor (Carol). Negative/positive predictive values and other performance metrics are reported in Table 16. ROC curves of each predictor are reported in Figure 47.

Set	Tool	AUC	Balanced Accuracy	Sens	Spec	PPV	NPV	F-m	MCC
# 1	**PaPI**	**.922**	**.8575**	**.852**	**.863**	**.899**	**.803**	**.875**	**.708**
	Carol	.912	.8492	.821	.877	.905	.774	.861	.689
	PRO	.893	.8264	.789	.863	.892	.741	.837	.643
	SIFT	.883	.8142	.763	.865	.889	.718	.821	.618
	PP2	.914	.8425	.850	.834	.880	.796	.865	.680
	FHMM	.830	.7517	.626	.876	.878	.621	.731	.502
	LRT	.845	.8249	.800	.848	.883	.749	.840	.640
	MutAss	.889	.812	.757	.866	.89	.714	.818	.614
# 2	**PaPI**	**.925**	**.863**	**.862**	**.864**	**.899**	**.817**	**.880**	**.721**
	Carol	.912	.8442	.811	.877	.902	.767	.854	.679
	Provean	.898	.8354	.807	.863	.892	.761	.847	.662
	SIFT	.883	.8091	.753	.865	.887	.713	.814	.609
	PolyPhen2	.918	.8491	.863	.834	.880	.813	.871	.695
	FATHMM	.835	.7603	.644	.876	.880	.636	.743	.518
	LRT	.850	.8317	.814	.848	.883	.765	.847	.656
	MutAssessor	.892	.8134	.760	.866	.888	.720	.819	.617
# 3	**PaPI**	**.933**	**.8721**	**.875**	**.869**	**.905**	**.831**	**.89**	**.74**
	Carol	.923	.8551	.818	.891	.914	.776	.863	.700
	Provean	.915	.8444	.815	.873	.901	.769	.856	.679
	SIFT	.891	.8166	.759	.874	.895	.719	.821	.623
	PolyPhen2	.930	.8542	.872	.835	.882	.822	.877	.706
	FATHMM	.843	.7643	.641	.876	.889	.635	.745	.52
	LRT	.868	.8408	.828	.852	.888	.778	.857	.674
	MutAssessor	.898	.8273	.777	.877	.899	.735	.834	.644

Table 16. Performances of different prediction tools on the three filtered test sets. Comparison of PaPI, PolyPhen2, SIFT, Carol, PROVEAN, FATHMM, LRT and MutationAssessor on the three test sets filtered for unpredictable variants by the other prediction tools. Area under the curve (AUC), balanced accuracy (sensitivity/2 + specificity/2), sensitivity (Sens), specificity (Spec), Positive Predictive Value (PPV), Negative Predictive Value (NPV), F-measure (F-m) and Matthews correlation coefficient (MCC) are reported for each method

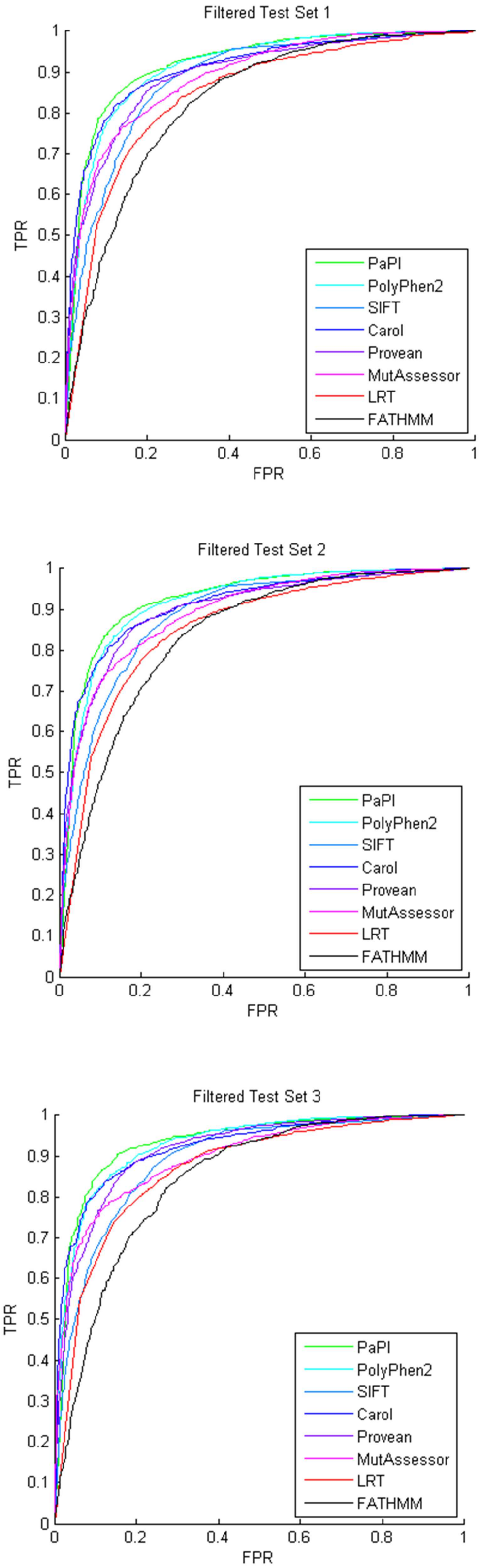


Figure 47.ROC curves comparison between prediction tools. ROC curves of PaPI, PolyPhen2, SIFT, Carol, PROVEAN, FATHMM, LRT and MutationAssessor on the three filtered test sets.

4.4.3. PaPI Exploits Pseudo Amino Acid Composition Substitution Patterns for Disease-related Variant Prediction

We hereby show the case of two disease-related variants for which the RF is the solely to correctly assign the right prediction in contrast to PolyPhen2, SIFT and the other cited tools. These two examples show therefore how the RF can positively contribute to a correct variant evaluation by exploiting pseudo amino acid composition substitution patterns in specific protein-coding regions.

4.4.3.1. GDF6 - Ala249Glu

Several studies associated the mutation p.(Ala249Glu) in the growth differentiation factor 6 (GDF6) with skeletal and ocular abnormalities [199-201].

The p.(Ala249Glu) mutation is within the C-terminal-half of the GDF6 prodomain, a region thought to facilitate correct disulfide folding of the mature secreted peptide and latent complex formation[202].Despite segregation and functional analysis conducted by the studies mentioned above demonstrated the mutation hypomorphicity, its function within the domain remains unclear; moreover, the residue is not highly conserved and only two homologue proteins hold the same residue. Interestingly, the mutation is in a region rich in GC-content that overlaps with a CpG island of 1267 nucleotide bases. The p.(Ala249Glu) mutation, given by c.746C>A nucleotide variant, is part of ten adjacent bases that exhibit a variable percentage in cytosine methylation according to the Reduced Representation Bisulfite Sequencing ENCODE data from five different human cell types (see Figure 48). CpG islands are known to be regions with high germline mutational rate in methylated CpGs[203]and in coding regions they cause biases into the amino acid sequences[186, 187]. For example, Arg residue mutates more frequently respect to any other residue and this can be in part explained by the fact that CpG in the Arg codons occur in the no-wobble positions resulting into missense variants.

	Damaging	Tolerated
CpG	15031	3098
¬ CpG	137793	22506

Table 17.Contingency table to compare the proportion of variants overlapping CpG islands into the whole Damaging and Tolerated data set.Fisher exact test has been performed by Matlab function developed by Michael Boedigheimer "fexact" (http://www.mathworks.com/matlabcentral/fileexchange/22550-fisher-s-exact-test).

By analysing our training data set, we found 15,031 and 3,098 variants overlapping CpG island for the damaging and tolerated variant set, respectively, resulting in a significant difference in numbers between the two sets (p = 1.7 x 10-27; Fisher exact test, see Table 17).

The proportion of the amino acid changes inside and outside methylated regions of CpG islands (human cell line GM12878, downloaded by UCSC Genome Browser) resulted significantly higher for the damaging variant set with respect to the tolerated one (p = 3.2 x 10^{-2}; Fisher exact test, see Table S4 in Additonal File 1).

	Damaging	**Tolerated**
mCpG	806	137
¬mCpG	14225	2961

Table 18. Contingency table to compare the proportion of variants inside and outside methylated regions of CpG islands into the whole Damaging and Tolerated data set.Fisher exact test has been performed by Matlab function developed by Michael Boedigheimer "fexact" (http://www.mathworks.com/matlabcentral/fileexchange/22550-fisher-s-exact-test)

These results show biases between damaging and tolerated data sets, thus amino acid composition following amino acid changes within these regions may be different, too. We suppose that the RF may have learnt the model of substitution patterns in CpG islands within the tolerated and damaging datasets, thus correctly predicting the p.(Ala249Glu) in the GDF6 encoded protein.

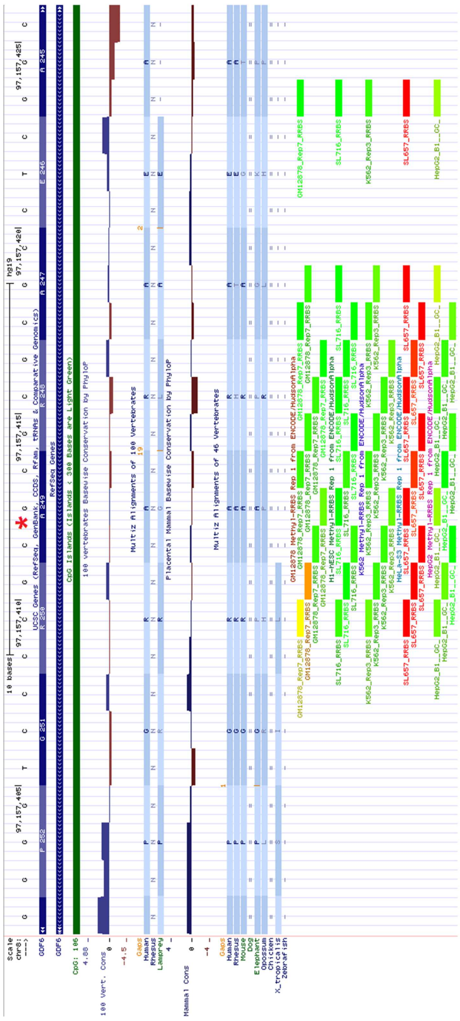

Scale
chr8:
10 bases
hg19
97,157,405
97,157,410
97,157,415
97,157,420
97,157,425
GDF6
UCSC Genes (RefSeq, GenBank, CCDS, Rfam, tRNAs & Comparative Genomics)
RefSeq Genes
CpG Islands (Islands < 300 Bases are Light Green)
100 vertebrates Basewise Conservation by PhyloP
Multiz Alignments of 100 Vertebrates
Placental Mammal Basewise Conservation by PhyloP
Multiz Alignments of 46 Vertebrates
Human
Rhesus
Mouse
Dog
Elephant
Opossum
Chicken
X_tropicalis
Zebrafish
GM12878_Rep7_RRBS
SL716_RRBS
K562_Rep3_RRBS
SL657_RRBS
HepG2_B1__GC_

Figure 48. Mutation p.(Ala249Glu) (red asterisk) in the GDF6 gene from UCSC Genome Browser. CpG island (dark green), Mammal and Vetrebrate conservations (light blue), Methylation sites from 5 cell types (from light green to red respect to the percentage of DNA molecules that exhibit cytosine methylation) are shown.

4.4.3.2. GUCY2C - Asp387Gly

Romi et al.[194] identified a single mutation p.(Asp387Gly) in the guanylate cyclase 2C (GUCY2C) transmembrane receptor causing meconium ileus (MI), an intestinal obstruction in newborns. GUCY2C has an extracellular domain that is activated by ligands (guanylin and related peptide uroguanylin or E.coli heat-stable enterotoxin STa). The p.(Asp387Gly) mutation is within an essential region of its extracellular ligand-binding domain and is adjacent to seven other pivotal amino acids for the ligand binding [204]. The resulting significant reduction of ligand-binding leads to a reduction in guanylate cyclase activity and activates a signalling cascades that finally leads to MI.

The GUCY2C extracellular domain belongs to the periplasmic binding protein-like I superfamily domain and corresponds to the extracellular ligand-binding receptor, IPR001828 in InterPro database[205].The same domain is shared by other 157 human proteins (including the one encoded by GUCY2C). Among the disease variant set used for RF training, 242 disease variants belonging to the IPR001828 domain and related to 7 proteins are present. We suppose that the RF learned the model of substitution patterns in the extracellular domain of these proteins and therefore it was able to assign the correct prediction for p.(Asp387Gly) in the GUCY2C encoded protein.

4.4.4. PaPI Leads to Right Prediction in Case of PolyPhen2 and SIFT Conflict

Here we report an example that shows how PaPI can correctly classify variants for which PolyPhen2 and SIFT are discordant in prediction.

Tchernitchko *et al.*[206]compared PolyPhen and SIFT considering several variants known to be responsible for affecting the products of hemoglobin and glucose-6-phosphate dehydrogenase genes, leading to several forms of sickle cell anemia and G6PD deficiency, respectively. In the results in that paper, PolyPhen and SIFT had discordant predictions on ten pathogenic variants, for which experimental evidence was reported. We therefore run PaPI on the same variant set and we correctly classified all of them as damaging. Five variants were predicted both by PolyPhen2 and SIFT (thanks to updated versions now available) as damaging while for the other five variants there were still discordant predictions between these tools (Table 19).

Gene	Protein	PP2	SIFT	RF	Related Phenotype	PaPI
HBB	p.E7V	B (0.002)	D (0.01)	D (0.891)	Sickle cell anemia	**D (0.901)**
	p.E122Q	B (0.007)	D (0.01)	D (0.975)	Severe sickle cell syndromes	**D (0.975)**
	p.E122K	B (0.109)	D (0.0)	D (0.96)		**D (1.0)**
	p.E7K	B	D	D		**D**

		(0.006)	(0.01)	(0.94)		**(0.94)**
G6PD	p.S188F	B (0.039)	D (0.04)	D (0.988)	G6PD deficiency	**D (0.988)**

Table 19. Examples of known disease-related variants. Known disease-related variants reported by Tchernitchko et al. for which occurs a different outcome in prediction by PolyPhen2 (PP2) and SIFT. For these cases, the RF is able to vote for the right class leading PaPI to the correct prediction as well. In brackets the score for each variant predictor is reported. B= tolerated, D=damaging.

For these cases, the RF vote allowed obtaining the right class assignment. Notably, SIFT is able to assign the right class for each variant as well, despite we show that RF and SIFT have the lowest concordance rate in prediction in case of PolyPhen2 conflict (see Figure 44).

4.4.5. Web Accessible Tool

PaPI software is freely available online (http://papi.unipv.it) as a web accessible tool.

The user interface allows to submit a single variant or to perform queries in bulk by uploading a plain text file with a list of variants (see for a screenshot).

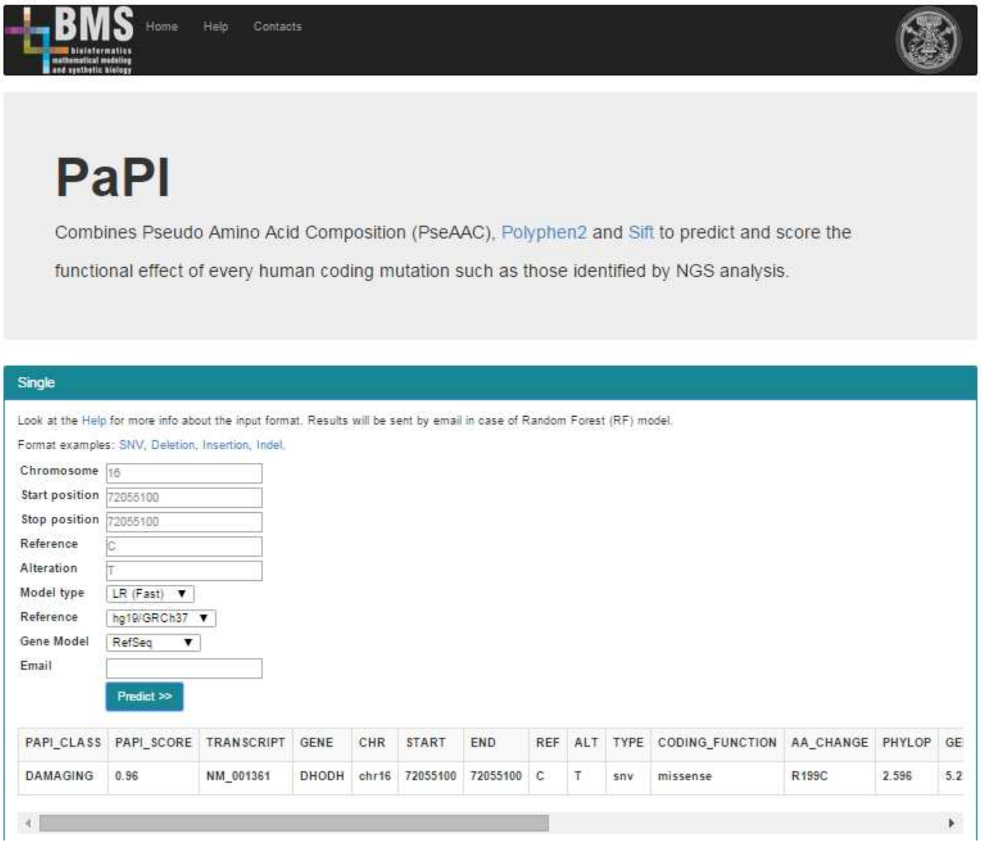
Users can choose between the RF and LR model. Although we showed LR is less accurate than RF, it is faster and can be used for a quick response.

Two different gene annotation models are available (RefSeq and GENCODE) and a variant score is given for each different transcript.

Results are reported in a tab-delimited text file and can be sent by email: PaPI prediction (damaging or tolerated) along with its confidence score plus prediction/scores of RF/LR, PolyPhen2 and SIFT. Each variant comes with information about transcript, gene, type (missense, synonymous, frameshift etc.) and evolutionary conservation scores. Prediction runtime takes, in average, between 0.3 and 0.7 seconds per variant.

The business logic of the web application has been developed in Java, allowing asynchronous processes managed by two main queues: one for the RF requests and another for the LR ones. The RF queue run requests one by one (due to the RAM usage to put in memory the Random Forest model) while the LR can execute 4 requests in parallel.

Presentation tier has been designed using a Model-View-Controller pattern (MVC) by employing Java Server Faces (JSF) technology.



PaPI

Combines Pseudo Amino Acid Composition (PseAAC), Polyphen2 and Sift to predict and score the functional effect of every human coding mutation such as those identified by NGS analysis.

Single

Look at the Help for more info about the input format. Results will be sent by email in case of Random Forest (RF) model.

Format examples: SNV, Deletion, Insertion, Indel.

Chromosome	16
Start position	72055100
Stop position	72055100
Reference	C
Alteration	T
Model type	LR (Fast) ▼
Reference	hg19/GRCh37 ▼
Gene Model	RefSeq ▼
Email	

PAPI_CLASS	PAPI_SCORE	TRANSCRIPT	GENE	CHR	START	END	REF	ALT	TYPE	CODING_FUNCTION	AA_CHANGE	PHYLOP	GE
DAMAGING	0.96	NM_001361	DHODH	chr16	72055100	72055100	C	T	snv	missense	R199C	2.596	5.2

Figure 49. PaPI web interface, results for single query are shown in the browser, or can be sent by email.

4.4.6. Conclusions

We developed a new method, called PaPI, to classify and score human coding variants potentially leading to functional alterations of related proteins, especially as human inherited diseases are concerned, since the algorithm has been trained on HGMD database, which is known to be biased towards human Mendelian diseases. The main novelty of the approach is the introduction of features based on the difference in pseudo amino acid composition between snippets of wild and altered protein sequences where coding variants occur. Hydrophobicity and hydrophilicity pairwise relationships between amino acids are encoded by these features. Evolutionary conservation scores and quantitative descriptors at the whole protein level were included in the feature set as well. A RF classifier was trained on these features to mine disease and neutral pseudo amino acid composition substitution patterns and classify unseen coding variants into damaging or tolerated class.

Despite it has been shown that the combination of variant classifiers is not always beneficial[207],we showed that the implemented voting strategy between PolyPhen2, SIFT and our RF model improves performances in terms of area under the curve, accuracy and other reported metrics in comparison to the ones of each predictor alone. Considering only those variants that PolyPhen2 and SIFT are unable to predict, PaPI maintains high performances thanks to the RF model. Moreover, it has to be noted that in case of prediction by both PolyPhen2 and SIFT, PaPI is biased toward sequence conservation, because of the majority voting system between the RF and these two tools[196].We compared PaPI with other variant prediction tools (PolyPhen2, SIFT, Carol, PROVEAN, FATHMM, MutationAssessor, LRT) and we showed that PaPI performances were the highest on the data sets used. Notably, PaPI is able to score any variant, including the ones that the other mentioned methods were unable to predict.

We have reported two examples for which the RF model is the only algorithm that predicts the correct class, thanks to its capability of exploiting potential disease-related pseudo amino acid composition substitution patterns such as protein ligand-binding domains and CpG regions. We also showed several examples where the RF model vote leads to a correct prediction, in case of conflict between PolyPhen2 and SIFT.

To our knowledge, PseAAC has never been used in variant prediction. We are confident that the algorithm can be further improved by optimizing other parameters (e.g. length of sequence snippets surrounding variants) or by exploring other PseAAC descriptors (e.g. including amino acid side chain mass property).

Chapter 5

Clinical Applications

In this Chapter, some clinical applications of the Variant Management System based on the Relational Database (RDBVMS) approach discussed in 3.1 are reported.

The presented cases have been published on international scientific journals, as referenced below.

Multiple clinical forms of dehydrated hereditary stomatocytosis arise from mutations in PIEZO1.*Andolfo I, Alper SL, De Franceschi L, Auriemma C, Russo R, De Falco L, Vallefuoco F, Esposito MR, Vandorpe DH, Shmukler BE, Narayan R, Montanaro D, D'Armiento M, Vetro A, Limongelli I, Zuffardi O, Glader BE, Schrier SL, Brugnara C, Stewart GW, Delaunay J, Iolascon A.Blood. 2013 May 9;121(19):3925-35, S1-12. doi: 10.1182/blood-2013-02-482489. Epub 2013 Mar 11.*

Improving molecular diagnosis in epilepsy by a dedicated high-throughput sequencing platform.*Della Mina E, Ciccone R, Brustia F, Bayindir B, Limongelli I, Vetro A, Iascone M, Pezzoli L, Bellazzi R, Perotti G, De Giorgis V, Lunghi S, Coppola G, Orcesi S, Merli P, Savasta S, Veggiotti P, Zuffardi O. Eur J Hum Genet. 2014 May 21. doi: 10.1038/ejhg.2014.92. (Epub ahead of print)*

Lower motor neuron disease with respiratory failure caused by a novel MAPT mutation.*Di Fonzo A, Ronchi D, Gallia F, Cribiù FM, Trezzi I, Vetro A, Della Mina E, Limongelli I, Bellazzi R, Ricca I, Micieli G, Fassone E, Rizzuti M, Bordoni A, Fortunato F, Salani S, Mora G, Corti S, Ceroni M, Bosari S, Zuffardi O, Bresolin N, Nobile-Orazio E, Comi GP.*

Neurology. 2014 Jun 3;82(22):1990-8. doi: 10.1212/WNL.0000000000000476. Epub 2014 May 7.

The contents (including tables and figures) of the following sections have been extracted from the aforementioned papers. Contribution to these studies comprise sequencing data analysis, from raw sequencing data to the list of the genomic variant candidates for each patient, by using the RDBVMS presented in 3.1. Furthermore, statistical analysis has been performed for *Della Mina et al* related work.

5.1. Multiple clinical forms of dehydrated hereditary stomatocytosis arise from mutations in PIEZO1

Autosomal dominant dehydrated hereditary stomatocytosis (DHSt) usually presents as a compensated hemolytic anemia with macrocytosis and abnormally shaped red blood cells (RBCs). DHSt is part of a pleiotropic syndrome that may also exhibit pseudohyperkalemia and perinatal edema. We identified PIEZO1 as the disease gene for pleiotropic DHSt in a large kindred by exome sequencing analysis within the previously mapped 16q23-q24 interval. In 26 affected individuals among 7 multigenerational DHSt families with the pleiotropic syndrome, 11 heterozygous PIEZO1 missense mutations cosegregated with disease. PIEZO1 is expressed in the plasma membranes of RBCs and its messenger RNA, and protein levels increase during in vitro erythroid differentiation of CD341 cells. PIEZO1 is also expressed in liver and bone marrow during human and mouse development. We suggest for the first time a correlation between a PIEZO1 mutation and perinatal edema. DHSt patient red cells with the R2456H mutation exhibit increased ion-channel activity. Functional studies of PIEZO1 mutant R2488Q expressed in Xenopus oocytes demonstrated changes in ion-channel activity consistent with the altered cation content of DHSt patient red cells. Our findings provide direct evidence that R2456H and R2488Q mutations in PIEZO1 alter mechanosensitive channel regulation, leading to increased cation transport in erythroid cells.

5.1.1. Introduction

Dehydrated hereditary stomatocytosis (DHSt), also known as hereditary xerocytosis (OMIM=194380), is an autosomal dominant congenital hemolytic anemia associated with a monovalent cation leak. DHSt consists of a usually compensated hemolysis, associated with moderate splenomegaly[208]. Blood smears show variable numbers of stomatocytes, sometimes rare and ill-formed, and likely to be overlooked. The reticulocyte count is elevated, and red cell mean corpuscular volume (MCV) is slightly increased. DHSt red blood cells (RBCs) exhibit decreased intraerythrocytic K+ content and increased intraerythrocytic Na+ content, usually accompanied by increased mean corpuscolar hemoglobin (Hb) concentration.

