



André Santos

DEVELOPMENT OF A DATABASE FOR ARRAY COMPARATIVE GENOMIC HYBRIDIZATION

Development presented to the University of Coimbra to complete the necessary requirements for obtaining the degree of Master Science in Biomedical Engineering, held under scientific orientation of Maria Joana Lima Barbosa Melo, PhD, Helmut Wolters, PhD and João Carlos Lopes Carvalho, PhD

September 2015



UNIVERSIDADE DE COIMBRA



Universidade de Coimbra

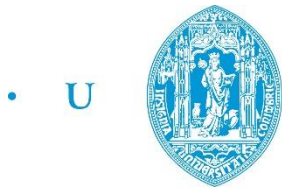
Faculdade de Ciências e Tecnologia

Departamento de Física

**Development of a Database for Array
Comparative Genomic Hybridization**

André Ferreira Santos

Coimbra, 2015



• C •

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

André Ferreira Santos

Development of a Database for Array Comparative Genomic Hybridization

Dissertation submitted to the
Faculty of Sciences and Technology
In partial fulfilment of the requirements
For the Master degree in Biomedical Engineering

Mentors:

Prof. Dr^a. Maria Joana Lima Barbosa Melo

Prof. Dr. Helmut Wolters

Prof. Dr. João Carlos Lopes Carvalho

Coimbra, 2015

Este trabalho foi desenvolvido em colaboração com:

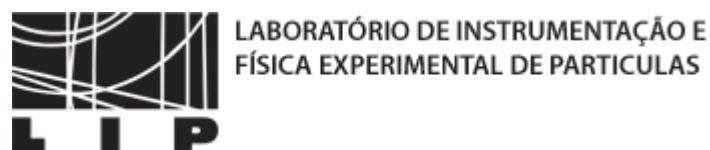
Laboratório de Citogenética e Genómica da Faculdade de Medicina da
Universidade de Coimbra



Centro de Física da Universidade de Coimbra



Laboratório de Instrumentação e Física Experimental de Partículas



Esta cópia da tese é fornecida na condição de que quem a consulta reconhece que os direitos de autor são pertença do autor da tese e que nenhuma citação ou informação obtida a partir dela pode ser publicada sem a referência apropriada.

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognize that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without proper acknowledgement.

Acknowledgements

I would like to thank my scientific mentors, Prof. Dr^a. Joana Melo, Prof. Dr. Helmut Wolters and Prof. Dr. João Carvalho for their support, knowledge and advices for this work.

A special thanks to the laboratory geneticists Miguel Pires and Susana Ferreira for their total availability to help in understanding the arrayCGH technique.

I would also like to extend my thanks to all members of the *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra*, coordinated by Prof. Dr^a. Isabel Carreira, for receiving me in such a nice environment.

I wish to thank to all my friends who directly or indirectly helped in the development of this work.

Finally, I wish to thank to my family, my sisters