The cation leak of DHSt red cells resembles that of control RBCs in its temperature dependence, but is of greater magnitude at all temperatures[209]. The definitive diagnosis of DHSt is ascertained by osmotic gradient ektacytometry, which shows a leftward shift of the bell-shaped curve[210]. Occasionally, associated hepatosiderosis beyond that expected from the mild hemolytic state suggests a strong tendency to iron overload[211]. Unlike hereditary spherocytosis, in which splenectomy can be beneficial, splenectomy in DHSt is contraindicated due to increased risk of thromboembolic complications[212]. DHSt can present as an isolated erythroid phenotype or as associated with pseudohyperkalemia, with pre- and/or perinatal edema, or with both pseudohyperkalemia and effusions.

The pre- and/or perinatal edema is of chylous type and may lead to life-threatening hydrops fetalis requiring therapeutic drainage[213]. Remarkably, the edema recede spontaneously before birth or within several months postnatally, and do not reappear. In contrast, edema may also be restricted to prenatal, clinically silent ascites detectable only by ultrasound.

Isolated familial pseudohyperkalemia (FP) is defined by the time-dependent elevation in serum (K+) when blood samples are left for several hours or more prior to analysis at temperatures below body temperature, whereas serum (K+) is normal in freshly drawn blood. FP may be associated with DHSt or, when linked to chromosome 2, as isolated FP. The causative gene of isolated FP linked to 2q35-36 was recently identified as ABCB6, encoding a porphyrin transporter[214].

Mapping of gene(s) responsible for familial DHSt identified a cosegregating critical region at the telomeric region of 16q[215]. Zarychanski and colleagues reported for the first time, in 2 families with isolated DHSt, 2 missense mutations in the FAM38A gene encoding PIEZO1[216]. We report here our independent findings in 7 unrelated families with isolated DHSt, DHSt with pseudohyperkalemia, or DHSt with both pseudohyperkalemia and pre-/perinatal fluid effusion, novel mutations in the PIEZO1 gene that cosegregate with the multiple disease phenotypes. We have further characterized PIEZO1 expression in erythroid cells and during mouse and human development, and performed functional studies on R2488Q and R2456H mutations in human erythrocytes and Xenopus oocytes.

5.1.2. Material and Methods

5.1.2.1. Case Report

The clinical phenotypes of kindreds Arras (AR), Bicetre (BI), Dax (DA), Essex, and Troyes (TR) were previously described[217]. Definitive diagnosis of DHSt was made by ektacytometry. Families Dax and Troyes showed isolated DHSt.

Families Arras and Edinburgh showed DHSt plus pseudohyperkalemia [218]. Family Bicetre exhibited DHSt accompanied by pseudohyperkalemia and massive perinatal fluid effusions that spontaneously and permanently regressed within several months after birth. Similar massive but transient perinatal fluid effusions have been observed by others [219]. Following our initial studies, the patients were followed up locally.

The expressivity of the phenotype was generally similar among members from a given kindred, but some variability was noted. (1) In family Dax, the MCV and ektacytometric curve of the father (II.1; numbering from Grootenboer et al) were only slightly altered, whereas the son (II.2) exhibited a full-fledged DHSt [217].(2) In family Bicetre, whereas DHSt in father II.3 [220], was accompanied by pseudohyperkalemia and dramatic perinatal fuid effusions, his 2 children exhibited effusions which were less pronounced. Splenectomy was performed in only 2 patients. In keeping with Stewart et al [209], member II.2 of family AR developed a thrombosis after a period in an ankle cast, followed some years later by a moderately severe pulmonary embolus, treated by chronic anticoagulation. The second patient, member II.2 of family DA, was without thromboembolic complication at the time of examination.

Patient SF, a 38-year-old female triathlete, has not been reported previously. She was referred by her primary physician to a hematology clinic for evaluation of hemolytic anemia, first diagnosed at age 14 in the setting of severe weakness of 1-month duration. Similar episodes of weakness recurred once in her 20s and again at the age of 32, unrelated to medications or specific foods, and resolving with supportive care. The patient reported chronic “yellowing of her eyes,” without changes in color of urine or stool, and without fevers or gastrointestinal symptoms. Neither medication nor food triggered these episodes. Family history was notable for recently diagnosed hemolytic anemia in the patient’s brother, accompanied by 50% deficiency of pyruvate kinase, and a report of mild anemia of unclear etiology in the patient’s father.

Physical examination revealed mild scleral icterus and hepatomegaly (edge 1 cm below costal margin). Hematologic indices were: Hb, 13.2 g/L;

hematocrit, 37%; RBCs, 3.573106mm3; MCV, 103.7 fL; red cell distribution width, 13.1%; absolute reticulocyte count, 139,300/mL. The peripheral blood smear revealed spherocytes, macrocytes, and rare stomatocytes and tear drops. Total bilirubin was 1.4 mg/dL, with normal lactate dehydrogenase, and testing for Gilbert syndrome was negative. Direct Coombs test was negative, and red cell glutathione and enzyme activities (G6PD, PK, GPI, HK, ADA) were normal.

Osmotic fragility testing and ektacytometry revealed osmotic resistance, and ektacytometry also showed decreased RBC deformability in hypertonic solutions: hypoosmotic point was 112.8 (normal, 139.8 + 16.0 mOsm/kg); maximum deformability index was 0.63 (normal, 0.54 + 0.06 artificial units), osmotic point was 348.7 (normal, 405.6 + 18.3 mosmol/kg). These results supported the clinical diagnosis of dehydrated stomatocytosis (DHSt) in patient SF.

5.1.2.2. Bioinformatic Analysis for Exome Sequencing

Reads were aligned to the most recent version of human genome (GRCh37/hg19) using the BWA software package (version 0.5.9). Mapped reads were consequently filtered out for polymerase chain reaction (PCR) duplicates by Samtools (version 0.1.18), locally realigned around inferred insertions and deletions, and their base qualities recalibrated in the context of alignment by Genome Analysis Toolkit (version 1.4-21).

Single-nucleotide polymorphisms, short insertions, and deletions were identified by the GATK Unified Genotyper. Resulting variants were filtered out for possible sequencing and alignment artifacts, taking into consideration variant quality, variant read-depth, and the proportion of not-uniquely-mapped reads overlapping variants. Prediction tracks for each mutation were generated by automatic queries to MutationTaster and PolyPhen-2. Output data were filtered on the basis of an autosomal dominant model of inheritance, removing those annotated variants which were out of exome target, synonymous, common (as annotated in dbSNP135), or found in previous exome sequencing of uncorrelated samples. Candidate variants were also compared and prioritized with the 1000 Genome Project Database and the Exome Sequencing Project (ESP) Database (Exome Variant Server, HLBI ESP, Seattle, WA; ESP5400 release). The remaining, filtered variants were assessed for pathogenicity by SIFT. The filtered exome sequencing data were graphically visualized with Integrative Genomics Viewer (IGV) [221], allowing interactive exploration of these genomic datasets.

5.1.2.3. RNA isolation and cDNA synthesis from CD34$^+$cells

Total RNA was isolated from CD341 cells at days 0,7, and 14 of erythroid differentiation. Single-strand complementary DNA (cDNA) was synthesized from 2 µg of RNA template, using 2.5 units of VILO reverse transcriptase (Technologies, Milan, Italy).

5.1.2.4. Two microelectrode voltage clamp of PIEZO1-expressing oocytes

Oocytes were harvested from Xenopus laevis and treated with collagenase as previously described [222]. cRNA was injected in a volume of 50 nL, and oocytes were maintained at 17°C for 72 hours prior to experimentation.

Defolliculated oocytes were placed in ND96 (in mM, 96 NaCl, 2 KCl, 1.8 $CaCl_2$, 1$MgCl_2$, 5 HEPES (N-2-hydroxyethylpiperazine-N’-2-ethanesulfonic acid), pH 7.4) in a bath (RC-16; Warner Instruments, Hamden, CT) on the headstage of an upright microscope and imaged at x20 magnification.

Oocytes were impaled with pipettes fabricated from borosilicate glass (World Precision Instruments, Sarasota, FL) using a Sutter P87 puller. Resistances of electrodes were 2 to 10 megaohms when filled with 3M KCl.

A voltage clamp protocol was generated using the Clampex subroutine of PCLAMP 10 (Molecular Devices Corporation, Sunnyvale, CA), applying 10 sweeps of 400 ms, in 20-mV steps from -70 mV, with a sampling rate of 10 kHz. Holding potential was -30 mV in all groups throughout the experiment.

Currents were recorded using a Geneclamp 500 voltage clamp (Molecular Devices). The bath reference electrode was a silver chlorided wire with a 3M KCl agar bridge. Junction potentials were minimized by using 3M KCl in the pipettes and by use of a bath clamp.

Current voltage relationships were determined by fitting the currents recorded at t = 370 ms using the Clampfit subroutine of PCLAMP 10, and plotted with Sigmaplot graphics.

5.1.2.5. On-cell patch recording of PIEZO1-expressing oocytes

Defolliculated oocytes were placed in a hypertonic bath and the vitelline layer was removed by hand with Dumont no. 5 forceps under x40 magnification. The devitellinized oocyte was immediately placed in a low-volume bath (RC-25; Warner Instruments) on the stage of an Olympus IMT-2 inverted microscope, imaged at x40 and patched with fire-polished pipettes of 7 to 12 megaohms resistance. Bath and pipette solutions contained (in mM) 150 Na methanesulfonate, 10 Na EDTA, and 10 Na HEPES, pH 7.4. On-cell patch recording was performed as previously described in this paragraph, except that currents were elicited by imposition of a 250-ms linear voltage ramp from -100 mV to +100 mV during application of negative pressure (0 or -25 mm Hg) to the pipette port and recorded by pneumatic transducer (Biotek DPM-1B, Winooski, VT).

5.1.3. Results

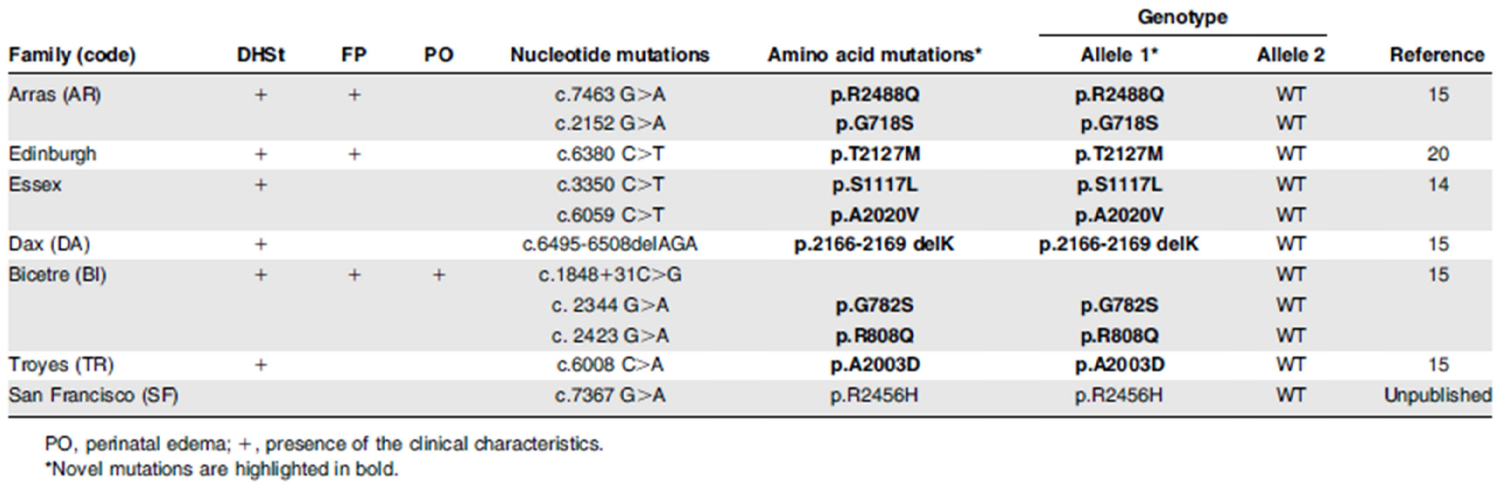
5.1.3.1. Whole Exome analysis

Whole-exome analysis was performed on 2 affected and 2 unaffected members from family Edinburgh (DHSt plus pseudohyperkalemia).After filtering out of likely false-positive single-nucleotide variations(SNVs) and short insertions/deletions (InDels), an average of 32 362variants was called for each of the 4 exomes, spanning about 12 053genes, with about 1700 novel SNVs/InDels per sample. Among novel SNVs/InDels, we focused on heterozygous variants falling in exons, splice-site junctions and 5’ and 3’ untranslated regions that segregated with disease among the 4 individuals and were absent from 38 unrelated exomes from our internal database. This approach highlighted 13 variants in as many genes, 7 of which were exonic, 5 of which were predicted as likely pathological by the in silico tools MutationTaster, PolyPhen-2, and SIFT (see Table 20).

Filtered Variants		
Total variants	30,435	
Variants called under dominant model	1,170	
Variants cosegregating with the disease phenotype	13	
	CDS	Intron / UTR
	7	6
Predicted damaging	5	0

Table 20. Number of called variants through sequential filtering steps

One of these variants mapped within the previously defined critical region on chromosome 16 [218], and was identified asc.6380C>T, T2127M of PIEZO1 (see Figure 50).



Family (code)	DHSt	FP	PO	Nucleotide mutations	Amino acid mutations*	Genotype Allele 1*	 Allele 2	Reference
Arras (AR)	+	+		c.7463 G>A	**p.R2488Q**	**p.R2488Q**	WT	15
				c.2152 G>A	**p.G718S**	**p.G718S**	WT	
Edinburgh	+	+		c.6380 C>T	**p.T2127M**	**p.T2127M**	WT	20
Essex	+			c.3350 C>T	**p.S1117L**	**p.S1117L**	WT	14
				c.6059 C>T	**p.A2020V**	**p.A2020V**	WT	
Dax (DA)	+			c.6495-6508delAGA	**p.2166-2169 delK**	**p.2166-2169 delK**	WT	15
Bicetre (BI)	+	+	+	c.1848+31C>G			WT	15
				c. 2344 G>A	**p.G782S**	**p.G782S**	WT	
				c. 2423 G>A	**p.R808Q**	**p.R808Q**	WT	
Troyes (TR)	+			c.6008 C>A	**p.A2003D**	**p.A2003D**	WT	15
San Francisco (SF)				c.7367 G>A	p.R2456H	p.R2456H	WT	Unpublished

PO, perinatal edema; +, presence of the clinical characteristics.
*Novel mutations are highlighted in bold.

Figure 50.PIEZO1 mutations found in the families here analyzed

5.1.3.2. PIEZO1 Mutational Analysis

PIEZO1 has been sequenced in 26 affected and 16 healthy subjects among 7 affected families. The mutations and the number of affected and unaffected subjects are described in Figure 50.

The nucleotide changes are shown in Figure 51A. None of the noted nucleotide changes were present in the 1000 genomes database, or in 50 healthy subjects here analyzed. The amino acids affected by the PIEZO1 missense mutations identified in the DHSt patients are located in 2 regions of the 2521 amino acid PIEZO1 polypeptide: 1 between residues 718 and 1117, and the carboxy-terminal region beyond residue 2000 (Figure 51B). The DHSt phenotype in families AR, Essex, and BI co-segregated with > 1 novel missense mutation in cis in PIEZO1 (Figure 50). The mutated amino acid residues are all conserved in PIEZO1 of macaque and mouse; 9 of the 10 residues are conserved in rat PIEZO1, and 7 of the 10 are conserved in Xenopus tropicalis and Danio rerio (Figure 51C).

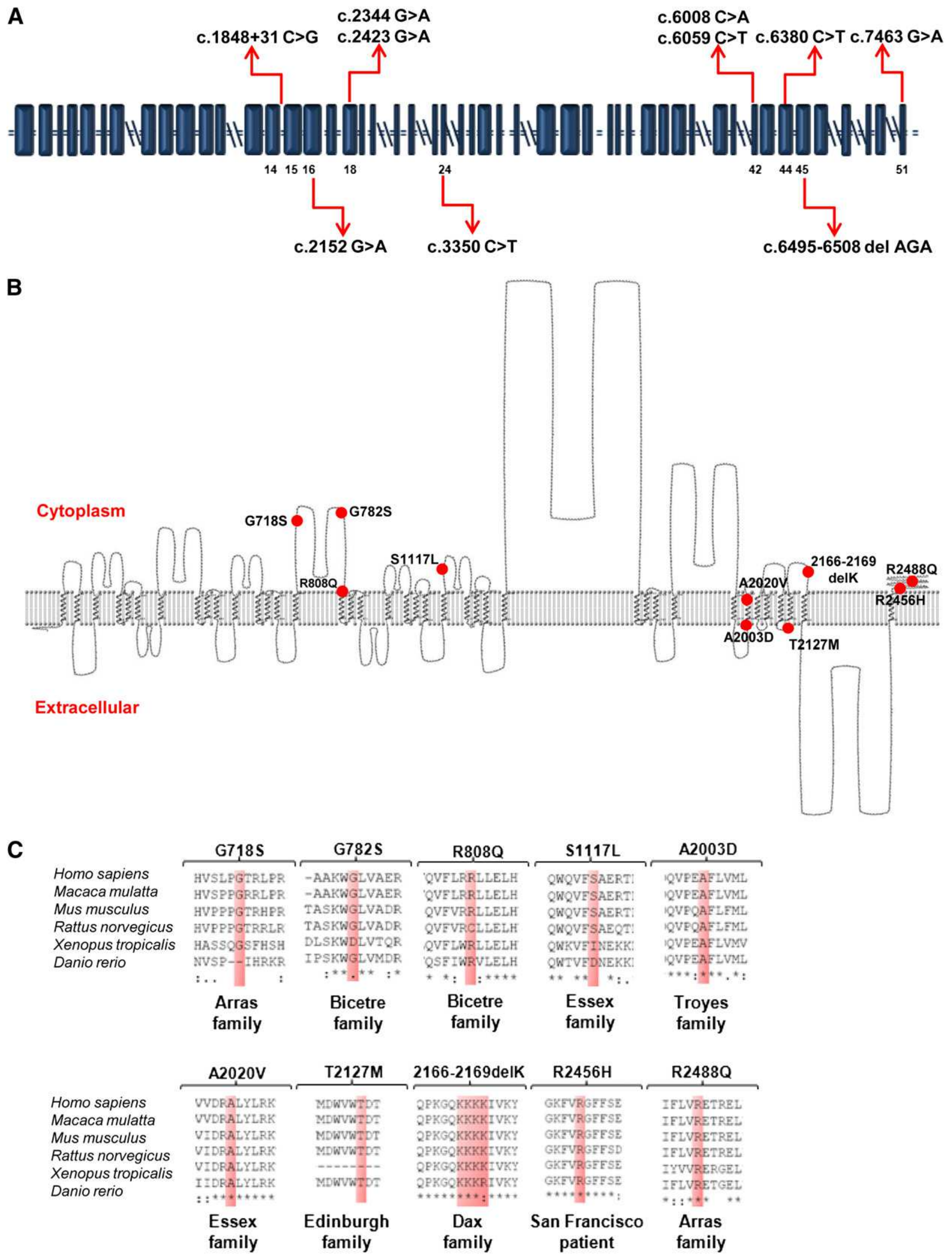


Figure 51. Mutational analysis. (A) Schematic representation of PIEZO1 (blue squares, exons; horizontal double lines, introns; double slashes, large introns); red arrows indicate exonic positions of the nucleotides mutated in the 7 families in whom DHSt was previously mapped to chromosome 16q. (B) A 2-dimensional (2D) hydropathy profile of human PIEZO1 protein. The transmembrane regions of PIEZO1 (UniProt accession Q92508) predicted by TMHMM were displayed using TMRPres2D. Red circles mark approximate locations of DHSt-associated missense mutations. (C) Evolutionary conservation of the residues mutated in our DHSt patients (red shaded boxes) among the species indicated at left.

5.1.3.3. PIEZO1 expression in human and mouse during fetal and embryonic development

PIEZO1 expression has been analyzed in several mouse and human tissues during embryonic development to account for the erythroid involvement of DHSt and for the fluid effusions occurring in some DHSt families. We collected mouse embryos at embryonic day (E) 9.5, E10.5, E12.5, E15.5 and postnatal day 0 (P0; at birth) for quantitative PCR analysis of murine PIEZO1 expression. PIEZO1 messenger RNA (mRNA) abundance increased gradually from E9.5 to E15.5 and was sustained through birth (Figure 52B). We further showed PIEZO1 polypeptide expression in adult RBC membranes from mice (Figure 52C) and humans (Figure 52D).

PIEZO1 immunohistochemical analyses were performed on human fetal tissues (17 weeks of gestation) to verify PIEZO1 expression in liver, spleen, and peritoneum lymphatic vessels. In fetal liver, PIEZO1 showed strong cytoplasmic and membrane signals particularly in hepatic erythroblasts. Fetal spleen at 17 weeks showed positive cytoplasmic staining patterns in splenic plasma cells. PIEZO1 expression in lymphatic vessel of fetal peritoneum at gestational week 17 has been also evaluated to examine the correlation between PIEZO1 expression and occurrence of perinatal edema. PIEZO1 showed a marked signal in lymphatic vessels (Figure 52E). In contrast, PIEZO1 immunoreactivity was absent from peritoneal lymphatic vessels of healthy human adult subjects. This observation provides a first link between PIEZO1 mutations and pre- or perinatal edema.

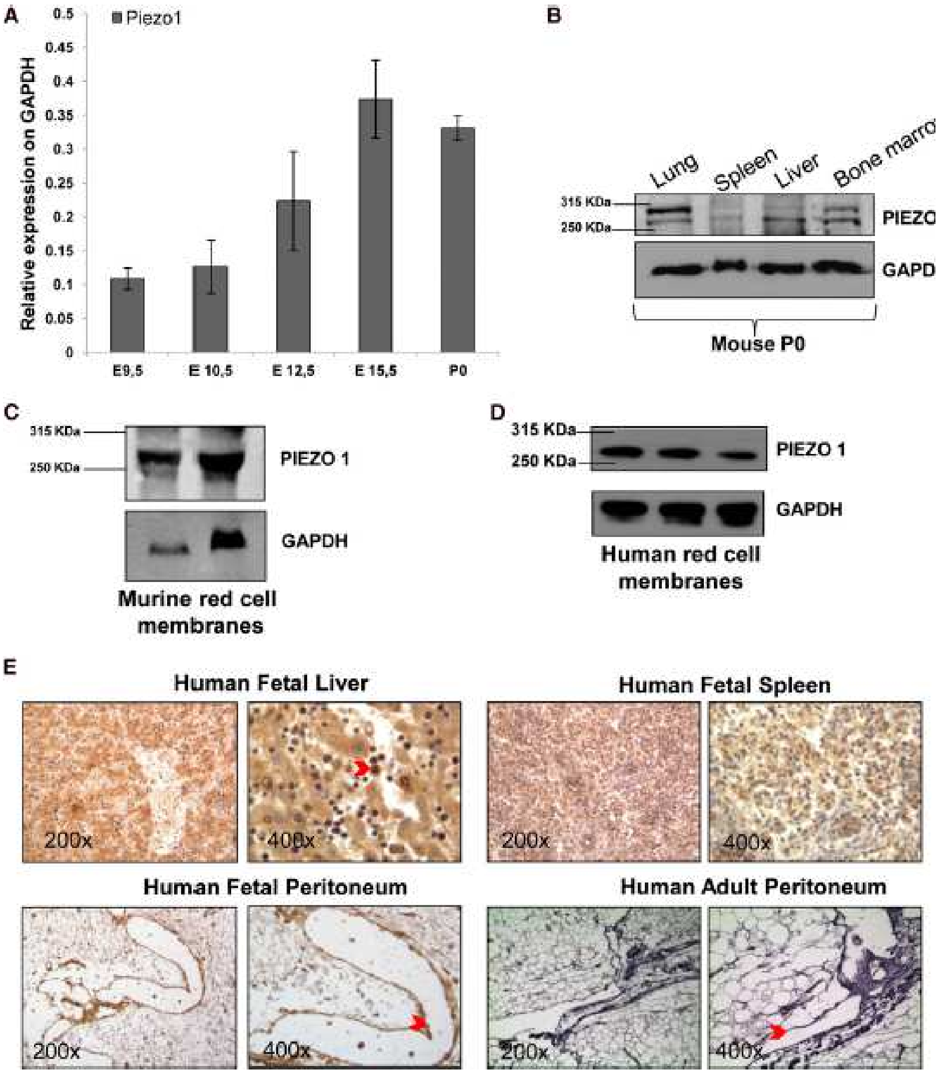


Figure 52.PIEZO1 characterization during mouse and human embryonic development. (A) PIEZO1 mRNA levels (normalized to GAPDH) in murine C57BL/6 embryos at E9.5, E10.5, E12.5, E15.5 and P0. Mean 1 SEM of 3 experiments. (B) Immunoblot showing expression of PIEZO1 protein expression in lung, spleen, liver, and bone marrow from P0 C57BL/6 mouse. Protein (50 mg) was loaded in each lane, with GAPDH as loading control. Representative of 2 independent fresh tissue lysate preparations. (C) Immunoblot showing PIEZO1 protein RBC membranes prepared from blood pooled from 8 adult C57BL/6 mice. Protein (50 or 100 mg) was loaded in each lane, with GAPDH as loading control. One of 3 similar experiments with independent membrane preparations. (D) Immunoblot showing PIEZO1 protein in human RBC membranes prepared from blood pooled from 3 healthy subjects for each lane. Protein (50 mg) was loaded in each lane, with GAPDH as loading control. One of 3 similar experiments with independent membrane preparations. (E) Immunohistochemical expression in human fetal (17 weeks of gestations) and adult tissues with PIEZO1 rabbit polyclonal antibody. The red arrow in the 3400 liver panel indicates a positive erythroblast. The red arrows in the fetal peritoneum panels indicate positive staining in the lymphatic vessels, while in the adult peritoneum panels indicate negative staining in the lymphatic vessels. Antigen is stained brown; nuclei are stained in purple with hematoxylin. Tissues were imaged with a Leica microscope equipped with 203 and 633 objectives. Representative of 3 independent experiments.GAPDH, glyceraldehyde-3-phosphate dehydrogenase.

5.1.3.4. PIEZO1 expression and localization in RBCs

PIEZO1 localization in RBCs was evaluated in healthy controls. Confocal microscopy analysis showed that PIEZO1was expressed on the RBC membrane, as demonstrated by its complete colocalization with the erythroid membrane marker glycophorin A (Figure 53A), and confirmed by colocalization with the membrane marker CD55/DAF. The data confirm previous mass spectrometry data of Zarychanski et al [216]showing the presence of PIEZO1 protein in red cell membranes.

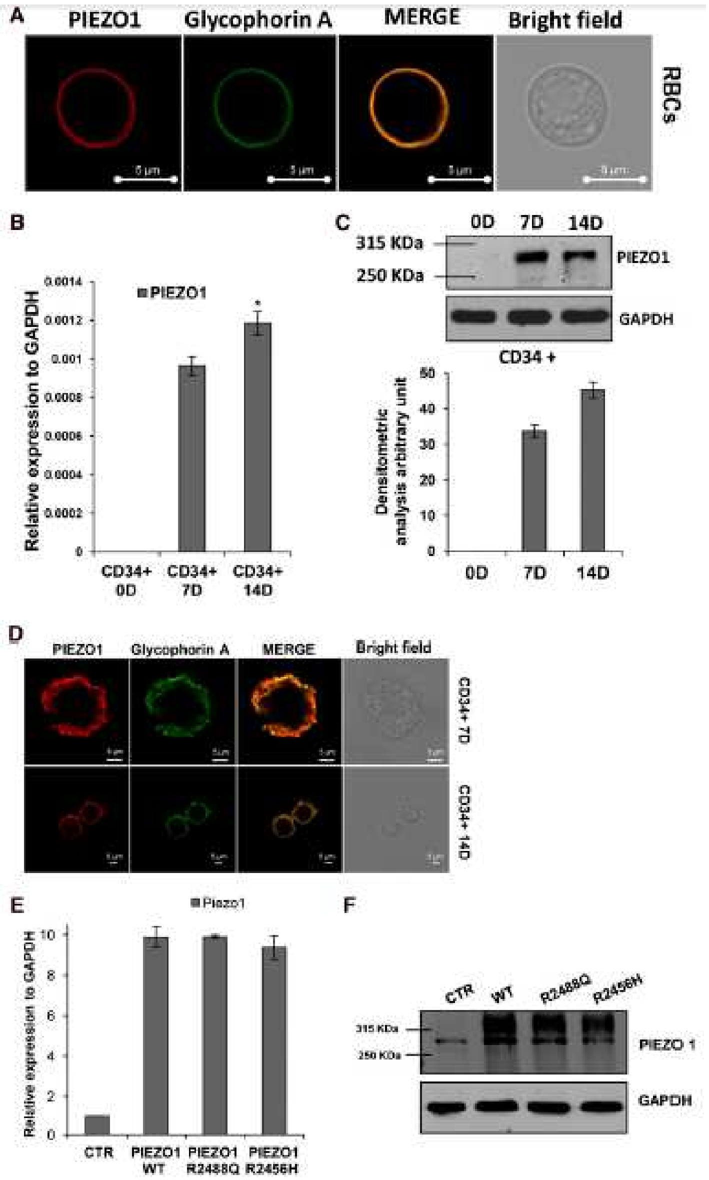


Figure 53.PIEZO1 characterization in RBCs and in CD341 blood cells during erythroid differentiation.

5.1.3.5. PIEZO1 expression during erythroid differentiation

To investigate the role of PIEZO1 in erythroid cells, we first examined PIEZO1 expression and localization in an ex vivo model of erythroid differentiation. CD34$^+$ cells isolated from the peripheral blood of healthy volunteers were induced to erythroid differentiation by 14 days of erythropoietin treatment. As shown in Figure 53B, PIEZO1 mRNA was significantly upregulated after 14 days of erythropoietin treatment ($P = .003$). These data were confirmed at the protein level by western blotting and densitometric analysis (Figure 53C).

We then assessed PIEZO1 protein localization in the same cell systems. As shown in Figure 53D, PIEZO1 colocalized with the plasma membrane marker glycophorin A at 7 and 14 days of CD34$^+$ erythroid differentiation (Figure 53D). No PIEZO1 immunoreactivity was detected in day 0 CD34$^+$ cells (not shown).

5.1.3.6. Expression of WT and mutant PIEZO1 polypeptides inHEK-293 cells

To evaluate expression of the PIEZO1 mutants, we cloned PIEZO1wild-type (WT) and PIEZO1 mutants R2488Q and R2456H inpCMV6-IRES-GFP and transiently transfected the recombinant plasmids into HEK-293 cells.We found that neither mutation impairedPIEZO1 expression at mRNA or protein levels (Figure 53E-F).

5.1.3.7. DHSt red cells exhibit altered ion content and transport

Patient SF red cells heterozygous for the PIEZO1 mutation R2456Hhad elevated Na content of 61 mmol/kg Hb and reduced K content of219 mmol/kg Hb (after overnight shipment).Magnesium (Mg) content was slightly elevated at 9.7 mmol/kg Hb. Red cell activities of K-Cl cotransport, Na-K-2Cl cotransport, and Na/H exchange were not higher than in unrelated control cells (not shown). On-cell patches recorded from patient SF DHSt red cells as described in5.1.2.5(Figure 54A) revealed spontaneous ion-channel activity (lower trace) not detected in cells from an unrelated subject (upper trace). This activity was characterized by a distribution of channel open probability (NPo) of1.47 +0.43 (Figure 54B), a single-channel ohmic conductance of 13single channel conductance (pS) with reversal potential -11 mV ($r^2 = 0.95$; Figure 54C), and was competely blocked by 2.5 mM Grammastola spatulata mechanotoxin-4 (GsMTx4) in the pipet ($P < .05$).

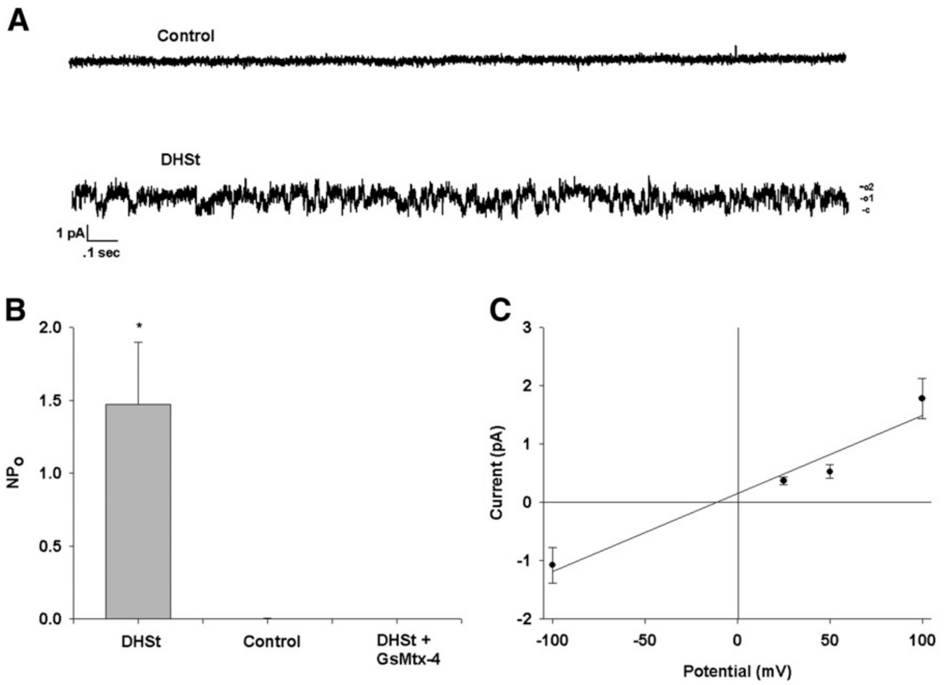


Figure 54. Cation channel activity in on-cell patch recordings of DHSt red cells from patient SF.

5.1.3.8. Function of WT and mutant PIEZO1 expressed in Xenopus oocytes

X laevis oocytes previously injected with hPIEZO1 complementary RNA (cRNA) were subjected after 72 hours to 2-electrode voltage clamp recording. Uninjected oocytes exhibited a small linear current at holding potentials between -100 and +80 mV, with reversal potential of -64 mV (Figure 55A). After 14 minutes of exposure to a moderately hypotonic bath (20% dilution of ND-96 which, based on the low intrinsic oocyte water permeability should produce only minimal swelling), these properties remained essentially unchanged. Oocytes previously injected with PIEZO1 cRNA exhibited slightly elevated currents in ND-96, but with a reversal potential depolarized to -40 mV. However, and in contrast to uninjected oocytes, 14- minute exposure of PIEZO1-expressing oocytes to hypotonic bath conditions substantially increased an outwardly rectifying current, while hyperpolarizing reversal potential to -56 mV (Figure 55A). Exposure of oocytes previously injected with PIEZO1 cRNA to hypertonic bath (ND-96 containing 200 mM mannitol, 15 minutes) also led to increased current (not shown). Because mPIEZO1 exhibits mechanosensitivity, multichannel on-cell patch currents of Xenopus oocytes expressing hPIEZO1 were recorded during voltage ramps before and during application of -25 mm Hg suction via pipet. As shown in Figure 55B, negative pressure did not alter current in uninjected oocytes. In contrast, negative pressure induced increased currents in oocytes expressing PIEZO1 ($P < .005$) and mutant R2488Q ($P < .001$), but not mutant R2456H (Figure 55C). The response of mutant R2488Q to negative pressure appeared to exceed that of WT PIEZO1 ($P = .057$). Occasional oocyte patches allowed resolution of single-channel activity, as illustrated in Figure 56. In these resting state patches, the uninjected oocyte NPo of 0.013 increased to 0.69 in oocytes expressing WT PIEZO1, 0.88 in oocytes expressing PIEZO1 mutant R22488, and 0.22 in oocytes expressing mutant R2456H. Respective single-channel conductances were 25 pS (WT), 26.5 pS (R2488Q), and 43 pS (R24566H).

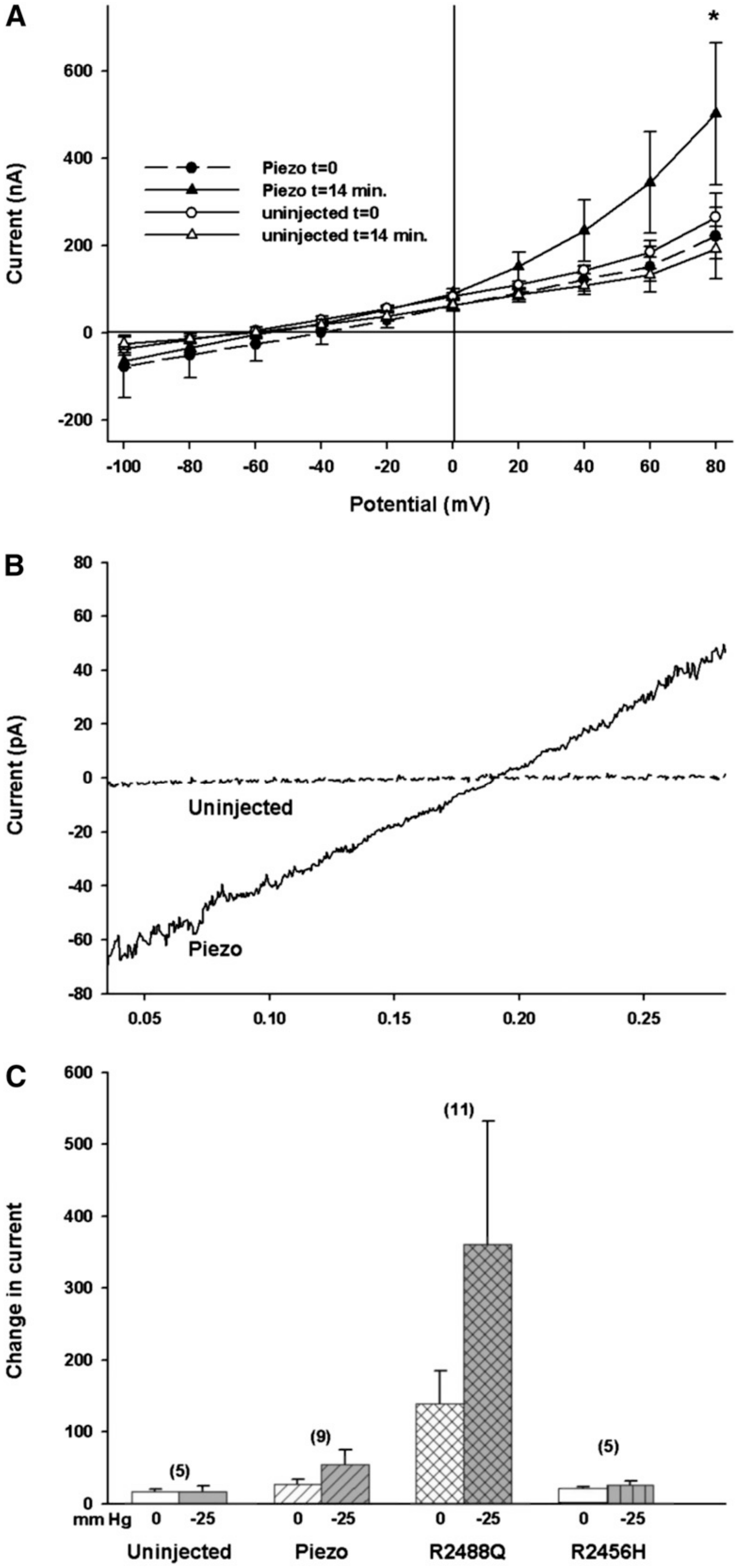


Figure 55. PIEZO1 expressed in Xenopus oocytes confers increased current elicited by hypotonic medium and negative pressure activates currents in on-cell membrane patches of WT PIEZO1 and mutant R2488Q.

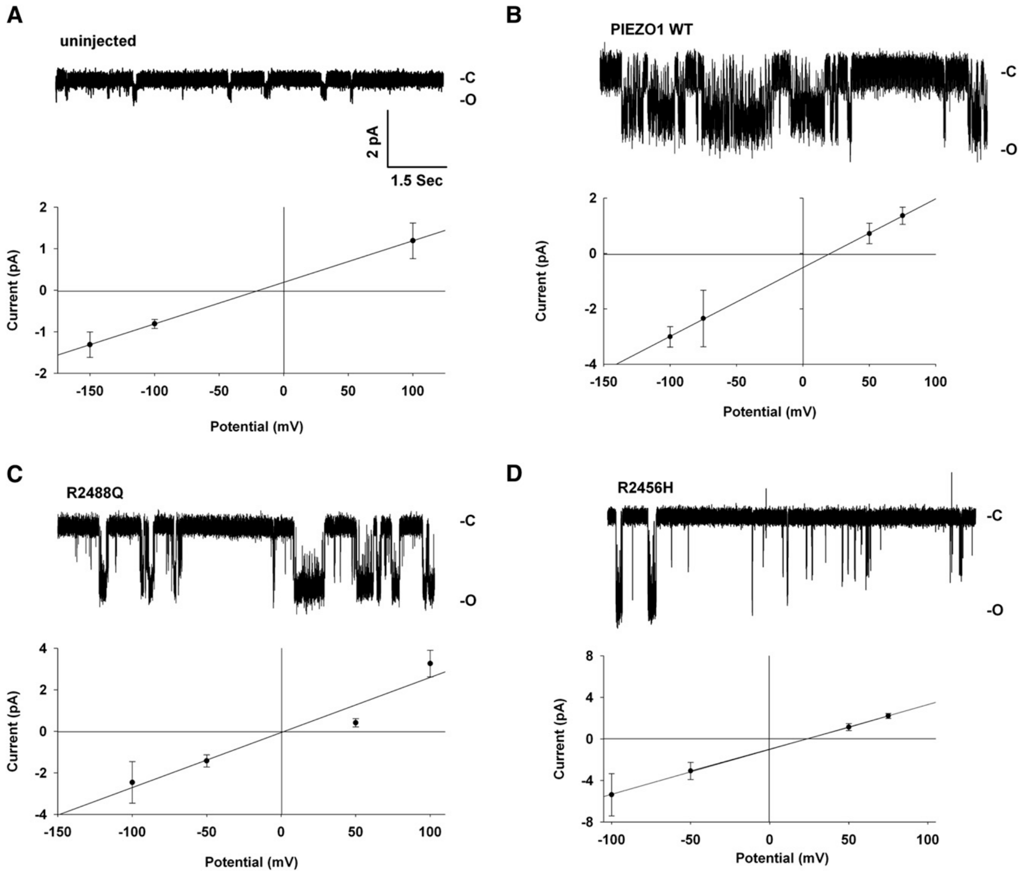


Figure 56. On-cell patch current traces of R2488Q and R2456H mutations in Xenopus oocytes.

5.1.4. Discussion

We have identified PIEZO1 as the causative gene for the varied clinical forms of autosomal dominant DHSt linked to chromosome 16p. PIEZO1 was selected as a strong candidate gene within the critical region previously mapped to 16q23-qter based on exome sequencing analysis in family Edinburgh. Subsequent targeted sequencing analysis identified several additional novel PIEZO1 mutations in 7 families with DHSt syndromes. PIEZO1 protein expression was characterized during human and murine development and during erythroid differentiation. Functional studies demonstrated for the first time that PIEZO1 mutations cause altered ion transport in erythroid cells. PIEZO1 protein was also detected in fetal lymphatic vessel endothelium, consistent with its proposed causative role in the pathogenesis of perinatal effusions. Electrophysiology analysis in oocytes demonstrated changes in ion transport consistent with the altered ion content of DHSt patient red cells.

The PIEZO1 open reading frame was first found in the human immature myeloid cell line KG-1, and transcript tissue profiles showed apparently ubiquitous expression [223]. Satoh et al demonstrated transcriptional induction of PIEZO1 in senile plaque-associated astrocytes from Alzheimer disease patients [224]. PIEZO1 involvement in integrin activation requires recruitment to the endoplasmic reticulum of the small GTPase R-Ras, promoting release of Ca^{2+} from intracellular stores to activate cytoplasmic calpain. Recently, PIEZO1 and PIEZO2 were both implicated in mechanosensation as stretch-activated cation channels [225]. Soon thereafter, expression in human cells of the single Drosophila melanogaster PIEZO (DmPIEZO or CG8486) was shown to resemble its mammalian counterparts in its ability to induce mechanically activated currents [226]. Behavioral responses to noxiousmechanical stimuli were severely reduced in DmPIEZO knockout larvae, whereas responses to light touch or to other types of noxious stimulus were unaffected. Human PIEZO1 is an N-linked glycoprotein that serves as substrate for both acetylation and phosphorylation [227]. Coste et al further showed that mouse PIEZO (MmPIEZO1) can assemble as a 1.2-megadalton homo-oligomer with a total of 120 to 160 transmembrane segments, the largest homomeric plasma membrane ion-channel complex identified to date [228]. Purified MmPIEZO1 reconstituted into asymmetric lipid bilayers and liposomes forms ruthenium-red-sensitive ion channels in the absence of any other protein.

PIEZO1 expression has been characterized during mouse and human development. We have shown that PIEZO1 expression increased during murine embryogenesis and, at birth, expression was predominant in liver and bone marrow. In fetal human tissues, PIEZO1 showed marked expression in liver and spleen. Of note, PIEZO1 was expressed in lymphatic vessels of the fetal peritoneum and was absent in adult lymphatic vessels, demonstrating for the first time a potential physiological link between PIEZO1 mutations and the perinatal edema that sometimes accompanies DHSt.

PIEZO1 expression has been also identified and localizated in the plasma membrane of RBCs, confirming immunologically the original mass spectroscopic identification of PIEZO1 as part of the red cell membrane proteome [229], and its subsequent detection in red cell membrane by targeted mass spectrometry.

In 3 unrelated families, we found multiple in cis missense mutations in PIEZO1. R2488Q mutation in family Arras altered a residue conserved in all analyzed species (Figure 51), and the linked mutation G718S altered a residue conserved in all tested species except D rerio. A2020V mutation in family Essex altered a completely conserved residue, and linked mutation S1117L altered a residue conserved in all tested species except X tropicalis and D rerio. The linked variants at sites of lesser evolutionary conservation are not, however, present among normal alleles and SNV databases, and so may not be harmless variants. Further studies will elucidate the origin of geographic clustering of these mutations. However, the contribution of each individual mutation to its linked clinical phenotype cannot yet be assigned. At this time, we do not know which one of the mutations coinherited in cis might be individually responsible for the disease phenotype, or whether disease arises from the combined effects of the coinherited mutations. Interestingly, the 2 families carrying 2 novel, linked mutations exhibited the phenotype of DHSt plus pseudohyperkalemia, but affected individuals in the single family characterized by 3 allelic mutations exhibited a more complex phenotype of DHSt plus perinatal edema and pseudohyperkalemia. The connections linking genotype to the red cell dehydration phenotype and to phenotypic variability could reflect mutation position within the 3-dimensional structure of the PIEZO1 polypeptide (allelic heterogeneity) and/or modifier gene coinheritance.

Patient SF exhibited the same mutation, R2456H, found in one of the families reported by Zarychanski et al. Our patient and the previously reported R2456H patients showed a similar phenotype characterized by DHSt unaccompanied by pseudohyperkalemia or perinatal edema. In contrast to the report of Zarychanski and colleagues, our families exhibited only heterozygous mutations, as predicted for a simple pattern of dominant inheritance.

The presence of PIEZO1 in the red cell membrane suggests a link to the erythroid ion imbalance and altered erythroid ionchannel activity of DHSt patients. Functional studies in Xenopus oocytes demonstrated that WT PIEZO1 expression increased 2- electrode voltage clamp current elicited by osmotic swelling, and channel activity in cell-attached patches. The PIEZO1 mutation R2488Q increased hydrostatic pressure-induced currents in on-cell patches of Xenopus oocytes, likely reflecting, in part, increased NPo. Oocytes expressing PIEZO1 mutant R2456H exhibited cell-attached patch currents of elevated single-channel conductance. These properties of oocytes expressing PIEZO1 mutants are consistent with the steady-state elevation of intracellular Na+ and reduction of intracellular K+ that characterize red cells of DHSt patients. However, the link between PIEZO1 mutations and the combination of elevated MCV and mean corpuscolar Hb concentration, which underlies the descriptor “dehydrated stomatocytosis”, will require further experimentation. Further experiments will be needed to confirm individual mutation-selective changes in single-channel characteristics suggested by the present data. Detailed comparison of functional effects of endogenous WT and mutant PIEZO1 in intact red cells with those of heterologous WT and mutant PIEZO1 expressed in Xenopus oocytes will also require additional experiments. These will be directed toward a greater understanding of differences between the chronic effects and influences of WT and mutant PIEZO1 channels on cell volume and ion content and the rapid kinetics of PIEZO1 channels recorded in whole-cell and patch modes. Such differences are influenced by the still incompletely defined changes in membrane tension and cytoskeletal dynamics inside the patch pipet containing the distinct plasma membranes of fetal erythrocytes, adult erythrocytes, and Xenopus oocytes. All of these, in turn, likely differ from the strains experienced by the membranes of intact red cells over their normal 120-day lifespan as they experience a range of laminar and turbulent shear stresses during their circulation through vessels ranging in diameter from the ventricular chamber and the aorta to capillary tortuosities, and accompanied by sequential adhesions to and releases from other blood cells and endothelial cells.

In conclusion, DHSt is a pleiotropic syndrome caused by dominant PIEZO1 mutations. In particular, R2456H and R2488Q mutations in PIEZO1 likely alter mechanosensitive channel regulation, leading to increased cation transport in erythroid cells. Ongoing functional analysis should further elucidate the pathogenic mechanisms of all PIEZO1 mutations found in simple and syndromic forms of DHSt.

5.2. Improving molecular diagnosis in epilepsy by a dedicated high-throughput sequencing platform

We analyzed by next-generation sequencing (NGS) 67 epilepsy genes in 19 patients with different types of either isolated or syndromic epileptic disorders and in 15 controls to investigate whether a quick and cheap molecular diagnosis could be provided. The average number of nonsynonymous and splice site mutations per subject was similar in the two cohorts indicating that, even with relatively small targeted platforms, finding the disease gene is not an univocal process. Our diagnostic yield was 47% with nine cases in which we identified a very likely causative mutation. In most of them no interpretation would have been possible in absence of detailed phenotype and familial information. Seven out of 19 patients had a phenotype suggesting the involvement of a specific gene. Disease-causing mutations were found in six of these cases. Among the remaining patients, we could find a probably causative mutation only in three. None of the genes affected in the latter cases had been suspected a priori. Our protocol requires 8–10 weeks including the investigation of the parents with a cost per patient comparable to sequencing of 1–2 medium-to-large-sized genes by conventional techniques. The platform we used, although providing much less information than whole-exome or whole genome sequencing, has the advantage that can also be run on 'benchtop' sequencers combining rapid turnaround times with higher manageability.

5.2.1. Introduction

Epilepsy is one of the most common neurological disorders in humans with a prevalence of 1% and a lifetime incidence of up to 3%[230]. Epilepsies present with a broad range of clinical features and their genetic causes remain unknown in the vast majority of cases, although several genes have been identified in rare Mendelian forms, either heritable or sporadic. Finding the disease genes can be challenging, as the same epileptic phenotype may be associated with several genes. A molecular diagnosis of epilepsy is important especially in a pediatric setting in order to (1) establish the recurrence risk in following pregnancies, (2) stop the diagnostic odyssey that is frequently restless for undiagnosed epilepsies, and (3) provide, at least in some cases, specific therapies. Recently, genomewide association studies revealed a few regions harboring high-ranking candidate genes, although these studies still necessitate further replication efforts[231]. Genomic arrays had been more successfully, as they allowed identifying several possible pathogenic copy-number variants not present in controls in about 9% of the cases[232]. Presently, high-throughput sequencing is becoming the most promising approach to improve molecular diagnosis of this condition, although the interpretation of the results is far from being a standardized process. Indeed, next-generation sequencing (NGS) does not magically make diagnoses but typically provides a handful of possibilities requiring further studies on the function of each candidate gene. To overcome these problems, we composed a panel containing most epilepsy genes, covering several relevant phenotypes. With this NGS platform, we studied 19 index patients suffering from a range of seizures, either familial or sporadic. Although initially we performed a blind study trying to interpret the sequencing data without any knowledge of the clinical history, we then realized that no analysis was possible in absence of detailed phenotypes and familial information. This study allowed us to evaluate not only the diagnostic capability of this approach but also the cost and the time required to report the final result to the family.

5.2.2. Materials and Methods

5.2.2.1. Patient Cohort

The 19 index cases ranged from few days to 4 years of age at the time of the first clinical examination. Most of them have been then followed-up for several years. This cohort has been randomly selected from patients afferent to our epileptic center for children and adolescents. All available family members have been enrolled for segregation analysis. Informed consent was obtained from each family and clinical evaluation and genetic testing were carried out in accordance with the ethics approval granted (11017C-RC2011 IRCCS C. Mondino, Diagnosis and therapy of epileptic syndromes).

Each patient has come along clinical diagnosis and history, presence of the epilepsy in relatives (family history), electroencephalograms, magnetic resonance findings, and administered antiepileptic drugs. In order to assess the diagnostic capability of our approach, we collected patients presenting with a wide range of epilepsy phenotypes: 11 are sporadic cases, whereas the remaining eight have a history of familial epilepsy.

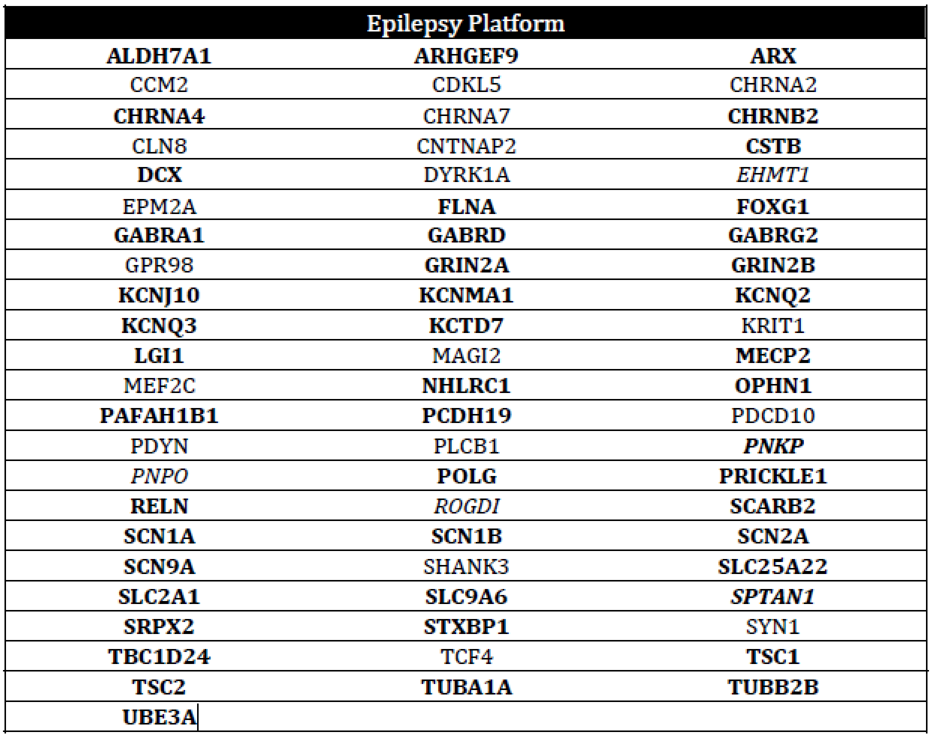
Patients have been subdivided in two groups: the first one, subcohort A, was constituted by seven patients whose clinical features were either strongly or more loosely suggestive for a syndrome associated with a specific gene; the second group, subcohort B, included 12 subjects with very different types of epilepsy, presumably heterogeneous in their genetic basis and not suggestive of any or a single specific gene. In both subgroups other clinical features such as language impairment, psychomotor delay, or autism spectrum disorder were present in several patients. Magnetic resonance abnormalities were detected in some of them. All cases had been previously analyzed by array comparative genomic hybridization and a few (6-A, 8-B(i), 8-B(ii)) by Sanger Sequencing for specific genes without any positive result.

5.2.2.2. Control Cohort

Control subjects (nine females and six males), ranging from 18 to 35 years of age, were recruited among blood donors as controls for both this and a cardiovascular study. Besides, they were requested to answer a structured general medical questionnaire with specific emphasis on neurological and cardiac symptoms, control subjects had to answer two specific questions: (1) have you ever suffered from epilepsy or seizures, and (2) have you become acquainted with any seizure disorders or EEG abnormalities present in some of your family members? Only those who answered negatively were recruited.

5.2.2.3. Platform Design

A custom-designed target enrichment library for 67 genes has been designed by using the Agilent eArray website (https://earray.chem.agilent.com/earray/). This library contains unique baits covering the exons, the UTRs, and the intron–exon junctions of the selected genes. The estimated base coverage of the library is 0.45Mb. The selection was made on the basis of the following criteria: (1) genes associated with idiopathic epilepsy; (2) genes associated with syndromic epilepsy; (3) genes associated with epilepsy and cerebral malformations excluding holoprosencephaly; (4) genes that appeared to be the best candidates for epilepsy in microdeletion syndromes. Selected genes are reported in Figure 57.



Epilepsy Platform		
ALDH7A1	**ARHGEF9**	**ARX**
CCM2	CDKL5	CHRNA2
CHRNA4	CHRNA7	**CHRNB2**
CLN8	CNTNAP2	**CSTB**
DCX	DYRK1A	*EHMT1*
EPM2A	**FLNA**	**FOXG1**
GABRA1	**GABRD**	**GABRG2**
GPR98	**GRIN2A**	**GRIN2B**
KCNJ10	**KCNMA1**	**KCNQ2**
KCNQ3	**KCTD7**	KRIT1
LGI1	MAGI2	**MECP2**
MEF2C	**NHLRC1**	**OPHN1**
PAFAH1B1	**PCDH19**	PDCD10
PDYN	PLCB1	***PNKP***
PNPO	**POLG**	**PRICKLE1**
RELN	*ROGDI*	**SCARB2**
SCN1A	**SCN1B**	**SCN2A**
SCN9A	SHANK3	**SLC25A22**
SLC2A1	**SLC9A6**	***SPTAN1***
SRPX2	**STXBP1**	SYN1
TBC1D24	TCF4	**TSC1**
TSC2	**TUBA1A**	**TUBB2B**
UBE3A		

Figure 57. Epilepsy genes platform

5.2.2.4. Sample Preparation

DNA (5 µg) extracted from peripheral blood by standard methods were diluted in 700 µl of nebulization buffer (Illumina, San Diego, CA, USA) and sheared using a nebulization technique (Invitrogen, Carlsbad, CA, USA), which breaks up DNA into pieces < 500 bp, through the application of 60–70 psi (pound force per square inch) of purified air for 4min. This process generates double-stranded DNA fragments containing 3’ or 5’ overhangs that were cleaned up using QIAquick spin columns (Qiagen, Hilden, Germany). A quality control step on the recovered DNA was then performed using Nanodrop 1000 to quantify the DNA by a 260-nm reading and Bioanalyzer 2100 (Agilent, Santa Clara, CA, USA) to check the size of the fragments.

According to the Agilent SureSelectXT protocol, sheared DNA overhangs were end-repaired and then purified using the magnetic bead-based Agencourt AMPure XP purification system (Beckman Coulter Genomics, Brea, CA, USA). Then we performed the adding of ‘A’ bases to the 3’ end of the DNA fragments and the ligation of indexing-specific paired-end adapters.

After a few cycles of PCR amplification, 500 ng of DNA from the resulting libraries were hybridized to the bait set using the SureSelectXT MP Capture Library Kit (Design no. 5190-0312931—Agilent) at 65°C for 24 h. Hybrids capture was performed according to the manufacturer’s protocol with Streptavidin-coated Dynal magnetic beads (Invitrogen). Captured samples were further purified through Agencourt AMPure XP beads and subjected to a PCR-based amplification reaction to add index tags (each is a sequence of six bases in length allowing to identify samples after pooling), accordingly to the Agilent SureSelectXT protocol. For each step of library preparation, all samples were quantified on a Bioanalyzer 2100 (Agilent). We performed a multiplexed run on the Illumina Genome Analyzer IIx, where nine multiple samples were sequenced in a single lane of a flow cell; number of samples to be pooled has been calculated on the basis of the enriched target’s size, according to Agilent’s instructions. The sample libraries from nine individuals were denatured with NaOH and loaded on a single lane of a Illumina Flowcell v4 where DNA clusters were generated through a one-step workflow (according to Illumina protocol) on the Cluster Station using TruSeq PE Cluster Kit v5 (Illumina).

One percent volume of a PhiX control library (Illumina) was used as internal control and loaded in each lane of the flowcell.

The capture was considered successful if at least 99% of our target regions were covered by more than eight reads of high quality (that is a Phred-scaled mapping quality score of at least 20 for each).

5.2.2.5. Annotation and interpretation of data

Sequences for each sample were generates by Illumina software CASAVA v1.8.1. Reads were filtered by quality relying on the standard Illumina quality filter test. Reads were aligned to the most recent version of human genome (GRCh37/hg19) using BWA software package v0.6.07 and filtered out for PCR duplicates by Samtools v0.1.18. Reads were realigned around inferred indels and their base qualities were recalibrated taking into account the context of alignment by the Genome Analysis Toolkit (GATK) v1.6 suite. SNPs and short indels were called using GATK UnifiedGenotyper module and the resulting variants were filtered using GATK Variant Filtration module and specific Perl scripts, as such variants were probably owing to alignment errors and, in general, they cannot be considered reliable variants. Several filtering constraints were also applied, such as minimum variant quality (50, Phred scaled), a minimum of five reads supporting variant, number of ambiguous mapped reads overlapping variant, neighborhood of each reliable indel or homopolymer excluding those single-nucleotide variations that overlap within.

Variants annotation was performed on the resulting data set by in-house genomic database application (the RDBVMS system). Prediction tracks for each annotated variant were generated by automatic remote calling procedures to MutationTaster and Polyphen-2 (version 2.2.2).

In order to identify potential causative mutations, we applied the so-called discrete filtering approach. We first excluded synonymous out of target and UTR-overlapping variants. We then excluded the variants present in dbSNP135 and Exome Sequencing Project Databases (ESP) with a frequency higher than 1%. Moreover, we discarded variants reported in our in-house database (66 whole exomes) that were identified in at least two individuals without epilepsy or other neurological disorders. We then took into account only variants predicted to alter the protein structure or function by at least one of the three prediction tools we used (Mutation Taster, SIFT, Polyphen2) as well as variants for which all prediction tools failed. At the end, we excluded all variants occurring in at least three cases and/or at least two subjects of the control cohort.

We prioritized the candidate alterations on the base of the expression and function of the altered gene, the type of mutation and its effect at protein level, presence of the variant in the Human Gene Mutation Database (HGMD) or in the literature, the metabolic pathway involved, and obviously the clinical features of the patient/family. A manual inspection of the variants eliminated by the prediction tools filtering step allowed us to reconsider them on the base of possible correlations with the patients' phenotype. For example, this permitted reconsidering a variant of ALDH7A1, which was then ascertained as causative.

The entire protocol of data analyses is illustrated in the flowchart reported in Figure 58.

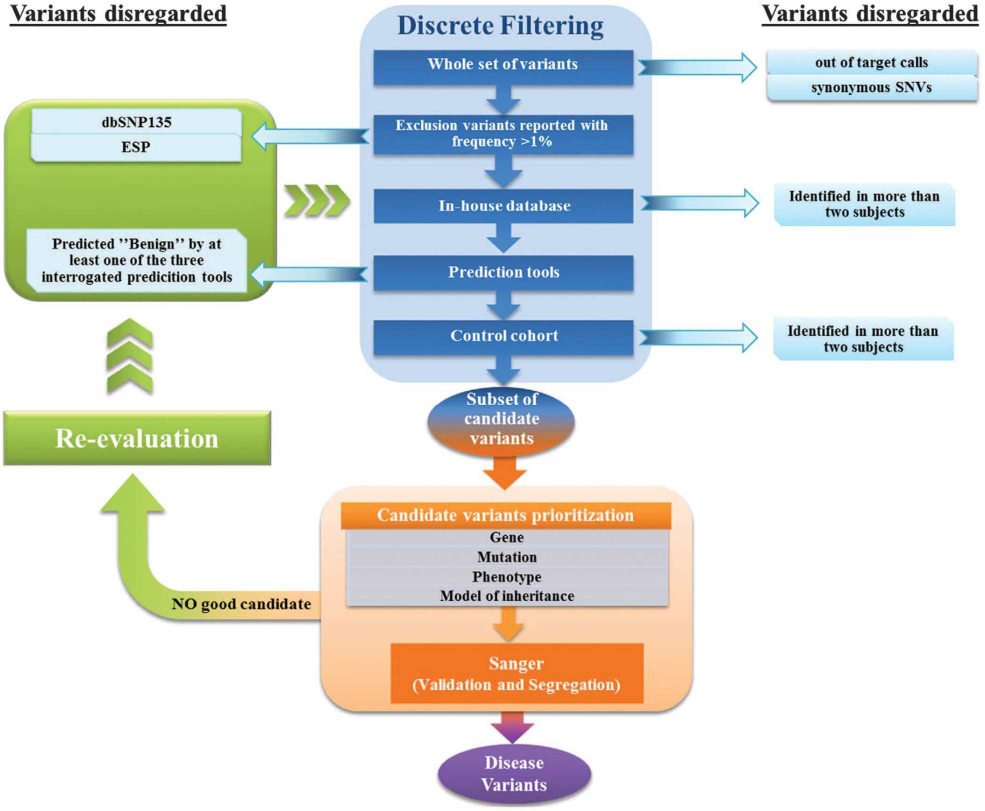

Variants disregarded
Discrete Filtering
Variants disregarded
Whole set of variants
out of target calls
synonymous SNVs
dbSNP135
ESP
Exclusion variants reported with frequency >1%
In-house database
Identified in more than two subjects
Predicted ''Benign'' by at least one of the three interrogated predicition tools
Prediction tools
Control cohort
Identified in more than two subjects
Subset of candidate variants
Re-evaluation
Candidate variants prioritization
Gene
Mutation
Phenotype
Model of inheritance
NO good candidate
Sanger
(Validation and Segregation)
Disease Variants

Figure 58.Flow chart representing the strategy adopted to analyze sequencing data. Discrete filtering, prioritization, and re-evaluation steps are highlighted inblue, orange and green, respectively.

The final subset of mutations was confirmed by Sanger sequencing followed by segregation analysis in each family. Only a close collaboration with the specialist allowed us to find a specific genotype–phenotype correlation discarding those variants that were either pathogenic in recessive state in families where the condition segregated in a dominant manner or those that correlated to a neurological phenotype totally different from that of the patient. This point was taken with extreme caution, as it could not be excluded, a priori, that a novel mutation might cause a totally different phenotype with respect to the ones known to be associated with the same gene. Some of the remaining alterations, even if predicted damaging by at least one tool, have been discarded when they did not segregate with the epileptic phenotype in familial cases (i.e., MAGI2 in case 5-A that was inherited by the healthy mother, whereas KCNQ2 was considered causative because it was inherited by the affected father).

We applied the same filtering strategies to the control cohort in order to perform a statistical test to assess whether the number of deleterious variants in cases was significantly higher than in controls.

5.2.2.6. Deleterious variants: Cases vs Controls

To assess the significance of difference in the number of variants between patient and control cohorts we applied the non-parametric statistical hypothesis test of Wilcoxon-Mann-Whitney (WMW).

The choice of this statistical test is justified by the assumption of independence of all observations (number of variants) in both the two cohorts and therefore by their discrete and ordinal nature.

We performed the WMW test using R v.2.15.1 typing the following command:

```
wilcox.test(Cases,Controls,correct=TRUE,conf.level=0.95)
```

Where Cases and Controls are two arrays containing number of variants for patient and control groups respectively, continuity correction factor is applied and the significance level is set to 0,05. The test was performed in two-sided and one-sided way according to test the significance of difference in number of variants between cases and controls before and after discrete filtering respectively.

Because of the discrete nature of our observations we applied ties correction for standard deviation (s1.1) to be sure that the presence of ex-aequo observations was not affecting results.

.

$$Z = \frac{(T \pm 0{,}5) - \mu_T}{\sigma_T}, \tag{1.1}$$

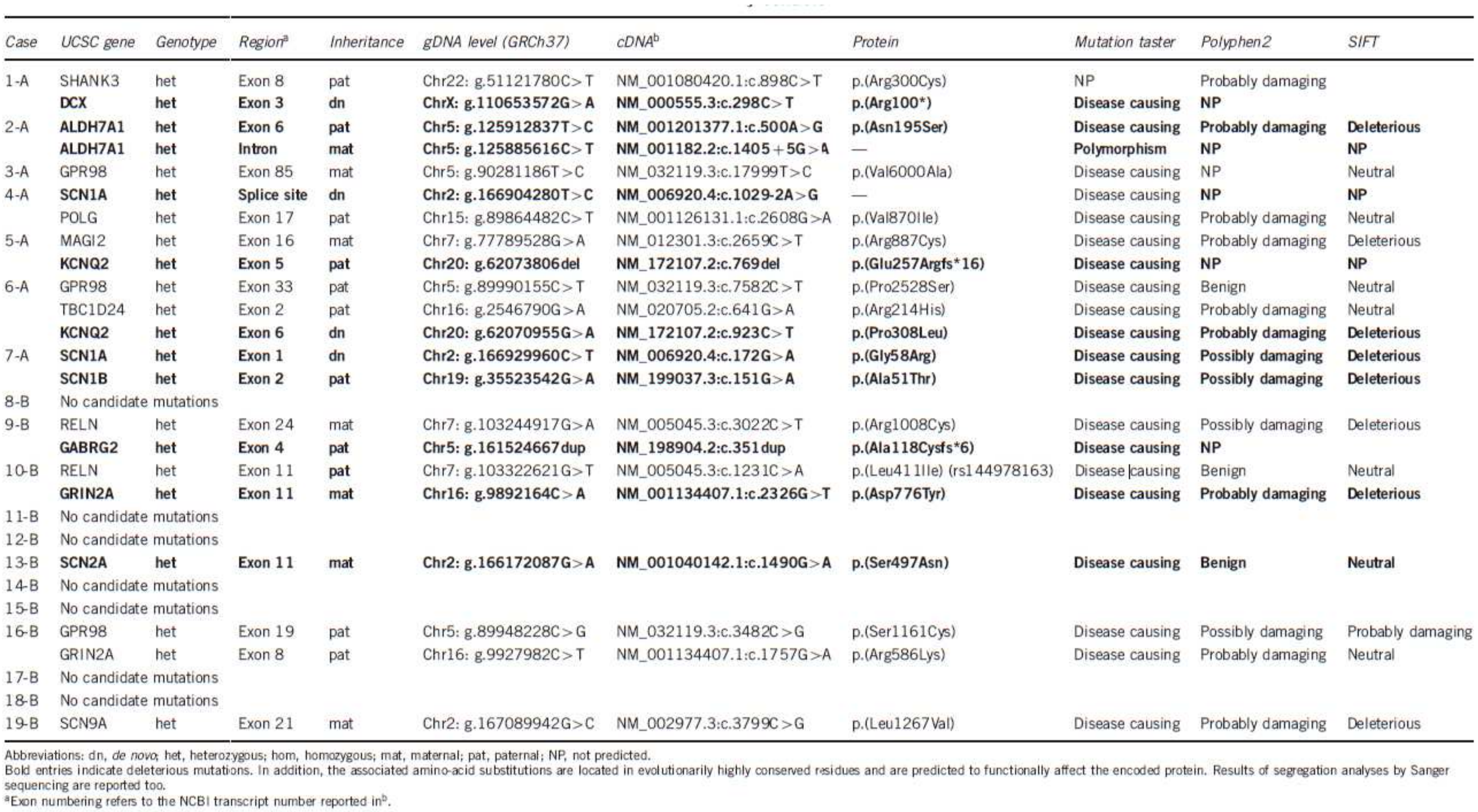
$$\mu_T = \frac{N_1 \cdot (N_1 + N_2 + 1)}{2},$$

$$\sigma_T = \sqrt{\frac{N_1 \cdot N_2}{N \cdot (N-1)} \cdot \left(\frac{N^3 - N}{12} - \sum_{j=1}^{g} \frac{t_j^3 - t_j}{12} \right)} \; .$$

- $N_1, N_2$are the number of patients and controls respectively
- $N = N_1 + N_2$
- $g$is the number of ties
- $t$is the number of observations with the same rank within each tie

5.2.3. Results

Targeted massive parallel sequencing of patient and control cohorts produced for each subject about 180Mb of sequence, which yielded an average coverage of about 400x at each targeted base. On average, 96% of target bases exceeded the 15x coverage threshold required for confident analysis, defined as 99% power to detect a variant. We compared the number of variants per subject between patients and controls using the non-parametric statistical hypothesis test of WMW discussed in the previous section. We did not find any significant difference between the two cohorts after filtering (Pvalue = 0.4928). This might be owing to the limited number of subjects analyzed by this platform. Figure 59 reports all the variants remaining after the filtering and prioritization processes (ranging from one to three per subject). In bold are highlighted those variants considered as having the main effect on the patient's phenotype. By this approach we were able to identify candidate SNVs, very likely causative of the epileptic phenotype, in nine out of 19 patients, six of which belong to subcohort A and three belonging to subcohort B previously discussed (seeFigure 60). In the remaining 10 cases, we could not find any potential causative alterations even after a careful re-evaluation by manual inspection of variants filtered out. Causative mutations were validated by Sanger sequencing (data not shown). All variants thereafter reported have been submitted to Leiden Open Variation Database 3.0[233].



Case	UCSC gene	Genotype	Region[a]	Inheritance	gDNA level (GRCh37)	cDNA[b]	Protein	Mutation taster	Polyphen2	SIFT
1-A	SHANK3	het	Exon 8	pat	Chr22: g.51121780C>T	NM_001080420.1:c.898C>T	p.(Arg300Cys)	NP	Probably damaging	
	DCX	**het**	**Exon 3**	**dn**	**ChrX: g.110653572G>A**	**NM_000555.3:c.298C>T**	**p.(Arg100*)**	**Disease causing**	**NP**	
2-A	**ALDH7A1**	**het**	**Exon 6**	**pat**	**Chr5: g.125912837T>C**	**NM_001201377.1:c.500A>G**	**p.(Asn195Ser)**	**Disease causing**	**Probably damaging**	**Deleterious**
	ALDH7A1	**het**	**Intron**	**mat**	**Chr5: g.125885616C>T**	**NM_001182.2:c.1405+5G>A**	—	**Polymorphism**	**NP**	**NP**
3-A	GPR98	het	Exon 85	mat	Chr5: g.90281186T>C	NM_032119.3:c.17999T>C	p.(Val6000Ala)	Disease causing	NP	Neutral
4-A	**SCN1A**	**het**	**Splice site**	**dn**	**Chr2: g.166904280T>C**	**NM_006920.4:c.1029-2A>G**	—	Disease causing	**NP**	**NP**
	POLG	het	Exon 17	pat	Chr15: g.89864482C>T	NM_001126131.1:c.2608G>A	p.(Val870Ile)	Disease causing	Probably damaging	Neutral
5-A	MAGI2	het	Exon 16	mat	Chr7: g.77789528G>A	NM_012301.3:c.2659C>T	p.(Arg887Cys)	Disease causing	Probably damaging	Deleterious
	KCNQ2	**het**	**Exon 5**	**pat**	**Chr20: g.62073806del**	**NM_172107.2:c.769del**	**p.(Glu257Argfs*16)**	**Disease causing**	**NP**	**NP**
6-A	GPR98	het	Exon 33	pat	Chr5: g.89990155C>T	NM_032119.3:c.7582C>T	p.(Pro2528Ser)	Disease causing	Benign	Neutral
	TBC1D24	het	Exon 2	pat	Chr16: g.2546790G>A	NM_020705.2:c.641G>A	p.(Arg214His)	Disease causing	Probably damaging	Neutral
	KCNQ2	**het**	**Exon 6**	**dn**	**Chr20: g.62070955G>A**	**NM_172107.2:c.923C>T**	**p.(Pro308Leu)**	**Disease causing**	**Probably damaging**	**Deleterious**
7-A	**SCN1A**	**het**	**Exon 1**	**dn**	**Chr2: g.166929960C>T**	**NM_006920.4:c.172G>A**	**p.(Gly58Arg)**	**Disease causing**	**Possibly damaging**	**Deleterious**
	SCN1B	**het**	**Exon 2**	**pat**	**Chr19: g.35523542G>A**	**NM_199037.3:c.151G>A**	**p.(Ala51Thr)**	**Disease causing**	**Possibly damaging**	**Deleterious**
8-B	No candidate mutations									
9-B	RELN	het	Exon 24	mat	Chr7: g.103244917G>A	NM_005045.3:c.3022C>T	p.(Arg1008Cys)	Disease causing	Possibly damaging	Deleterious
	GABRG2	**het**	**Exon 4**	**pat**	**Chr5: g.161524667dup**	**NM_198904.2:c.351dup**	**p.(Ala118Cysfs*6)**	**Disease causing**	**NP**	
10-B	RELN	het	Exon 11	**pat**	Chr7: g.103322621G>T	NM_005045.3:c.1231C>A	p.(Leu411Ile) (rs144978163)	Disease causing	Benign	Neutral
	GRIN2A	**het**	**Exon 11**	**mat**	**Chr16: g.9892164C>A**	**NM_001134407.1:c.2326G>T**	**p.(Asp776Tyr)**	**Disease causing**	**Probably damaging**	**Deleterious**
11-B	No candidate mutations									
12-B	No candidate mutations									
13-B	**SCN2A**	**het**	**Exon 11**	**mat**	**Chr2: g.166172087G>A**	**NM_001040142.1:c.1490G>A**	**p.(Ser497Asn)**	**Disease causing**	**Benign**	**Neutral**
14-B	No candidate mutations									
15-B	No candidate mutations									
16-B	GPR98	het	Exon 19	pat	Chr5: g.89948228C>G	NM_032119.3:c.3482C>G	p.(Ser1161Cys)	Disease causing	Possibly damaging	Probably damaging
	GRIN2A	het	Exon 8	pat	Chr16: g.9927982C>T	NM_001134407.1:c.1757G>A	p.(Arg586Lys)	Disease causing	Probably damaging	Neutral
17-B	No candidate mutations									
18-B	No candidate mutations									
19-B	SCN9A	het	Exon 21	mat	Chr2: g.167089942G>C	NM_002977.3:c.3799C>G	p.(Leu1267Val)	Disease causing	Probably damaging	Deleterious

Abbreviations: dn, *de novo*; het, heterozygous; hom, homozygous; mat, maternal; pat, paternal; NP, not predicted.

Bold entries indicate deleterious mutations. In addition, the associated amino-acid substitutions are located in evolutionarily highly conserved residues and are predicted to functionally affect the encoded protein. Results of segregation analyses by Sanger sequencing are reported too.

[a] Exon numbering refers to the NCBI transcript number reported in[b].

Figure 59.Selected variants after the filtering protocol. In bold are reported those variants that fulfill the criteria for disease-causing mutations: affected genes already associated withpatient's phenotype, exhibit complete segregation with the disease, and are absent in healthy controls

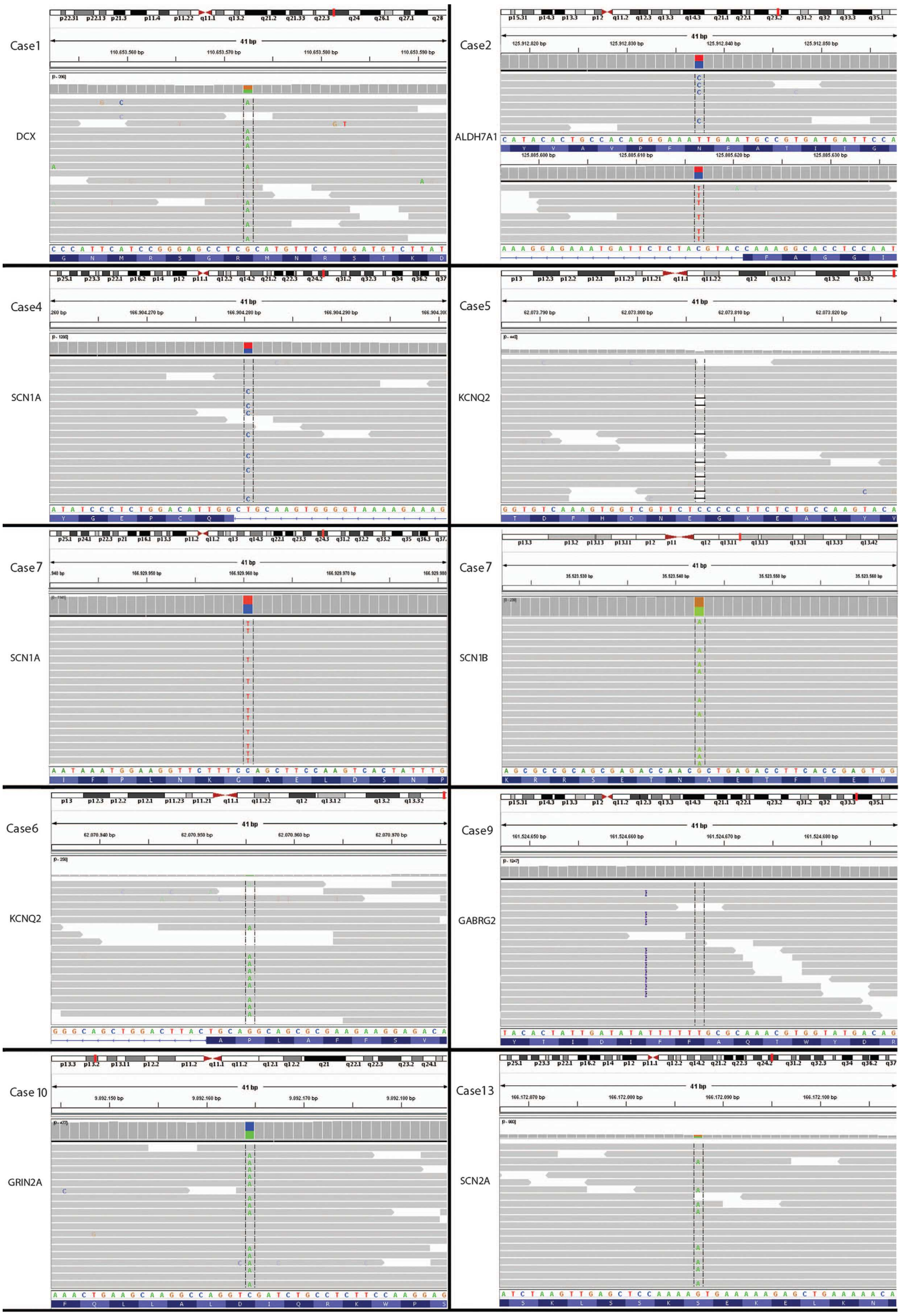


Figure 60. Alignments loaded on IGV 2.1(Integrative Genomics Viewer) for every causative mutation reported in bold in Table 2. Chromosomal view has the covered 41bp delimitated in red, followed by genomic coordinates, relative coverage for single base pair, and alignments covering 101 bp in average. At the bottom of every image the reference sequence and the corresponding amino-acid sequence are visible.

5.2.3.1. Subcohort A

Case 1-A, a female, showed a de novo transition in exon 3 of the DCX gene (NM_000555.3) leading to a nonsense mutation. The reading frame was interrupted by a premature stop codon possibly leading to nonsense-mediated decay (NMD) of the mRNA as suggested by the prediction tools (Polyphen2, Mutation Taster, and SIFT). Leger et al has reported the same mutation[234]. A paternally inherited SHANK3 mutation, considered probably damaging by Polyphen2, was also present. Case 2-A showed two mutations both in ALDH7A1 gene, one inherited from the father and the other one from the mother. The paternal allele had a transition reported in ESP (MAF 0.0077%) and HGMD (CM087549). The maternal allele had a mutation occurring in intron 16, 5 bp upstream to the previous exon. This mutation was present in ESP (MAF 0.0231%) in heterozygosis and was also reported in HGMD (CS091873). The intronic variant was predicted to alter the splicing by skipping of exon 16. Case 3-A had a maternally inherited missense mutation at GPR98 gene that was predicted as damaging by one prediction tool. Case 4-A, showed a de novo splice site mutation in the SCN1A gene. A heterozygous missense mutation of POLG, was also detected both in the patient and in his normal father. This alteration was given as damaging by prediction tools. Case 5-A showed a 1-bp deletion, in the KCNQ2 gene creating a frameshift with a premature stop 15 codons downstream. The mutation was inherited from the father who suffered from the same type of epilepsy. A missense mutation in MAGI2, predicted as damaging and inherited from her normal mother was also present. Case 6-A showed a de novo missense substitution in the KCNQ2 gene. Other two heterozygous missense mutations were detected in GPR98 and TBC1D24, both inherited from the normal father. Case 7-A had two missense mutations at SCN1A and SCN1B, both predicted as damaging. SCN1A mutation was de novo, whereas SNC1B alteration was inherited from the father.

5.2.3.2. Subcohort B

Case 9-B showed a 1-bp exonic duplication in the GABRG2 gene. This duplication creates a frameshift starting at codon Ala118 with the new reading frame ending in a stop five codons downstream. The mRNA was predicted as target for NMD by MutationTaster. The same duplication was found in the father who suffered from the same condition. A mutation of RELN, predicted as damaging and inherited from her normal mother was also detected. Case 10-B had a heterozygous missense mutation in the GRIN2A gene inherited from his mother who showed an overlapping phenotype. This transversion occurred in a highly conserved nucleotide. A missense mutation in the RELN gene, predicted damaging by MutationTaster, resulted to be inherited from the normal father. Case 13-B showed a maternally inherited missense transition in the SCN2A gene, affecting a highly conserved nucleotide and predicted to be damaging by SIFT and Mutation Taster but not by Polyphen2. Case 16-B had two missense mutations at GPR98 and GRIN2A, both predicted to be damaging by at least one prediction tool. Both GRIN2A and GPR98 mutations were inherited from the normal father. Case 19-B had a predicted damaging missense mutation at SCN9A, inherited from the mother. Cases 8-B, 11-B, 12-B, 14-B, 15-B, 17-B, and 18-B did not show any possible causative mutation.

5.2.3.3. Control Cohort Variants

The examination of the 22 variants found in controls after discrete filtering revealed that only 4 of these, all heterozygous, were potentially disease causing. In particular, we found a missense substitution p.(Gly216Ala) (c.647G4C) in CHRNA4 gene (NM_000744) in one case and a heterozygous missense mutation p.(Gly1602Ser) (c.4804G4A) in FLNA gene (NM_001110556) in a female subject. A third control subject had a mutation of UBE3A (NM_000462) (c.1735G4A, p.(Val579Met)) and a second mutation of SCN1B (NM_199037) (c.178C4T, p.(Arg60Cys)).

5.2.4. Discussion

We used a NGS-based approach to test 67 epilepsy genes in 19 patients with different types of epilepsy. Patients had been stratified in two groups according to their neurological phenotypes. In the group A, including seven patients whose clinical features were rather suggestive for a specific syndrome, we detected a likely causative mutation in six (cases 1-A, 2-A, 4-A, 5-A, 6-A, 7-A). In the group B, including 12 patients with a phenotype not distinctive for any specific gene, we have been able to find a plausible causative mutations only in three (cases 9-B, 10-B, 13-B), whereas the remaining cases were either negative (seven cases) or had mutations whose role was unclear (cases 16-B and 19-B). These results were not unexpected and emphasize the restriction of this approach is a lack of knowledge about the functional role of most variants, resulting in a large number of variants of uncertain significance. For this reason, the diagnostic yield of 47% (9/19) is quite high. The 12 cases who had at least one mutation are discussed below. Patient 1-A had a typical clinical picture of Lennox–Gastaut syndrome and magnetic resonance imaging showed a very large double cortex overlapping with the subcortical band heterotopias syndrome. The de novo truncating mutation of DCX fits very well with her double cortex. The mutation of SHANK3 was inherited from her normal father. SHANK3 alterations are causative of Phelan- McDermid syndrome [235] and are characterized by complete penetrance. Thus, we assumed that this missense mutation was likely benign. Patient 2-A showed neonatal seizures and multifocal epileptiform discharges at EEG, which became normalized with pyridoxine. Presently the patient is 14 years old and the treatment with pyridoxine allowed a complete control of seizures with a normal psychomotor development. She had a compound heterozygous mutation for ALDH7A1. The intronic mutation would have been lost if we had disregarded all SNPs reported in public databases, whereas we have taken into consideration all the SNPs with a MAF<1%. Actually, both mutations have already been described in patients with pyridoxine-dependent epilepsy [236]. The finding that her epileptic crisis ceased after pyridoxine treatment indeed demonstrated the causality of the ALDH7A1 mutations. Patient 3-A, who was also affected by pyridoxine-dependent epilepsy, had normal IQ No mutations were detected in the candidate ALDH7A1 gene, whereas a missense mutation was present in GPR98 inherited from the normal mother. Alterations of GPR98 have been associated with familial febrile seizures and autosomal recessive or digenic dominant Usher syndrome. However, the clinical phenotype of the patient was completely different from these conditions. Our findings suggest that pyridoxine-dependent seizure does not only rely on ALDH7A1 mutations. Patient 4-A had a clinical diagnosis of Dravet syndrome, so the de novo splice site mutation of SCN1A fitted well with his phenotype. The heterozygous mutation in POLG was considered not associated with his condition because it was also present in the healthy father, whereas dominant POLG mutations are associated with adult onset progressive external ophthalmoplegia [237]. Patient 5-A had neonatal generalized tonic-clonic seizures, occurring on the second day of life, not responsive to any therapy. Seizures ceased spontaneously at the 25th day of life. Her father, who was found to carry the same mutation, had the same neonatal condition also ending at the 25th day of life. The frameshift mutation at KCNQ2 well correlated with the phenotype [238]. The mutation in MAGI2, inherited from the healthy mother, was not considered disease causing, although it was predicted as probably damaging. Haploinsufficiency for MAGI2 has been associated with hypsarrhythmia [239], a condition different from the one we observed in this family. Patient 6-A showed neonatal seizures since his 3rd day of life that ceased 1 month later. He then suffered from sporadic seizures episodes persisting until now (8 years old). He also presented a severe pervasive developmental disorder. The family history was negative. He had a de novo missense mutation of KCNQ2. This type of mutation has been reported in several patients with early onset epileptic encephalopathy[240] in contrast to frameshift or nonsense mutations that are more frequently found among patients with BFNS. Patient 7-A was affected by generalized epilepsy with febrile seizures plus (GEFSP) and had normal IQ. Our investigations revealed a de novo missense mutation of SCN1A and a missense mutation of SCN1B inherited by the father. The two mutations could explain her phenotype as GEFSP is extremely heterogeneous, and both SCN1A and SCN1B are among the genes associated with this condition[241]. Patient 9-B, during her first year of life suffered from numerous febrile seizures that diminished later on. She had a frameshift mutation at GABRG2 present also in her father who suffered from febrile seizure until 5 years of age. In fact, mutations of GABRG2, either missense or truncating, cause a spectrum of seizure disorders, ranging from early-onset isolated febrile seizures to GEFSP, type 3, which represents the most severe phenotype[242]. The type of febrile seizures present in this patient and in her father fits well with the milder phenotype. The heterozygous mutation of RELN has not been considered as causal of her phenotype because only recessive mutations of this gene are associated with a pathogenic condition characterized by lissencephaly not present in our patient who has a normal psychomotor development. Patient 10-B showed a typical clinical and EEG’s picture of benign childhood epilepsy with centrotemporal spikes. We detected a missense mutation at GRIN2A that was also present in his mother and aunt (the sister of the mother) showing the same clinical and EEG’s picture with remission of seizures in adolescence and borderline cognitive level. Our patient did not have overt seizures until the age of 7 years when rolandic epilepsy appeared. A series of mutations of this gene have been described in subjects/families with a phenotype overlapping that of our patient including learning disabilities. Patient 13-B had a missense mutation of SCN2A inherited from her mother. Mutations of this gene are associated with noteworthy clinical variability ranging from familial benign seizures to generalized epilepsy with febrile seizures or epileptic encephalopathy. The patient presented only febrile seizures, whereas her mother had benign generalized epilepsy with absences. Patient 16-B had an epileptic encephalopathy with severe cognitive impairment. The two missense mutations highlighted in GPR98 and GRIN2A did not seem related to his phenotype. As he was born from healthy consanguineous parents, an autosomal recessive condition has to be taken into consideration. Finally, patient 19-B had a missense mutation of SCN9A predicted as damaging. However, mutations of this gene are usually associated with febrile seizure, GEFSP, and Dravet syndrome, whereas the patient’s phenotype was suggestive of an epileptic encephalopathy strongly resembling West syndrome with hypsarrhythmia, spasm, and psychomotor regression. Both his parents were healthy.

Our targeted platform was thought with the aim to provide a quick and cheap molecular diagnosis to most patients with an epileptic disorder. When we built the platform, we thought we could identify the causative mutation independently from any clinical information. In this sense we required the specialist to select the cases in a totally random way and without giving us any information about their phenotype and family history. The only request was to exclude cases with holoprosencephaly for which we have a dedicated NGS platform. Actually, the finding of multiple candidate mutations made clear that the culprit gene could be highlighted only by knowing in detail both the patient's phenotype and the family history. Moreover, predicted damaging mutations had been detected in the healthy controls as well. Eventually, in 9 of 19 patients we identified a very likely causative mutation (Figure 59) with most of them detected in cohort A including patients whose phenotype indeed suggested the involvement of a specific gene. Among the 12 patients owing to cohort B, affected by different types of epilepsy not suggestive for any or a single specific gene, we could find the most likely causative mutation only in three and in all of them (cases 9-B, 10-B, 13-B) the alteration could be hardly suspected a priori.

The absence of any mutation in seven patients (cases 8-B, 11-B, 12-B, 14-B, 15-B, 17-B, 18-B) indicated that alterations in many other genes not present in our platform are associated with epilepsy, stressing the high genetic heterogeneity of this disorder.

The analysis of the control cohort revealed four potentially damaging mutations in three healthy individuals. None of these variants were previously reported in HGMD. One female subject had a FLNA mutation predicted to be damaging. We could not define whether this was a benign variant rather than a really damaging mutation with incomplete penetrance as reported for females with mutation of this gene and cardiac valvular dysplasia (OMIM #314400). The interpretation of the CHRNA4 mutation was also difficult as alterations of this gene can cause either nocturnal frontal lobe epilepsy type 1, although with incomplete penetrance, or nightmares and other sleep disorders that are often undiagnosed. Variants in SCN1B and UBE3A were identified in the same subject. As in the case of FLNA, the SCN1B mutation could be either neutral or pathological with incomplete penetrance, whereas the UBE3A mutation could be either a benign variant or disease causing but inherited from the father.

To conclude, we were unable to interpret some of the genetic lesions we met in the control cohort, further stressing that our knowledge of genetic variants is presently limited, increasing the risk of false-positive and false-negative information. If these lesions were indeed pathogenic, we should consider the hypothesis that common disease traits such as epilepsy are the result of different genetic components. In fact, the observation of deleterious mutations in 237 ion-channel genes with the same prevalence in individuals with epilepsy and control subjects suggested that, at least for these genes, the personal risk assessment in epilepsy depends more on the combination of the variants rather than specific deleterious variants[243]. The lack of enrichment of protein-disrupting ion-channel mutations in individuals with epilepsy has been confirmed by Heinzen et al[244]. These authors also demonstrated that single epilepsy-susceptibile variants identified by exome sequencing in patients with idiopathic generalized epilepsy (juvenile myoclonic epilepsy and absence epilepsy), although rare, were possibly real risk factors, each of them accounting for only a small fraction of individuals with epilepsy. This burden of data makes evident the complex architecture of epilepsy with genetic heterogeneity much higher than expected. It is conceivable that in the near future the collection of clinical history, EEG, and imaging will be combined with the analysis of NGS-dedicated platforms. The negative cases will be analyzed for whole exome if not for whole genome. The advantages of this approach are evident both in the immediate (consulting for risk of recurrence) and in the long run when specifically targeted treatments will be adopted.

We have been able to conclude the analysis of nine patients (the number of patients we pool in a single lane), including the enlargement of the investigation to parents, in 8–10 weeks with a cost per patient comparable to sequencing 1–2 medium-to-large-sized genes by conventional techniques, a result overlapping that reported by Lemke et al[35]. Our results suggest that using a single platform to sequence all or most of the epilepsy genes may increase the diagnostic yield. This is particularly true in absence of clinical signs suggestive for involvement of a specific gene like in three patients of our cohort. Obviously novel mutations require that their causative role is further confirmed by segregation or functional analyses. On the contrary, smaller platforms containing a limited number of genes may reduce the efficacy of the NGS-based approach, as epilepsy is extremely heterogeneous both under the genetic and phenotypic point of view. Our platform, as well as the one already described by Lemke et al, has the advantage that can also be run on ‘benchtop’ sequencers, which forego high yields in exchange for low capital costs, small physical footprints, and more rapid turnaround times, making them far more attractive to smaller biomedical laboratories.

5.3. Lower motor neuron disease with respiratory failure caused by a novel MAPT mutation

Lower motor neuron diseases (LMNDs) are a group of heterogeneous clinical presentations accounting for approximately 10% of all motor neuron disorders. Various names have been used for different forms of LMND, such as progressive muscular atrophy (PMA), distal muscular atrophy, and segmental distal/proximal spinal muscular atrophy. PMA is diagnosed clinically by the presence of progressive generalized muscular involvement and proven anterior horn cell degeneration at postmortem examination. PMA differs from amyotrophic lateral sclerosis (ALS), in which both upper and lower motor neurons are clinically involved. Progression may be rapid or display a gradual deterioration, with PMA showing a longer survival compared with ALS. The underlying genetic defects are heterogeneous [245]. Mutations in the microtubule (MT)-associated protein tau (MAPT) gene have been reported in neurodegenerative disorders with abnormal tau protein accumulation, such as frontotemporal dementia with parkinsonism linked to chromosome 17 (FTDP-17), progressive supranuclear palsy, corticobasal degeneration, and late onset Parkinson disease dementia [246]. MAPT involvement in the etiology of motor neuron degeneration was derived from the observation of tau pathology in subjects with ALS/Parkinson-dementia complex from Guam, New Guinea, and the Kii peninsula of Japan[247]. Nevertheless, MAPT mutations have not yet been linked to a pure motor neuron disease phenotype.

Hereby we describe the identification of a novel MAPT mutation underlying adult-onset autosomal dominant LMND with prominent respiratory insufficiency, proximal weakness of the upper limbs, and no signs of frontotemporal lobar degeneration or semantic dementia in a large Italian family.

5.3.1. Methods

5.3.1.1. Linkage Analysis

DNA samples were obtained from 10 subjects (IV_17, 18, 19, 20,21, 22, 23, 24, 25,26, and 29). Genome-wide genotyping of samples IV_17, 18, 20,21,22, and 26 was performed using Affimetrix GeneCHip Human Mapping 250K Array. Nonparametric linkage and parametric analyses were performes using ALLEGRO software [248]. Twenty-five short tandem repeats (STR) markers were also genotyped by PCR.Haplotypes were reconstructed by ALLEGRO and displayed using HaploPainter (http://haplopainter.sourceforge.net).

5.3.1.2. Exome capture and sequencing

A total of 5 µgof genomic DNA from 3 affected (IV_17, 18, and 21) and one unaffected (IV_23) family members underwent exome analysis using the SureSelect Human All Exon Kit (Agilent Santa Clara, CA). Samples were processed according to the Illumina protocol, following manufacturer's instructions. Sequencing was performed on the Illumina Genome Analyzer IIx platform as paired-end 100-base pair reads. An exome capture was considered successful if >80% of the target regions covered with a high-quality genotype.

5.3.1.3. Bioinformatics analysis of exome data

Reads were aligned to the GRCh37/hg19 genomic reference using BWA (v.0.5.9). Mapped reads were filtered for PCR duplicates using SAMtools (v.0.1.18), locally realigned around inferred insertions and deletions and their base qualities recalibrated in the context alignment using GATK (v. 1.4-21). Variant calls were obtained by GATK (Unified Genotyper) as well.

5.3.2. Results and Clinical findings

Five of the family members (IV_20, 21, 17, 18, and 26) presented common sympthoms: progressive proximal weakness (mainly affecting upper limbs) with weak tendon jerks, no bulbar or pyramid signs, early development of restrictive respiratory insufficiency with the need of mechanical ventilation, and no dementia.

Patient IV_20 was 55 years old when she presented with lumbar backhache and difficulties staying upright. Progressive weackness of the proximal upper limbs and dyspnea with restrictive lung disease appeared a few months later. The patient needed noninvasive ventilation at age of 67, and from the age of 71, she could not walk and required a wheelchair. The patient started therapy with 550 mg riluzole twice daily, in the hypothesis of a motor neuron disease. At the age of 70, she developed mild anxiety, partly due to the respiratory distress and the use of noninvasive ventilation. She had no frontal release signs. The patient died of respiratory failure at the age of 72.

Patient IV_21 presented at 65 years of age with lumbar backache. One year later, she developed dyspnea and progressive restrictive respiratory insufficiency, which required noninvasive ventilation. She died at 72 years of age of respiratory failure and underwent autopsy.

Five cousins (subjects IV_17, 18, 26, 27, and 28) had similar symptoms such as lumbar backache, leg cramps, fatigue, mild dyspnea, and proximal upper and lower limb weakness. None of them had psychiatric symptoms. Neuropsychologic examination in patients IV_17 and IV_18 was normal.

Two other cousins (subject IV_27 and IV_28) and uncle (subject III_10) of the probands died after short history of fatigue and dyspnea. Their clinical records have been collected. Patients IV_27 and IV_28 presented with bradykinesia and camptocormia and were initially diagnosed elsewhere with Parkinson disease, which was intriguing because of the known association of parkinsonism with motor neuron disease as part of the FTDP-17 and Guam syndromes. However, there was no mention of levodopa response, and whether the anterior flexion of the trunk without resting tremor and muscular rigidity was due to Parkinsonism or to axial muscle weakness remains unclear.

5.3.2.1. Genetic studies

Using a core pedigree (IV_17, 18, 20, 21, 22, and 26) a genome-wide linkage analysis revealed a nonparametric linkage LOS score of 1.88 and parametric LOD score of 1.78 on chromosome 17q21 (Figure 61B). STR haplotype analysis confirmed this finding (Figure 61C) and defined an 8-Mb candidate region.

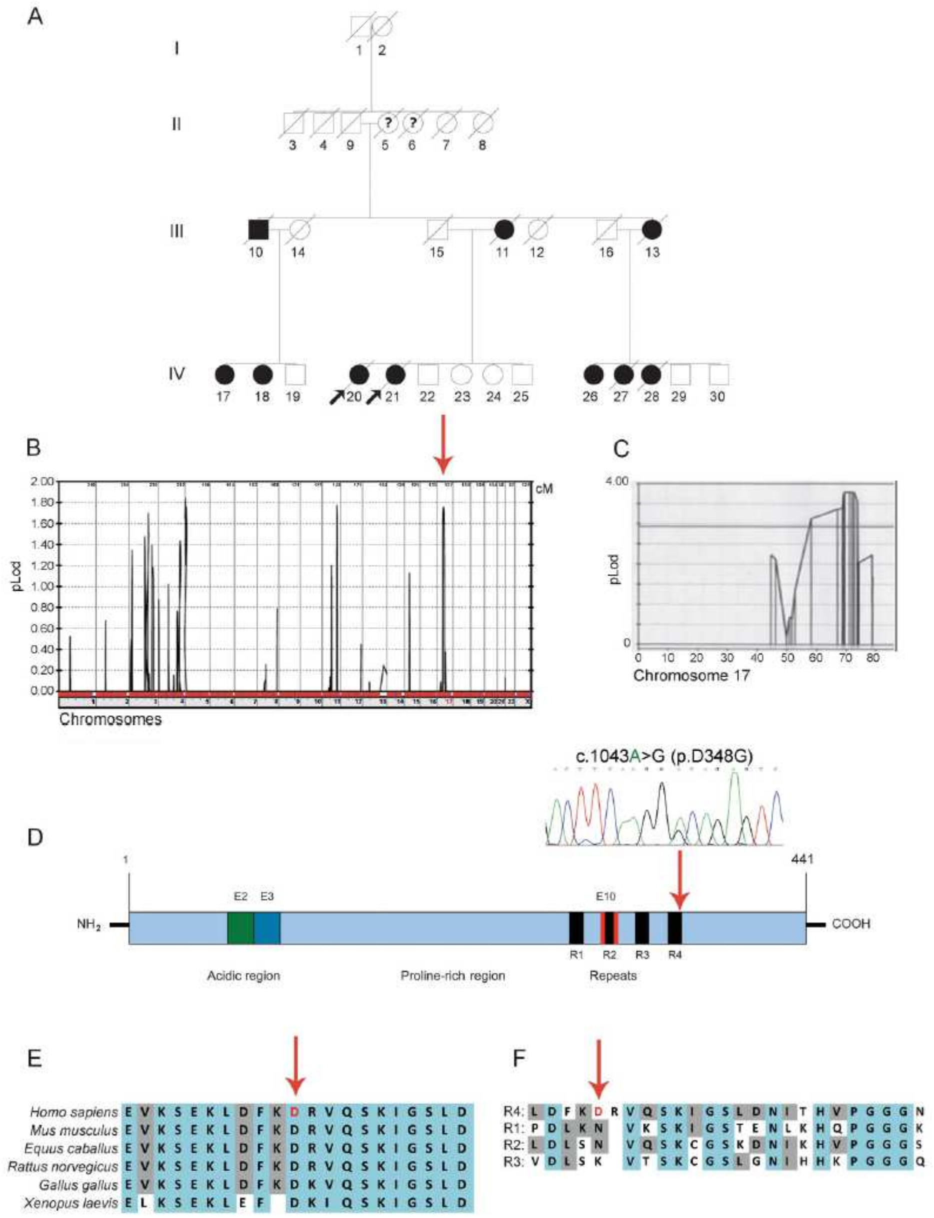

A
I
II
III
IV
1 2
3 4 9 5 6 7 8
10 14 15 11 12 16 13
17 18 19 20 21 22 23 24 25 26 27 28 29 30
B
pLod
cM
Chromosomes
C
pLod
4.00
0
0 10 20 30 40 50 60 70 80
Chromosome 17
c.1043A>G (p.D348G)
D
1
441
NH2
COOH
E2 E3
E10
R1 R2 R3 R4
Acidic region
Proline-rich region
Repeats
E
Homo sapiens E V K S E K L D F K D R V Q S K I G S L D
Mus musculus E V K S E K L D F K D R V Q S K I G S L D
Equus caballus E V K S E K L D F K D R V Q S K I G S L D
Rattus norvegicus E V K S E K L D F K D R V Q S K I G S L D
Gallus gallus E V K S E K L D F K D K V Q S K I G S L D
Xenopus laevis E L K S E K L E F D K I Q S K I G S L D
F
R4: L D F K D R V Q S K I G S L D N I T H V P G G G N
R1: P D L K N V K S K I G S T E N L K H Q P G G G K
R2: L D L S N V Q S K C G S K D N I K H V P G G G S
R3: V D L S K V T S K C G S L G N I H H K P G G G Q

Figure 61. A. Pedigree of the family. Arrows indicate probands IV_20 and IV_21. (B) Multipoint linkage analysis of the genome-wide scan. The peak on chromosome 17 is indicated by the arrow. (C) Parametric multipoint linkage analysis with short tandem repeats on chromosome 17. (D) Scheme of the human tau protein encoded by MAPT. The electropherogram of the c.1043A>G mutation in exon 12 resulting into the p.D348G amino acid substitution. (E) ClustalW multiple sequence alignment of the tau region containing the mutated residue in the family.

Whole-exome analysis was performed on affected subjects IV_17, IV_18, and IV_21 and one unaffected (IV_23) member of the family. We focused on heterozygous variants in exons, in and near (20 bases from exons) splice site junctions, in 5’ and 3’ untranslated regions, and that segregated with the disease among the 4 sequenced individuals. After filtering out variants found in uncorrelated samples from an internal exome-sequencing database and the ESP database, only 2 variants in 2 genes remained, both single-nucleotide substitutions affecting sperm associated antigen 5 (SPAG5) and MAPT, which are found within the 8-Mb linked region on chromosome 17. The MAPT variant (NM_005919.5:c.1043A>G) involves exon 12, resulting in the p.D348G change at protein level (NP_005901.2). Sanger sequencing of MAPT in affected patients (IV_17, 18, 20, 21, 22, and 26) confirmed the same heterozygous mutation (Figure 61D) and its autosomal dominant pattern of inheritance.

The transition was not detected in more than 500 controls. The affected residue, located in the fourth repeat domain (R4), is highly conserved among higher eukaryotes (Figure 61E) but not among other repeat domain in the human tau sequence (Figure 61F).

5.3.2.2. Transcript analysis

The expression of MAPT exons 2,3 4A and 10 is temporally and spatially regulated. In particular, the inclusion of exon 10 leads to the production of isoforms containing 4 MT-binding repeats (4R-tau), whereas its exclusion leads to the synthesis of 3.repeat isoforms (3R-tau). Similar levels of 3R and 4R isoforms are detected physiologically in normal adult human cerebral cortex. However, this ratio is often altered in the cortex patients with MAPT mutation. Therefore, the possible effects of the c.1043A>G MAPT mutation at the transcript level has been checked. Total MAPT expression has been assessed by quantitative reverse transcription-PCR analysis in postmortem samples from patient IV_21 including cervical spinal cord. There was no a significant difference between patient and control MAPT RNA levels, with the exception of the frontal lobe of patient IV_21, which had less MAPT transcript (ratio 0.58 ± 0.09). The levels of expression of 4R:3R isoforms were estimated as well, by using specific probes designed to detect the 2 distinct isoforms. The data were normalized to housekeeping gene 18S and total MAPT. We failed to detect any difference between samples from IV_20 and controls. Conventional reverse transcription-PCR analysis confirmed these findings.

Taken together, these findings argue against a transcriptional effect of the c.1043A>G mutation on expression of MAPT transcript, as previously observed for other mutations in exon 12[249]. Moreover, no quantitative or quantitative alteration in MAPT transcript was observed in the spinal cord, making it an unlikely cause of the selective involvement of the spinal cord observed in our patients.

5.3.2.3. Immunocytochemical analysis in NSC34 cell models

One of the physiologic roles of tau within the cell is thought to be the stabilization of microtubules (MTs). Therefore, has been explored whether the mutation p.D348G mutation impairs tau stability and affects its function on MT assembly and organization. A differentiation protocol was used in NSC-34 motoneuron-like cells to achieve a motoneuron phenotype and to promote neurite elongation. The established cell models underwent confocal immunocytochemical analysis using antibodies directed against tau and acetylated tubulin (Figure 62).

In some cells, MT bundles were observed after tau transfection and this phenomenon was more evident for mutated tau protein than for wild-type moiety. Cells overexpressing mutated tau isoforms displayed a consistent reduction in neurite length and arborization. A disturbance in MT stability and organization was also confirmed when evaluating the acetylated tubulin signal, which was decreased in transfected cells expressing mutated tau (Figure 62A). Confocal analysis showed that mutated tau caused a reorganization of the MTs and the creation of thick MT bundles, which appeared as swirls around the nucleus. These cells consistently displayed a poor MT network with few, short axons compared with nontransfected cells or cells expressing wild-type tau (Figure 62B).

Despite these findings, the colocalization of tau and acetylated tubulin was preserved, suggesting that the p.D348G mutation does not impair the binding of tau to MTs. This concept has been further confirmed by biochemical studies (Western blot analysis).

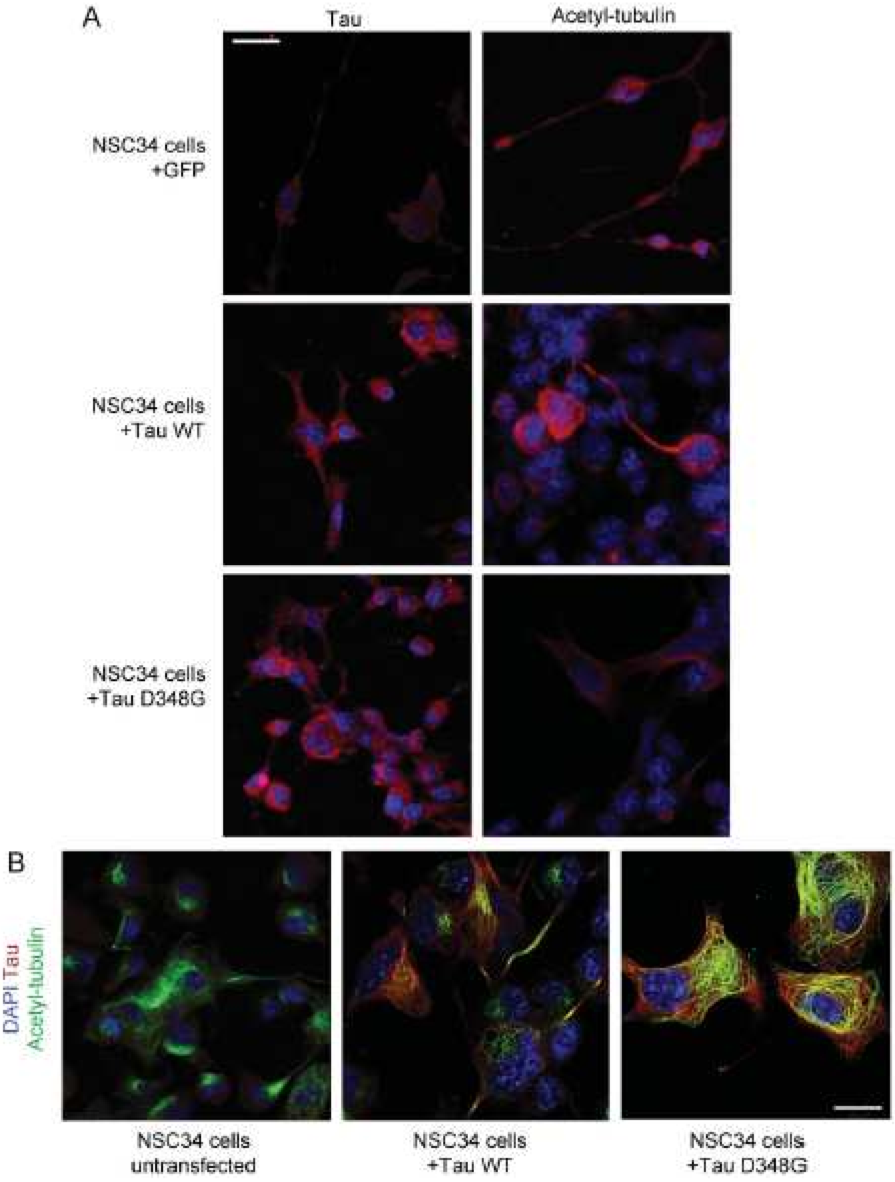


Figure 62. Immunocytochemical studies. (A) NSC34 cells were stably transfected with vector overexpressing complementary DNA containing wild-type (WT) and mutated (D348G) tau open reading frames. Confocal microscope immunocytochemical analysis was performed using antibodies directed against human tau (5A6). Microtubule networks were detected by immunolabeling with antibodies against acetylated tubulin. No difference was observed in the subcellular localization of tau protein. Impaired axonal elongation was more evident in the presence of mutated tau protein. (B) Double-immunofluorescence staining of tau transfectants. Cells were labeled with antibodies to tau (red) and acetylated tubulin (green). Cells overexpressing D348G tau exhibited short neurites, but colocalization of tau and acetylated tubulin signals (orange and yellow color) was observed for both WT and mutant transfectans, suggesting that the D348G mutation does not likely impair tau binding to microtubules.

5.3.3. Discussion

MAPT mutations have been described in different neurodegenerative diseases, including FTDP-17, progressive supraneuclear palsy, and corticobasal degeneration. In some of the cases described so far, MAPT mutations generated neurodegenerative disease phenotypes in which the spinal cord was clinically and/or neuropathologically involved, as in the case of the disinhibition-dementia-parkinsonism-amyotrophy complex [250].

Degeneration of spinal motor neurons with tau inclusions but without amyotrophy has been described in patients with FTDP-17 affected by the MAPT p.N279K mutation [251]. Similarly, a MAPT p.L317M mutation in exon 11 has been found to cause a phenotype characterized by dysarthria, tremor, amyotrophy, and frontal signs. Some of the patients developed levodopa-resistant Parkinsonism and supranuclear palsy, whereas others displayed corticobasal degeneration. Notably, histologic examination of the cervical spinal cords from these patients revealed degeneration of the spinal motor neurons and tau inclusions.

The suggestion that tauopathy may be involved in motor neuron pathology was supported by transgenic models. Mice expressing human P301L tau exhibit a motor phenotype with evidence of motor neuron loss, and transgenic mice expressing human tau in both neurons and glia also displayed axonal degeneration at the spinal cord level [252].

The family we examined clearly proves the link between MAPT mutation and tau pathology in motor neuron pathology. The phenotype associated with the novel p.D348G mutation described here resembles that of PMA, including the involvement of the lower motor neuron of the proximal limbs and trunk, subsequent respiratory failure, and absence of dementia and pyramidal and bulbar signs. A low degree of clinical variability was observed among affected family members.

Patient IV_20 presented a slow disease course (12 years), whereas the course was slightly faster (5-7 years) in the case of his sister and cousins (IV_21, IV_27, and IV_28).

None of these patients developed cognitive deficits, not even patient IV_20, who had the longest disease course. This point represents a novelty in the framework phenotype associated with mutations in MAPT, in which the frontotemporal deficits have so far been reported as hallmarks.

Only one patient (III_10) would have developed delirium at the last disease stage. Unfortunately, whether this suggests phenotypic heterogeneity or was due to unknown comorbidities remains to be elucidated because detailed clinical and instrumental data were not available.

All affected patients developed respiratory insufficiency, early in the disease course in some cases. Some other cases have reported a similar involvement, despite the age at disease onset being different.

A homozygous APT S352L mutation in exon 12 has been reported to cause autosomal recessive restrictive respiratory failure during youth in 2 siblings[253]. However, we can exclude that this phenotype is due to mutations within a specific tau protein domain, because other mutations affecting surrounding residues often result in frontotemporal lobar degeneration without spinal cord involvement.

Most of the described MAPT mutations, especially those in intron 10, pN279K, and p.S305N, have been shown to induce a preponderance of tau protein isoforms 4 MT-binding repeats, which alters kinase anterograde transport because tau competes with the kinases at the MT-binding site. Other exonic mutations located within the MT-binding repeats decrease tau binding to MTs, causing destabilization and resulting in faulty axonal transport [254]. The p.D348G mutation does not alter the expression of the 3R and 4R isoforms or the ability of tau to bind MTs, which suggests that the mutation causes degeneration via different mechanism. We speculate that mutated tau escapes natural proteasome degradation and consequently accumulates in neurons, leading to neurotoxicity. The neuropathologic findings in our patient seem to support this hypothesis.

The discovery of a new MAPT mutation causing autosomal dominant motor neuron disease associated with tau pathology represents a new finding for the etiology and pathogenesis of these neurodegenerative diseases and offers new possible diagnostic and therapeutic approaches for a category of presently incurable diseases. Moreover, the reported mutation broadens the spectrum of phenotypes associated with MAPT alterations, suggesting that patients with autosomal dominant LMDs with respiratory involvement should also be screened for MAPT mutations.

Chapter 6

Conclusions and Future Works

The aim of the work described in this thesis consisted in the design and in the development of innovative strategies and technologies dedicated to manage and interpret the overwhelming amount of data produced by high throughput sequencing platforms, particularly focusing on genomic variants.

It has been discussed how new generation sequencing is revolutionizing the Genomic research empowering clinical diagnostics and other aspects of medical care. Nonetheless, we highlighted how these technologies would be useless if not supported by Bioinformatics, dedicated to analyze sequencing data, manage and support genomic data interpretation.

With these goals in mind, we have developed a system written in Java and based on a MySQL database to annotate, store and extract genomic variants coming from targeted enrichment sequencing experiments.

This system allowed us to manage hundreds of sequenced samples and millions of related variants. By using its web client interface, we extracted subsets of interesting annotated genomic variants ready to be further analyzed for each sequencing project. One of the most important features of the developed system has been its capacity to perform case-control studies by reporting cases genomic variants and matched controls data aggregates on the fly.

We have discussed how the limitations of the system in terms of computational performances and flexibility jointly with the introduction of new developed technologies leaded us to change our strategy and bet on a new approach to the problem.

Therefore a new system to manage both genotype and phenotype of the sequenced individuals has been developed. Based on CouchDB, a NoSQL database, we showed how it allows annotating, storing and retrieving millions of genomic variants and significantly reduced computational times if compared to the previous system. Moreover, we showed how the integration into the i2b2 framework allows first to select the patient cohort of interest by the available phenotypic data and then to retrieve related genomic variants, creating the bases for phenotype-genotype correlation studies.

Future directions comprehend the extension of the developed NoSQL system in order to set up the update procedures that guarantee the continuous state-of the art of the genomic variant annotations. The system flexibility makes this effort more straightforward respect to a relational database application. Another point would be the enhancing of the developed i2b2 plug-in and cell, allowing for knowledge-base queries, such as pathways, protein-protein interactions, list of known-disease genes or by implementing those discussed ontologies-based search algorithms for gene prioritization. The latter point, giving the inbound phenotype data from i2b2, should be straightforward at least for phenotype-ontologies based gene prioritization tools such as Phevor [112] or Extasy [118].

We also dealt with one of the most important problem in genomic variant interpretation, which is the in-silico prediction of genomic variant pathogenicity inferred by their probability to alter protein stability and/or function.

We discussed how the outcome of existing variant prediction tools is one of the massively used parameters to discern, among the plethora of sequencing variants, potential disease-related variants from neutral ones.

We therefore developed PaPI, a new algorithm that applies machine learning algorithms with features derived from differences into the pseudo amino acid composition (PseAAC) of known disease and neutral protein variants. The algorithm is able to classify unseen genomic variants into damaging or tolerated class within a confidence score. We demonstrated that PaPI results in higher accuracy with respect to the state-of-art variant prediction tools, and in higher coverage, encompassing in fact every genomic variant and type. Moreover it is suitable to assert the true variants pathogenicity of those variants altering amino acid patterns such as in binding and methylation sites. We also developed a free web accessible application (http://papi.unipv.it) able to predict thousands of variants in runtime.

We argue that PaPI could be further improved and/or specialized by e.g. tuning some parameters we omitted in development phase such as the amino acid snippet length where PseAAC is computed. Also, the inclusion of other amino acid properties such as charge or side chain mass into the PseAAC should be tested. Being PaPI a machine learning method, one can also thing to specialize it to predict mutation within a particular gene or selected genes, assuming to have a consistent number of known observations. This should lead to an improved accuracy for the particular case respect to the general model we made available for the community.

We finally reported some successful clinical applications in which sequencing data analysis procedures joined with the developed relational-based variant management system allowed the geneticists to investigate the genetic causes of heterogeneous diseases such as dehydrated hereditary stomatocytosis, epilepsy and lower motor neuron disease.

However, as highlighted in these works, the solely identification of possible disease-related candidate genomic variants is not enough to assess variant pathogenicity and should be complemented by a rich phenotype patient data collection and experimental evidences that may varies case by case.

We argue that the new developed variant management system integrated within the i2b2 framework will contribute to the first point, while PaPI can be used to better filter the list of possible disease-related genomic variants, but not avoiding the need of experimental evidence, especially in case of variants unreported in the scientific literature [255].

References

1. Dudley, J.T., et al., *Human genomic disease variants: a neutral evolutionary explanation*, in *Genome Res*2012. p. 1383-94.
2. Adzhubei, I.A., et al., *A method and server for predicting damaging missense mutations.* Nat Methods, 2010. **7**(4): p. 248-9.
3. Schwarz, J.M., et al., *MutationTaster evaluates disease-causing potential of sequence alterations.* Nat Methods, 2010. **7**(8): p. 575-6.
4. Murphy, S., et al., *Instrumenting the health care enterprise for discovery research in the genomic era.* Genome Res, 2009. **19**(9): p. 1675-81.
5. Genomes Project, C., et al., *An integrated map of genetic variation from 1,092 human genomes.* Nature, 2012. **491**(7422): p. 56-65.
6. Chou, K.C., *Prediction of protein cellular attributes using pseudo-amino acid composition.* Proteins, 2001. **43**(3): p. 246-55.
7. Breiman, L., *Random Forests.* Machine Learning, 2001(45): p. 27.
8. Kumar, P., S. Henikoff, and P.C. Ng, *Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm.* Nat Protoc, 2009. **4**(7): p. 1073-81.
9. Hall, M., et al., *The WEKA data mining software: an update.* SIGKDD Explor. Newsl., 2009. **11**(1): p. 10-18.
10. Ding, L., et al., *Expanding the computational toolbox for mining cancer genomes.* Nat Rev Genet, 2014. **15**(8): p. 556-70.
11. Rau, J.L., *Design principles of liquid nebulization devices currently in use.* Respir Care, 2002. **47**(11): p. 1257-75; discussion 1275-8.
12. Bartlett, J.M. and D. Stirling, *A short history of the polymerase chain reaction.* Methods Mol Biol, 2003. **226**: p. 3-6.
13. Ewing, B. and P. Green, *Base-calling of automated sequencer traces using phred. II. Error probabilities.* Genome Res, 1998. **8**(3): p. 186-94.
14. Shendure, J. and E. Lieberman Aiden, *The expanding scope of DNA sequencing.* Nat Biotechnol, 2012. **30**(11): p. 1084-94.
15. Bock, C., *Analysing and interpreting DNA methylation data.* Nat Rev Genet, 2012. **13**(10): p. 705-19.
16. Park, P.J., *ChIP-seq: advantages and challenges of a maturing technology.* Nat Rev Genet, 2009. **10**(10): p. 669-80.
17. Wang, Z., M. Gerstein, and M. Snyder, *RNA-Seq: a revolutionary tool for transcriptomics.* Nat Rev Genet, 2009. **10**(1): p. 57-63.

18. Motameny, S., et al., *Next Generation Sequencing of miRNAs - Strategies, Resources and Methods.* Genes (Basel), 2010. **1**(1): p. 70-84.
19. Mertes, F., et al., *Targeted enrichment of genomic DNA regions for next-generation sequencing.* Brief Funct Genomics, 2011. **10**(6): p. 374-86.
20. Mamanova, L., et al., *Target-enrichment strategies for next-generation sequencing.* Nat Methods, 2010. **7**(2): p. 111-8.
21. Sikkema-Raddatz, B., et al., *Targeted next-generation sequencing can replace Sanger sequencing in clinical diagnostics.* Hum Mutat, 2013. **34**(7): p. 1035-42.
22. Ng, S.B., et al., *Targeted capture and massively parallel sequencing of 12 human exomes.* Nature, 2009. **461**(7261): p. 272-6.
23. Ng, S.B., et al., *Exome sequencing identifies the cause of a mendelian disorder.* Nat Genet, 2010. **42**(1): p. 30-5.
24. Yang, Y., et al., *Clinical whole-exome sequencing for the diagnosis of mendelian disorders.* N Engl J Med, 2013. **369**(16): p. 1502-11.
25. Gilissen, C., et al., *Unlocking Mendelian disease using exome sequencing.* Genome Biol, 2011. **12**(9): p. 228.
26. Ng, S.B., et al., *Exome sequencing identifies MLL2 mutations as a cause of Kabuki syndrome.* Nat Genet, 2010. **42**(9): p. 790-3.
27. Lohmueller, K.E., et al., *Whole-exome sequencing of 2,000 Danish individuals and the role of rare coding variants in type 2 diabetes.* Am J Hum Genet, 2013. **93**(6): p. 1072-86.
28. Lange, L.A., et al., *Whole-exome sequencing identifies rare and low-frequency coding variants associated with LDL cholesterol.* Am J Hum Genet, 2014. **94**(2): p. 233-45.
29. Reddy, M.V., et al., *Exome sequencing identifies 2 rare variants for low high-density lipoprotein cholesterol in an extended family.* Circ Cardiovasc Genet, 2012. **5**(5): p. 538-46.
30. Lim, W.K., et al., *Exome sequencing identifies highly recurrent MED12 somatic mutations in breast fibroadenoma.* Nat Genet, 2014. **46**(8): p. 877-80.
31. Guo, G., et al., *Whole-genome and whole-exome sequencing of bladder cancer identifies frequent alterations in genes involved in sister chromatid cohesion and segregation.* Nat Genet, 2013. **45**(12): p. 1459-63.
32. Barbieri, C.E., et al., *Exome sequencing identifies recurrent SPOP, FOXA1 and MED12 mutations in prostate cancer.* Nat Genet, 2012. **44**(6): p. 685-9.
33. Rehm, H.L., *Disease-targeted sequencing: a cornerstone in the clinic.* Nat Rev Genet, 2013. **14**(4): p. 295-300.
34. Biswas, A., et al., *Next generation sequencing in cardiomyopathy: towards personalized genomics and medicine.* Mol Biol Rep, 2014. **41**(8): p. 4881-8.
35. Lemke, J.R., et al., *Targeted next generation sequencing as a diagnostic tool in epileptic disorders.* Epilepsia, 2012. **53**(8): p. 1387-98.

36. Haas, J., et al., *Atlas of the clinical genetics of human dilated cardiomyopathy*. Eur Heart J, 2014.
37. Walsh, T., et al., *Mutations in 12 genes for inherited ovarian, fallopian tube, and peritoneal carcinoma identified by massively parallel sequencing*. Proc Natl Acad Sci U S A, 2011. **108**(44): p. 18032-7.
38. Walsh, T., et al., *Detection of inherited mutations for breast and ovarian cancer using genomic capture and massively parallel sequencing*. Proc Natl Acad Sci U S A, 2010. **107**(28): p. 12629-33.
39. Papaemmanuil, E., et al., *Clinical and biological implications of driver mutations in myelodysplastic syndromes*. Blood, 2013. **122**(22): p. 3616-27; quiz 3699.
40. Consortium, G.R. *GRC and Collaborators*. Available from: http://www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/credits.shtml.
41. Langmead, B., et al., *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome*. Genome Biol, 2009. **10**(3): p. R25.
42. Li, H. and R. Durbin, *Fast and accurate long-read alignment with Burrows-Wheeler transform*. Bioinformatics, 2010. **26**(5): p. 589-95.
43. Li, H. and R. Durbin, *Fast and accurate short read alignment with Burrows-Wheeler transform*. Bioinformatics, 2009. **25**(14): p. 1754-60.
44. Hach, F., et al., *mrsFAST: a cache-oblivious algorithm for short-read mapping*. Nat Methods, 2010. **7**(8): p. 576-7.
45. Li, R., et al., *SOAP2: an improved ultrafast tool for short read alignment*. Bioinformatics, 2009. **25**(15): p. 1966-7.
46. Li, H., J. Ruan, and R. Durbin, *Mapping short DNA sequencing reads and calling variants using mapping quality scores*. Genome Res, 2008. **18**(11): p. 1851-8.
47. Li, H., et al., *The Sequence Alignment/Map format and SAMtools*. Bioinformatics, 2009. **25**(16): p. 2078-9.
48. Yu, X., et al., *How do alignment programs perform on sequencing data with varying qualities and from repetitive regions?* BioData Min, 2012. **5**(1): p. 6.
49. DePristo, M.A., et al., *A framework for variation discovery and genotyping using next-generation DNA sequencing data*. Nat Genet, 2011. **43**(5): p. 491-8.
50. Institute, B. *Picard*. Available from: http://picard.sourceforge.net.
51. Aaron, S. *Understanding genetics*. 2013; Available from: http://genetics.thetech.org/ask/ask166.
52. Das, S. and H. Vikalo, *OnlineCall: fast online parameter estimation and base calling for illumina's next-generation sequencing*. Bioinformatics, 2012. **28**(13): p. 1677-83.
53. Liu, Q., et al., *Steps to ensure accuracy in genotype and SNP calling from Illumina sequencing data*. BMC Genomics, 2012. **13 Suppl 8**: p. S8.

54. Koboldt, D.C., et al., *VarScan: variant detection in massively parallel sequencing of individual and pooled samples.* Bioinformatics, 2009. **25**(17): p. 2283-5.
55. Koboldt, D.C., et al., *VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing.* Genome Res, 2012. **22**(3): p. 568-76.
56. Garrison, E. and G. Marth *Haplotype-based variant detection from short-read sequencing*. ArXiv e-prints, 2012. **1207**, 3907.
57. Challis, D., et al., *An integrative variant analysis suite for whole exome next-generation sequencing data.* BMC Bioinformatics, 2012. **13**: p. 8.
58. Marinoni, A., et al., *A kinetic model-based algorithm to classify NGS short reads by their allele origin.* J Biomed Inform, 2014.
59. Cantarel, B.L., et al., *BAYSIC: a Bayesian method for combining sets of genome variants with improved specificity and sensitivity.* BMC Bioinformatics, 2014. **15**: p. 104.
60. Danecek, P., et al., *The variant call format and VCFtools.* Bioinformatics, 2011. **27**(15): p. 2156-8.
61. Gafni, E., et al., *COSMOS: Python library for massively parallel workflows.* Bioinformatics, 2014. **30**(20): p. 2956-8.
62. Niemenmaa, M., et al., *Hadoop-BAM: directly manipulating next generation sequencing data in the cloud.* Bioinformatics, 2012. **28**(6): p. 876-7.
63. McKenna, A., et al., *The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.* Genome Res, 2010. **20**(9): p. 1297-303.
64. Schatz, M.C., *CloudBurst: highly sensitive read mapping with MapReduce.* Bioinformatics, 2009. **25**(11): p. 1363-9.
65. Dean, J. and S. Ghemawat, *MapReduce: simplified data processing on large clusters.* Commun. ACM, 2008. **51**(1): p. 107-113.
66. Genome, E. *DRAGEN bio-IT Processor*. Available from: http://www.edicogenome.com/dragen/.
67. Gullapalli, R., et al., *Next generation sequencing in clinical medicine: Challenges and lessons for pathology and biomedical informatics*. Vol. 3. 2012. 40-40.
68. Wilks, C., et al., *The Cancer Genomics Hub (CGHub): overcoming cancer through the power of torrential data.* Database (Oxford), 2014. **2014**.
69. Mailman, M.D., et al., *The NCBI dbGaP database of genotypes and phenotypes.* Nat Genet, 2007. **39**(10): p. 1181-6.
70. Leinonen, R., et al., *The European Nucleotide Archive.* Nucleic Acids Res, 2011. **39**(Database issue): p. D28-31.
71. Marx, V., *When disease strikes from nowhere.* Nature, 2014. **513**(7518): p. 445-8.
72. Rios, D., et al., *A database and API for variation, dense genotyping and resequencing data.* BMC Bioinformatics, 2010. **11**: p. 238.
73. Ingenuity. *Ingenuity Variant Analysis*. Available from: http://www.ingenuity.com/.

74. Biobase. *Genome Trax*. Available from: www.biobase-international.com/genome-trax.

75. Dolled-Filhart, M.P., et al., *Computational and bioinformatics frameworks for next-generation whole exome and genome sequencing*. ScientificWorldJournal, 2013. **2013**: p. 730210.

76. Cooper, G.M. and J. Shendure, *Needles in stacks of needles: finding disease-causal variants in a wealth of genomic data*. Nat Rev Genet, 2011. **12**(9): p. 628-40.

77. Bellazzi, R., *Big data and biomedical informatics: a challenging opportunity*. Yearb Med Inform, 2014. **9**(1): p. 8-13.

78. Karolchik, D., A.S. Hinrichs, and W.J. Kent, *The UCSC Genome Browser*. Curr Protoc Bioinformatics, 2012. **Chapter 1**: p. Unit1 4.

79. Stalker, J., et al., *The Ensembl Web site: mechanics of a genome browser*. Genome Res, 2004. **14**(5): p. 951-5.

80. Wheeler, D.L., et al., *Database resources of the National Center for Biotechnology Information*. Nucleic Acids Res, 2008. **36**(Database issue): p. D13-21.

81. Fernandez-Suarez, X.M., D.J. Rigden, and M.Y. Galperin, *The 2014 Nucleic Acids Research Database Issue and an updated NAR online Molecular Biology Database Collection*. Nucleic Acids Res, 2014. **42**(Database issue): p. D1-6.

82. Lathe W, W.J., Mangan M, Karolchik D, *Genomic Data Resources: Challenges and Promises*. Nature Education, 2008. **1**(3): p. 2.

83. Karolchik, D., et al., *The UCSC Table Browser data retrieval tool*. Nucleic Acids Res, 2004. **32**(Database issue): p. D493-6.

84. Kuhn, R.M., D. Haussler, and W.J. Kent, *The UCSC genome browser and associated tools*. Brief Bioinform, 2013. **14**(2): p. 144-61.

85. Kent, W.J., et al., *The human genome browser at UCSC*. Genome Res, 2002. **12**(6): p. 996-1006.

86. Guttman, A., *R-trees: a dynamic index structure for spatial searching*. SIGMOD Rec., 1984. **14**(2): p. 47-57.

87. Lee, B. *Why does UCSC genome browser NOT using spatial index?* ; Available from: http://redmine.soe.ucsc.edu/forum/index.php?t=msg&goto=13979&S=8c781fc1d68484240637e514551f336c.

88. International HapMap, C., *The International HapMap Project*. Nature, 2003. **426**(6968): p. 789-96.

89. Buchanan, C.C., et al., *A comparison of cataloged variation between International HapMap Consortium and 1000 Genomes Project data*. J Am Med Inform Assoc, 2012. **19**(2): p. 289-94.

90. Smigielski, E.M., et al., *dbSNP: a database of single nucleotide polymorphisms*. Nucleic Acids Res, 2000. **28**(1): p. 352-5.

91. Nelson, S.C., et al., *Is 'forward' the same as 'plus'?...and other adventures in SNP allele nomenclature*. Trends Genet, 2012. **28**(8): p. 361-3.

92. *Illumina TOP/BOT convention*. Available from: http://res.illumina.com/documents/products/technotes/technote_topbot.pdf.

93. Tennessen, J.A., et al., *Evolution and functional impact of rare coding variation from deep sequencing of human exomes*. Science, 2012. **337**(6090): p. 64-9.

94. *MySQL*. Available from: http://www.mysql.com.

95. *The Perl Programming Language*. Available from: http://www.perl.org.

96. McLaren, W., et al., *Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor*. Bioinformatics, 2010. **26**(16): p. 2069-70.

97. Wang, K., M. Li, and H. Hakonarson, *ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data*. Nucleic Acids Res, 2010. **38**(16): p. e164.

98. Cingolani, P., 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 (Austin), 2012. **6**(2): p. 80-92.

99. Sana, M.E., et al., *GAMES identifies and annotates mutations in next-generation sequencing projects*. Bioinformatics, 2011. **27**(1): p. 9-13.

100. Ho, E.D., et al., *VAS: a convenient web portal for efficient integration of genomic features with millions of genetic variants*. BMC Genomics, 2014. **15**: p. 886.

101. Bush, W.S. and J.H. Moore, *Chapter 11: Genome-wide association studies*. PLoS Comput Biol, 2012. **8**(12): p. e1002822.

102. Welter, D., et al., *The NHGRI GWAS Catalog, a curated resource of SNP-trait associations*. Nucleic Acids Res, 2014. **42**(Database issue): p. D1001-6.

103. Bamshad, M.J., et al., *Exome sequencing as a tool for Mendelian disease gene discovery*. Nat Rev Genet, 2011. **12**(11): p. 745-55.

104. Cirulli, E.T. and D.B. Goldstein, *Uncovering the roles of rare variants in common disease through whole-genome sequencing*. Nat Rev Genet, 2010. **11**(6): p. 415-25.

105. Tsuji, S., *Genetics of neurodegenerative diseases: insights from high-throughput resequencing*. Hum Mol Genet, 2010. **19**(R1): p. R65-70.

106. Quintáns, B., et al., *Medical genomics: The intricate path from genetic variant identification to clinical interpretation*. Applied & Translational Genomics, 2014. **3**(3): p. 60-67.

107. MacArthur, D.G., et al., *A systematic survey of loss-of-function variants in human protein-coding genes*. Science, 2012. **335**(6070): p. 823-8.

108. MacArthur, D.G., et al., *Guidelines for investigating causality of sequence variants in human disease*. Nature, 2014. **508**(7497): p. 469-76.

109. Ashburner, M., et al., *Gene ontology: tool for the unification of biology. The Gene Ontology Consortium*. Nat Genet, 2000. **25**(1): p. 25-9.

110. Kibbe, W.A., et al., *Disease Ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data.* Nucleic Acids Res, 2014.

111. Kohler, S., et al., *The Human Phenotype Ontology project: linking molecular biology and disease through phenotype data.* Nucleic Acids Res, 2014. **42**(Database issue): p. D966-74.

112. Singleton, M.V., et al., *Phevor combines multiple biomedical ontologies for accurate identification of disease-causing alleles in single individuals and small nuclear families.* Am J Hum Genet, 2014. **94**(4): p. 599-610.

113. Masino, A.J., et al., *Clinical phenotype-based gene prioritization: an initial study using semantic similarity and the human phenotype ontology.* BMC Bioinformatics, 2014. **15**: p. 248.

114. Schlicker, A., T. Lengauer, and M. Albrecht, *Improving disease gene prioritization using the semantic similarity of Gene Ontology terms.* Bioinformatics, 2010. **26**(18): p. i561-7.

115. Kohler, S., et al., *Clinical diagnostics in human genetics with semantic similarity searches in ontologies.* Am J Hum Genet, 2009. **85**(4): p. 457-64.

116. Aerts, S., et al., *Gene prioritization through genomic data fusion.* Nat Biotechnol, 2006. **24**(5): p. 537-44.

117. Chen, J., et al., *ToppGene Suite for gene list enrichment analysis and candidate gene prioritization.* Nucleic Acids Res, 2009. **37**(Web Server issue): p. W305-11.

118. Sifrim, A., et al., *eXtasy: variant prioritization by genomic data fusion.* Nat Methods, 2013. **10**(11): p. 1083-4.

119. Petrovski, S., et al., *Genic intolerance to functional variation and the interpretation of personal genomes.* PLoS Genet, 2013. **9**(8): p. e1003709.

120. Huang, N., et al., *Characterising and predicting haploinsufficiency in the human genome.* PLoS Genet, 2010. **6**(10): p. e1001154.

121. Folkman, L., B. Stantic, and A. Sattar, *Towards sequence-based prediction of mutation-induced stability changes in unseen non-homologous proteins.* BMC Genomics, 2014. **15 Suppl 1**: p. S4.

122. Lazaridis, T. and M. Karplus, *Effective energy functions for protein structure prediction.* Curr Opin Struct Biol, 2000. **10**(2): p. 139-45.

123. Capriotti, E., P. Fariselli, and R. Casadio, *A neural-network-based method for predicting protein stability changes upon single point mutations.* Bioinformatics, 2004. **20 Suppl 1**: p. i63-8.

124. Chun, S. and J.C. Fay, *Identification of deleterious mutations within three human genomes.* Genome Res, 2009. **19**(9): p. 1553-61.

125. Reva, B., Y. Antipin, and C. Sander, *Predicting the functional impact of protein mutations: application to cancer genomics.* Nucleic Acids Res, 2011. **39**(17): p. e118.

126. Thomas, P.D., et al., *PANTHER: a library of protein families and subfamilies indexed by function.* Genome Res, 2003. **13**(9): p. 2129-41.

127. Choi, Y., et al., *Predicting the functional effect of amino acid substitutions and indels.* PLoS One, 2012. **7**(10): p. e46688.

128. Davydov, E.V., et al., *Identifying a high fraction of the human genome to be under selective constraint using GERP++.* PLoS Comput Biol, 2010. **6**(12): p. e1001025.

129. Pollard, K.S., et al., *Detection of nonneutral substitution rates on mammalian phylogenies.* Genome Res, 2010. **20**(1): p. 110-21.

130. Garber, M., et al., *Identifying novel constrained elements by exploiting biased substitution patterns.* Bioinformatics, 2009. **25**(12): p. i54-62.

131. Capriotti, E., R. Calabrese, and R. Casadio, *Predicting the insurgence of human genetic diseases associated to single point protein mutations with support vector machines and evolutionary information.* Bioinformatics, 2006. **22**(22): p. 2729-34.

132. Bromberg, Y. and B. Rost, *SNAP: predict effect of non-synonymous polymorphisms on function.* Nucleic Acids Res, 2007. **35**(11): p. 3823-35.

133. Calabrese, R., et al., *Functional annotations improve the predictive score of human disease-related mutations in proteins.* Hum Mutat, 2009. **30**(8): p. 1237-44.

134. Ramensky, V., P. Bork, and S. Sunyaev, *Human non-synonymous SNPs: server and survey.* Nucleic Acids Res, 2002. **30**(17): p. 3894-900.

135. Camacho, C., et al., *BLAST+: architecture and applications.* BMC Bioinformatics, 2009. **10**: p. 421.

136. Katoh, K. and D.M. Standley, *MAFFT multiple sequence alignment software version 7: improvements in performance and usability.* Mol Biol Evol, 2013. **30**(4): p. 772-80.

137. Thompson, J.D., V. Prigent, and O. Poch, *LEON: multiple aLignment Evaluation Of Neighbours.* Nucleic Acids Res, 2004. **32**(4): p. 1298-307.

138. Wicker, N., et al., *Secator: a program for inferring protein subfamilies from phylogenetic trees.* Mol Biol Evol, 2001. **18**(8): p. 1435-41.

139. Sunyaev, S.R., et al., *PSIC: profile extraction from sequence alignments with position-specific counts of independent observations.* Protein Eng, 1999. **12**(5): p. 387-94.

140. Henikoff, S. and J.G. Henikoff, *Amino acid substitution matrices from protein blocks.* Proc Natl Acad Sci U S A, 1992. **89**(22): p. 10915-9.

141. Finn, R.D., et al., *Pfam: the protein families database.* Nucleic Acids Res, 2014. **42**(Database issue): p. D222-30.

142. Berman, H.M., et al., *The Protein Data Bank.* Acta Crystallogr D Biol Crystallogr, 2002. **58**(Pt 6 No 1): p. 899-907.

143. Joosten, R.P., et al., *A series of PDB related databases for everyday needs.* Nucleic Acids Res, 2011. **39**(Database issue): p. D411-9.

144. Fayyad, U.M. and K.B. Irani, *Multi-interval discretization of continuous-valued attributes for classification learning*, in *International Joint Conference on Artificial Intelligence*1993. p. 1022-1029.

145. Smith, H.O., T.M. Annau, and S. Chandrasegaran, *Finding sequence motifs in groups of functionally related proteins.* Proc Natl Acad Sci U S A, 1990. **87**(2): p. 826-30.
146. Ng, P.C. and S. Henikoff, *Predicting deleterious amino acid substitutions.* Genome Res, 2001. **11**(5): p. 863-74.
147. Gribskov, M., A.D. McLachlan, and D. Eisenberg, *Profile analysis: detection of distantly related proteins.* Proc Natl Acad Sci U S A, 1987. **84**(13): p. 4355-8.
148. Henikoff, J.G. and S. Henikoff, *Using substitution probabilities to improve position-specific scoring matrices.* Comput Appl Biosci, 1996. **12**(2): p. 135-43.
149. Tatusova, T.A. and T.L. Madden, *BLAST 2 Sequences, a new tool for comparing protein and nucleotide sequences.* FEMS Microbiol Lett, 1999. **174**(2): p. 247-50.
150. Siepel, A., et al., *Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes.* Genome Res, 2005. **15**(8): p. 1034-50.
151. Reese, M.G., et al., *Improved splice site detection in Genie.* J Comput Biol, 1997. **4**(3): p. 311-23.
152. Consortium, E.P., et al., *Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project.* Nature, 2007. **447**(7146): p. 799-816.
153. Cooper, G.M., et al., *Distribution and intensity of constraint in mammalian genomic sequence.* Genome Res, 2005. **15**(7): p. 901-13.
154. Blanchette, M., et al., *Aligning multiple genomic sequences with the threaded blockset aligner.* Genome Res, 2004. **14**(4): p. 708-15.
155. Cibulskis, K., et al., *Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples.* Nat Biotechnol, 2013. **31**(3): p. 213-9.
156. *Apache commons.*
157. Li, H., *Tabix: fast retrieval of sequence features from generic TAB-delimited files.* Bioinformatics, 2011. **27**(5): p. 718-9.
158. *CouchDB.*
159. Kohane, I.S., D.R. Masys, and R.B. Altman, *The incidentalome: a threat to genomic medicine.* JAMA, 2006. **296**(2): p. 212-5.
160. *NoSQL databases*. Available from: http://nosql-database.org/.
161. Cattell, R., *Scalable SQL and NoSQL data stores.* SIGMOD Rec., 2011. **39**(4): p. 12-27.
162. *Erlang*. Available from: http://www.erlang.org/.
163. Kimball, R. and M. Ross, *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. 2002: John Wiley \&amp; Sons, Inc. 416.
164. Murphy, S.N., et al., *Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2).* J Am Med Inform Assoc, 2010. **17**(2): p. 124-30.
165. Nadkarni, P.M. and C. Brandt, *Data extraction and ad hoc query of an entity-attribute-value database.* J Am Med Inform Assoc, 1998. **5**(6): p. 511-27.

166. *Amazon AWS*. Available from: http://aws.amazon.com/.
167. *LightCouch Java API*. Available from: http://www.lightcouch.org/.
168. *MxGraph*. Available from: http://www.jgraph.com/mxgraph.html.
169. *Amazon AWS EC2 instances*. Available from: http://aws.amazon.com/ec2/instance-types/.
170. *1000 Genomes Project phase1 integrated release*. Available from: http://www.1000genomes.org/phase1-analysis-results-directory.
171. Valverde, P., et al., *Variants of the melanocyte-stimulating hormone receptor gene are associated with red hair and fair skin in humans.* Nat Genet, 1995. **11**(3): p. 328-30.
172. Innocenti, F., et al., *A genome-wide association study of overall survival in pancreatic cancer patients treated with gemcitabine in CALGB 80303.* Clin Cancer Res, 2012. **18**(2): p. 577-84.
173. *BigCouch*. Available from: http://bigcouch.cloudant.com/.
174. Athey, B.D., et al., *tranSMART: An Open Source and Community-Driven Informatics and Data Sharing Platform for Clinical and Translational Research.* AMIA Jt Summits Transl Sci Proc, 2013. **2013**: p. 6-8.
175. Shihab, H.A., et al., *Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models.* Hum Mutat, 2013. **34**(1): p. 57-65.
176. Gemovic, B., et al., *Feature-based classification of amino acid substitutions outside conserved functional protein domains.* ScientificWorldJournal, 2013. **2013**: p. 948617.
177. Della Mina, E., et al., *Improving molecular diagnosis in epilepsy by a dedicated high-throughput sequencing platform.* Eur J Hum Genet, 2014.
178. Kassahn, K.S., H.S. Scott, and M.C. Caramins, *Integrating massively parallel sequencing into diagnostic workflows and managing the annotation and clinical interpretation challenge.* Hum Mutat, 2014. **35**(4): p. 413-23.
179. Frousios, K., et al., *Predicting the functional consequences of non-synonymous DNA sequence variants--evaluation of bioinformatics tools and development of a consensus strategy.* Genomics, 2013. **102**(4): p. 223-8.
180. Gonzalez-Perez, A. and N. Lopez-Bigas, *Improving the assessment of the outcome of nonsynonymous SNVs with a consensus deleteriousness score, Condel.* Am J Hum Genet, 2011. **88**(4): p. 440-9.
181. Lopes, M.C., et al., *A combined functional annotation score for non-synonymous variants.* Hum Hered, 2012. **73**(1): p. 47-51.
182. Chou, K.C., *Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes.* Bioinformatics, 2005. **21**(1): p. 10-9.
183. Chauhan, J.S., N.K. Mishra, and G.P. Raghava, *Identification of ATP binding residues of a protein from its primary sequence.* BMC Bioinformatics, 2009. **10**: p. 434.

184. Khazanov, N.A. and H.A. Carlson, *Exploring the composition of protein-ligand binding sites on a large scale.* PLoS Comput Biol, 2013. **9**(11): p. e1003321.

185. Szalkowski, A.M. and M. Anisimova, *Markov models of amino acid substitution to study proteins with intrinsically disordered regions.* PLoS One, 2011. **6**(5): p. e20488.

186. de Beer, T.A., et al., *Amino acid changes in disease-associated variants differ radically from variants observed in the 1000 genomes project dataset.* PLoS Comput Biol, 2013. **9**(12): p. e1003382.

187. Khan, S. and M. Vihinen, *Spectrum of disease-causing mutations in protein secondary structures.* BMC Struct Biol, 2007. **7**: p. 56.

188. Stenson, P.D., et al., *The Human Gene Mutation Database: 2008 update.* Genome Med, 2009. **1**(1): p. 13.

189. Chen, X. and H. Ishwaran, *Random forests for genomic data analysis.* Genomics, 2012. **99**(6): p. 323-9.

190. Bahar, I., et al., *Understanding the recognition of protein structural classes by amino acid composition.* Proteins, 1997. **29**(2): p. 172-85.

191. Zhou, G.P. and K. Doctor, *Subcellular location prediction of apoptosis proteins.* Proteins, 2003. **50**(1): p. 44-8.

192. Du, P., et al., *PseAAC-Builder: a cross-platform stand-alone program for generating various special Chou's pseudo-amino acid compositions.* Anal Biochem, 2012. **425**(2): p. 117-9.

193. Ren, S., et al., *Short Linear Motifs recognized by SH2, SH3 and Ser/Thr Kinase domains are conserved in disordered protein regions.* BMC Genomics, 2008. **9 Suppl 2**: p. S26.

194. Romi, H., et al., *Meconium ileus caused by mutations in GUCY2C, encoding the CFTR-activating guanylate cyclase 2C.* Am J Hum Genet, 2012. **90**(5): p. 893-9.

195. Koel, B.F., et al., *Substitutions near the receptor binding site determine major antigenic change during influenza virus evolution.* Science, 2013. **342**(6161): p. 976-9.

196. Vihinen, M., *Majority vote and other problems when using computational tools.* Hum Mutat, 2014. **35**(8): p. 912-4.

197. Bellinger, F.P., et al., *Regulation and function of selenoproteins in human disease.* Biochem J, 2009. **422**(1): p. 11-22.

198. Liu, X., X. Jian, and E. Boerwinkle, *dbNSFP v2.0: a database of human non-synonymous SNVs and their functional predictions and annotations.* Hum Mutat, 2013. **34**(9): p. E2393-402.

199. Asai-Coakwell, M., et al., *Incomplete penetrance and phenotypic variability characterize Gdf6-attributable oculo-skeletal phenotypes.* Hum Mol Genet, 2009. **18**(6): p. 1110-21.

200. Tassabehji, M., et al., *Mutations in GDF6 are associated with vertebral segmentation defects in Klippel-Feil syndrome.* Hum Mutat, 2008. **29**(8): p. 1017-27.

201. Markunas, C.A., et al., *Stratified whole genome linkage analysis of Chiari type I malformation implicates known Klippel-Feil syndrome genes as putative disease candidates.* PLoS One, 2013. **8**(4): p. e61521.

202. Sha, X., L. Yang, and L.E. Gentry, *Identification and analysis of discrete functional domains in the pro region of pre-pro-transforming growth factor beta 1.* J Cell Biol, 1991. **114**(4): p. 827-39.

203. Bird, A.P., *CpG-rich islands and the function of DNA methylation.* Nature, 1986. **321**(6067): p. 209-13.

204. Hasegawa, M. and Y. Shimonishi, *Recognition and signal transduction mechanism of Escherichia coli heat-stable enterotoxin and its receptor, guanylate cyclase C.* J Pept Res, 2005. **65**(2): p. 261-71.

205. Hunter, S., et al., *InterPro in 2011: new developments in the family and domain prediction database.* Nucleic Acids Res, 2012. **40**(Database issue): p. D306-12.

206. Tchernitchko, D., M. Goossens, and H. Wajcman, *In silico prediction of the deleterious effect of a mutation: proceed with caution in clinical genetics.* Clin Chem, 2004. **50**(11): p. 1974-8.

207. Thusberg, J., A. Olatubosun, and M. Vihinen, *Performance of mutation pathogenicity prediction methods on missense variants.* Hum Mutat, 2011. **32**(4): p. 358-68.

208. Oski, F.A., et al., *Congenital hemolytic anemia with high-sodium, low-potassium red cells. Studies of three generations of a family with a new variant.* N Engl J Med, 1969. **280**(17): p. 909-16.

209. Stewart, G.W., *Hemolytic disease due to membrane ion channel disorders.* Curr Opin Hematol, 2004. **11**(4): p. 244-50.

210. Delaunay, J., *The hereditary stomatocytoses: genetic disorders of the red cell membrane permeability to monovalent cations.* Semin Hematol, 2004. **41**(2): p. 165-72.

211. Syfuss, P.Y., et al., *Mild dehydrated hereditary stomatocytosis revealed by marked hepatosiderosis.* Clin Lab Haematol, 2006. **28**(4): p. 270-4.

212. Jais, X., et al., *An extreme consequence of splenectomy in dehydrated hereditary stomatocytosis: gradual thrombo-embolic pulmonary hypertension and lung-heart transplantation.* Hemoglobin, 2003. **27**(3): p. 139-47.

213. Entezami, M., et al., *Xerocytosis with concomitant intrauterine ascites: first description and therapeutic approach.* Blood, 1996. **87**(12): p. 5392-3.

214. Andolfo, I., et al., *Missense mutations in the ABCB6 transporter cause dominant familial pseudohyperkalemia.* Am J Hematol, 2013. **88**(1): p. 66-72.

215. Houston, B.L., et al., *Refinement of the hereditary xerocytosis locus on chromosome 16q in a large Canadian kindred.* Blood Cells Mol Dis, 2011. **47**(4): p. 226-31.

216. Zarychanski, R., et al., *Mutations in the mechanotransduction protein PIEZO1 are associated with hereditary xerocytosis.* Blood, 2012. **120**(9): p. 1908-15.

217. Grootenboer, S., et al., *Pleiotropic syndrome of dehydrated hereditary stomatocytosis, pseudohyperkalemia, and perinatal edema maps to 16q23-q24.* Blood, 2000. **96**(7): p. 2599-605.

218. Iolascon, A., et al., *Familial pseudohyperkalemia maps to the same locus as dehydrated hereditary stomatocytosis (hereditary xerocytosis).* Blood, 1999. **93**(9): p. 3120-3.

219. Vicente-Gutierrez, M.P., et al., *Nonimmune hydrops fetalis due to congenital xerocytosis.* J Perinatol, 2005. **25**(1): p. 63-5.

220. Ami, O., et al., *First-trimester nuchal abnormalities secondary to dehydrated hereditary stomatocytosis.* Prenat Diagn, 2009. **29**(11): p. 1071-4.

221. Robinson, J.T., et al., *Integrative genomics viewer.* Nat Biotechnol, 2011. **29**(1): p. 24-6.

222. Flatt, J.F., et al., *Stomatin-deficient cryohydrocytosis results from mutations in SLC2A1: a novel form of GLUT1 deficiency syndrome.* Blood, 2011. **118**(19): p. 5267-77.

223. Nagase, T., et al., *Prediction of the coding sequences of unidentified human genes. VI. The coding sequences of 80 new genes (KIAA0201-KIAA0280) deduced by analysis of cDNA clones from cell line KG-1 and brain.* DNA Res, 1996. **3**(5): p. 321-9, 341-54.

224. Satoh, K., et al., *A novel membrane protein, encoded by the gene covering KIAA0233, is transcriptionally induced in senile plaque-associated astrocytes.* Brain Res, 2006. **1108**(1): p. 19-27.

225. Bae, C., F. Sachs, and P.A. Gottlieb, *The mechanosensitive ion channel Piezo1 is inhibited by the peptide GsMTx4.* Biochemistry, 2011. **50**(29): p. 6295-300.

226. Kim, S.E., et al., *The role of Drosophila Piezo in mechanical nociception.* Nature, 2012. **483**(7388): p. 209-12.

227. Dubin, A.E., et al., *Inflammatory signals enhance piezo2-mediated mechanosensitive currents.* Cell Rep, 2012. **2**(3): p. 511-7.

228. Coste, B., et al., *Piezo1 and Piezo2 are essential components of distinct mechanically activated cation channels.* Science, 2010. **330**(6000): p. 55-60.

229. Pasini, E.M., et al., *In-depth analysis of the membrane and cytosolic proteome of red blood cells.* Blood, 2006. **108**(3): p. 791-801.

230. Hauser, W.A., J.F. Annegers, and W.A. Rocca, *Descriptive epidemiology of epilepsy: contributions of population-based studies from Rochester, Minnesota.* Mayo Clin Proc, 1996. **71**(6): p. 576-86.

231. Consortium, E., et al., *Genome-wide association analysis of genetic generalized epilepsies implicates susceptibility loci at 1q43, 2p16.1, 2q22.3 and 17q21.32.* Hum Mol Genet, 2012. **21**(24): p. 5359-72.

232. Mefford, H.C., et al., *Genome-wide copy number variation in epilepsy: novel susceptibility loci in idiopathic generalized and focal epilepsies.* PLoS Genet, 2010. **6**(5): p. e1000962.

233. Fokkema, I.F., et al., *LOVD v.2.0: the next generation in gene variant databases.* Hum Mutat, 2011. **32**(5): p. 557-63.

234. Leger, P.L., et al., *The location of DCX mutations predicts malformation severity in X-linked lissencephaly.* Neurogenetics, 2008. **9**(4): p. 277-85.

235. Bonaglia, M.C., et al., *Molecular mechanisms generating and stabilizing terminal 22q13 deletions in 44 subjects with Phelan/McDermid syndrome.* PLoS Genet, 2011. **7**(7): p. e1002173.

236. Striano, P., et al., *Two novel ALDH7A1 (antiquitin) splicing mutations associated with pyridoxine-dependent seizures.* Epilepsia, 2009. **50**(4): p. 933-6.

237. Kiechl, S., et al., *Two families with autosomal dominant progressive external ophthalmoplegia.* J Neurol Neurosurg Psychiatry, 2004. **75**(8): p. 1125-8.

238. Weckhuysen, S., et al., *KCNQ2 encephalopathy: emerging phenotype of a neonatal epileptic encephalopathy.* Ann Neurol, 2012. **71**(1): p. 15-25.

239. Marshall, C.R., et al., *Infantile spasms is associated with deletion of the MAGI2 gene on chromosome 7q11.23-q21.11.* Am J Hum Genet, 2008. **83**(1): p. 106-11.

240. Kato, M., et al., *Clinical spectrum of early onset epileptic encephalopathies caused by KCNQ2 mutation.* Epilepsia, 2013. **54**(7): p. 1282-7.

241. Bonanni, P., et al., *Generalized epilepsy with febrile seizures plus (GEFS+): clinical spectrum in seven Italian families unrelated to SCN1A, SCN1B, and GABRG2 gene mutations.* Epilepsia, 2004. **45**(2): p. 149-58.

242. Macdonald, R.L., J.Q. Kang, and M.J. Gallagher, *Mutations in GABAA receptor subunits associated with genetic epilepsies.* J Physiol, 2010. **588**(Pt 11): p. 1861-9.

243. Klassen, T., et al., *Exome sequencing of ion channel genes reveals complex profiles confounding personal risk assessment in epilepsy.* Cell, 2011. **145**(7): p. 1036-48.

244. Heinzen, E.L., et al., *Exome sequencing followed by large-scale genotyping fails to identify single rare variants of large effect in idiopathic generalized epilepsy.* Am J Hum Genet, 2012. **91**(2): p. 293-302.

245. Drew, A.P., I.P. Blair, and G.A. Nicholson, *Molecular genetics and mechanisms of disease in distal hereditary motor neuropathies: insights directing future genetic studies.* Curr Mol Med, 2011. **11**(8): p. 650-65.

246. Ludolph, A.C., et al., *Tauopathies with parkinsonism: clinical spectrum, neuropathologic basis, biological markers, and treatment options.* Eur J Neurol, 2009. **16**(3): p. 297-309.

247. Caparros-Lefebvre, D., et al., *Guadeloupean parkinsonism: a cluster of progressive supranuclear palsy-like tauopathy.* Brain, 2002. **125**(Pt 4): p. 801-11.

248. Gudbjartsson, D.F., et al., *Allegro, a new computer program for multipoint linkage analysis.* Nat Genet, 2000. **25**(1): p. 12-3.

249. Lippa, C.F., et al., *Frontotemporal dementia with novel tau pathology and a Glu342Val tau mutation.* Ann Neurol, 2000. **48**(6): p. 850-8.

250. Lynch, T., et al., *Clinical characteristics of a family with chromosome 17-linked disinhibition-dementia-parkinsonism-amyotrophy complex. 1994.* Neurology, 2001. **57**(10 Suppl 3): p. S39-45.

251. Whitwell, J.L., et al., *Atrophy patterns in IVS10+16, IVS10+3, N279K, S305N, P301L, and V337M MAPT mutations.* Neurology, 2009. **73**(13): p. 1058-65.

252. Higuchi, M., et al., *Transgenic mouse model of tauopathies with glial pathology and nervous system degeneration.* Neuron, 2002. **35**(3): p. 433-46.

253. Nicholl, D.J., et al., *An English kindred with a novel recessive tauopathy and respiratory failure.* Ann Neurol, 2003. **54**(5): p. 682-6.

254. Andreadis, A., *Tau gene alternative splicing: expression patterns, regulation and modulation of function in normal brain and neurodegenerative diseases.* Biochim Biophys Acta, 2005. **1739**(2-3): p. 91-103.

255. Rehm, H.L., et al., *ACMG clinical laboratory standards for next-generation sequencing.* Genet Med, 2013. **15**(9): p. 733-47.

Appendix *A*

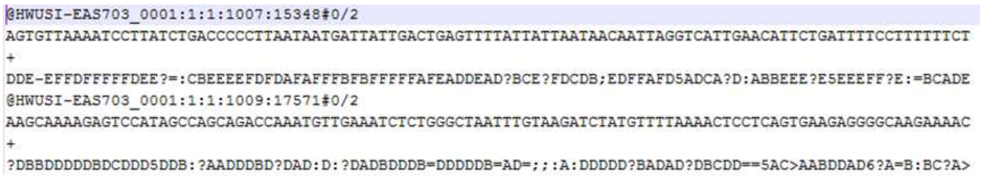
A.1 Data Format

Hereby several genomic data formats discussed in the thesis are reported.

Fastq

Fastq is a standard format to store DNA sequencing reads and their quality scores in Phred [13] scale. Qualities are encoded in ASCII code starting from the 33rd character and ranging 40 possible values.
A fastq file uses 4 lines to represent a single read:

- the header representing the ID of the read
- the sequence, a string of 5 possible characters: A,C,G,T and N
- a comment line (beginning with "+" symbol)
- the quality scores, one for each base, ASCII coded



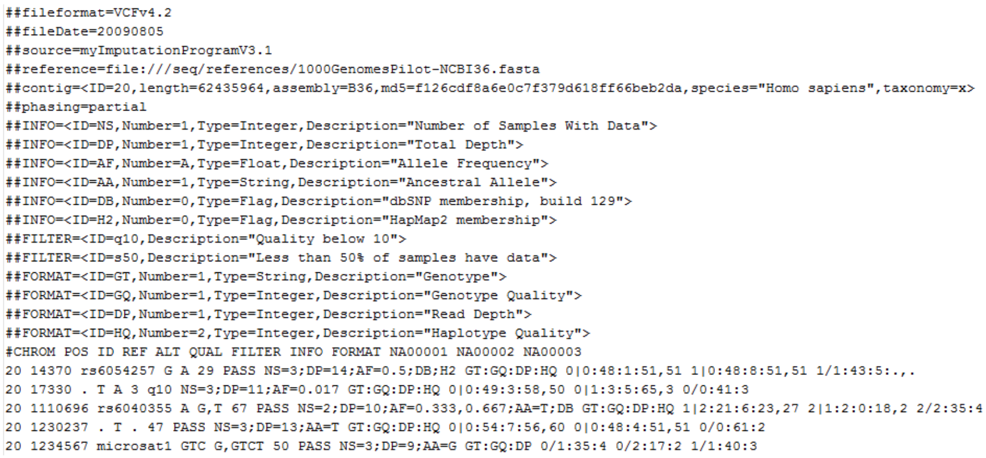
```
@HWUSI-EAS703_0001:1:1:1007:15348#0/2
AGTGTTAAAATCCTTATCTGACCCCCTTAATAATGATTATTGACTGAGTTTTATTATTAATAACAATTAGGTCATTGAACATTCTGATTTTCCTTTTTTCT
+
DDE-EFFDFFFFFDEE?=:CBEEEEFDFDAFAFFFBFBFFFFFAFEADDEAD?BCE?FDCDB;EDFFAFD5ADCA?D:ABBEEE?E5EEEFF?E:=BCADE
@HWUSI-EAS703_0001:1:1:1009:17571#0/2
AAGCAAAAGAGTCCATAGCCAGCAGACCAAATGTTGAAATCTCTGGGCTAATTTGTAAGATCTATGTTTTAAAACTCCTCAGTGAAGAGGGGCAAGAAAAC
+
?DBBDDDDDBDCDDD5DDB:?AADDDBD?DAD:D:?DADBDDDB=DDDDDB=AD=;;:A:DDDDD?BADAD?DBCDD==5AC>AABDDAD6?A=B:BC?A>
```

Figure A 1. an example of two reads in Fastq format.

Variant Calling Format (VCF)

VCF is a text file format introduce by 1000 Genome Project to store genomic variants belonging to one or more samples. The most recent version is the 4.2.

It comprises meta-informations, a header and a body with a line for each variant. Each line contains information about genomic variant position on the reference genome, eventually variant caller filters, base coverage information and genotypes on reported samples.



```
##fileformat=VCFv4.2
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT NA00001 NA00002 NA00003
20 14370 rs6054257 G A 29 PASS NS=3;DP=14;AF=0.5;DB;H2 GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20 17330 . T A 3 q10 NS=3;DP=11;AF=0.017 GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3 0/0:41:3
20 1110696 rs6040355 A G,T 67 PASS NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2 2/2:35:4
20 1230237 . T . 47 PASS NS=3;DP=13;AA=T GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20 1234567 microsat1 GTC G,GTCT 50 PASS NS=3;DP=9;AA=G GT:GQ:DP 0/1:35:4 0/2:17:2 1/1:40:3
```

Figure A 2. An example of VCF file

Meta-informationis in a key=value format and includes the description of the attributes that describe a variant.

Each line is tab-delimited and holds all variant data. Variant is univocally identified by the first five fields consisting of chromosome (CHR), relative position (POS), a dbSNP id (ID), reference base (REF) and comma-separated alternative bases (ALT). QUAL is a number correlated to variant quality. FILTER is a free text field with short name of the applied filter usually explained by meta-information. INFO can contains severaldata aggregates by samples: variant coverage, allele frequencies,number of samples holding variant. Variant sequence context, presence in variant databases and other optional annotations are usually defined in this field as well. FORMAT indicates how the following genotype fields are formatted and meta-information explain format abbreviations. Genotype fields having as the header the sample ids usually hold data on genotype by a numbered codification:0=REF, 1=first allele in ALT, 2 second allele in ALT etc. Genotype can be phased "|" or not "/". Other genotype field components can be coverage for each allele and genotype quality.

A.2 Code Snippets

Herby diverse code snippets and related /*comments*/ are reported.

RDBMS-VMS

ServletNGS.java

```
import javax.servlet.http.*;
.
.
public class ServletNgs extends HttpServlet{
```

```
/*Properties file with infos on database,directories
etc.*/
 private final static String PROPERTIES_FILE =
"/conf/ngs.properties";
.
.
/*”cmd” is the HttpServletRequest parameter used to call
the a specific method of the ServletNGS class*/

private final static String COMMAND = "cmd";

private ConnectionPool connPool=null;
.
.

/* Method “init” that is called only once at Servlet
initialization. It load properties and initialize the
Connection Pool to the database*/

public void init(ServletConfig config) throws
ServletException {
       super.init(config);
       URL url =
getClass().getResource(PROPERTIES_FILE);
       Properties properties = new Properties();

       try{
             connPool=ConnectionPool.getConnectionPool();
             }catch (ConnectionPoolException c){
                   System.out.println(c);
             }

       try {
             properties.load(url.openStream());
       } catch (IOException e) {
             throw new ServletException(e.getMessage());
       }
.
.
/* Method “doPost” tells the ServletNGS what to do in
case of HTTP POST requests. It search for the COMMAND
parameter to which is associated a specific method.*/

protected void doPost(HttpServletRequest request,
HttpServletResponse response)
 throws ServletException,IOException{

int command;
       try {
             command =
Integer.parseInt(request.getParameter(COMMAND));
       } catch (NumberFormatException ex) {
             command = 1;
       }

       switch (command) {

       case 1:
             try {readMarkers(request, response);
             }
       catch (SQLException e) {
             // TODO Auto-generated catch block
             e.printStackTrace();
       }
       break;
       .
       .
       case 12:
             try {chooseCaseAndControls(request, 
response);
             }catch (ConnectionPoolException e) {
             // TODO Auto-generated catch block
             e.printStackTrace();
       }
       break;
       .
       .
}

/*the method “readMarkers” calls the method 
“getMarker()”of the DBAnalize class which returns an 
iterable object that will be forwarded to the 
“markers.jsp” page as an attribute*/

private void readMarkers(HttpServletRequest request, 
HttpServletResponse response)
throws ServletException, IOException, SQLException {

 request.setAttribute("list", ((new 
DBAnalize(connPool)).getMarker()).iterator());
 RequestDispatcher 
rd=getServletContext().getRequestDispatcher("/markers.js
p");
 rd.forward(request, response);
}
.
.
```

ConnectionPool.java

```
import java.net.URL;
import java.sql.*;
.
.
public class ConnectionPool {

  /* The variable managing the only instance of 
ConnectionPool*/

private static ConnectionPool connectionPool = null;


/* queue of free connections */

private Vector freeConnections;

/* database driver */

private String dbDriver;

/* ConnectionPool constructor*/

  private ConnectionPool() throws 
ConnectionPoolException {
    freeConnections = new Vector();
```

```
/* Method that load parameters containing database infos
such as driver*/

    loadParameters();

/* Method that load database drivers*/

    loadDriver();
  }
.
.

/*invoke the constructor and return this class*/

public static synchronized ConnectionPool
getConnectionPool()
  throws ConnectionPoolException {
    if(connectionPool == null) {
      connectionPool = new ConnectionPool();
    }
    return connectionPool;
  }
.
.
private void loadParameters() {..}
.
.
private void loadDriver() {..}
.
.
/*The method “getConnection” returns a free connection
by pulling it from the queue of the free connections or,
in case of no available connections, it creates a new
one*/

public synchronized Connection getConnection()
  throws ConnectionPoolException {
    Connection con;

    if(freeConnections.size() > 0) {
      con = (Connection)freeConnections.firstElement();
      freeConnections.removeElementAt(0);
try {
        if(con.isClosed()) {
con = getConnection();
        }
      }
      catch(SQLException e) {
        con = getConnection();
      }
    }
    else {
      con = newConnection();
    }
    return con;
  }
.
.

/*”newConnection” builds up a new connection to the
database*/

private Connection newConnection() throws
ConnectionPoolException {
    Connection con = null;
    try {
      con = DriverManager.getConnection(
             dbUrl,
             dbLogin,
             dbPassword);  // crea la connessione
    }
    catch(SQLException e) {
throw new ConnectionPoolException();
 }
    return con;
  }
.
.
```

ReadVCFCallable.java

```
import java.util.concurrent.Callable;
.
.
/*The class ReadVcfCallable implements the Callable
interface for concurrent programming. SerlvletNGS push
this object into a ThreadPoolExecutor with a core pool
size equal to one*/

public class ReadVcfCallable implements Callable {

 private InputStream is;
 private String id_sample;
 private String build;

/*the ReadVcf class implements methods to parse and
manipulate the VCF file*/

 private ReadVcf read;
 private ConnectionPool connPool;

/*constructor*/

  public ReadVcfCallable(InputStream ais, String
aid_sample, String abuild,ReadVcf read,ConnectionPool
aconnPool) {
        is=ais;
        id_sample=aid_sample;
        build=abuild;
        this.read=read;
        connPool=aconnPool;

  }


/*override of he call() method*/

public boolean call() {
        try{

/* parse the VCF by a ReadVCF method*/


read.parseMutationVCF(is,id_sample,build,connPool);

.
.
.

       }

       return 1;

  }


}
```

PredictionsRunnable.java

```
import java.util.concurrent.Runnable;

/*This class implements the Runnable interface.
ServletNGS instanciates two objects (one per prediction
tools differentiated by the passed integer code)and
start two threads in parallel*/

public class PredictionsRunnable implements Runnable {

    private ConnectionPool conn;
    private int code;
    private Vector mutations;
    private static final int DELAY = 100;
.
.
    public PredictionsRunnable(ConnectionPool
conn,Vector mutations,SharedData data, int code) {
            this.mutations=mutations;
            this.conn=conn;
            this.data=data;
            this.code=code
          }
.
.
public void run() {
       try {

         switch (code){

/* MutationTaster */
            case 0:
                  try {

/* calls the method that send the data via HTTP POST to
MutationTaster web service and retrieve results through
HTTP GET method. Insert the results into the database*/


                  Vector mutOupDate1=new
PredictionTools().MutationTasterPrediction(conn,mutation
s,dir);
                        boolean
data.MutationTasterUpdate(mutOupDate1);
.
.
}
            break;

            /*PolyPhen-2*/
```

```
                 case 1:

                       try {

     Vector mutOupDate2=new
     PredictionTools().PolyphenPrediction(conn,mutations);

                             boolean
     data.PolyphenUpdate(mutOupDate2);
     .
     .
     break;


              }
     Thread.sleep(DELAY);

            }
     .
     .
}
```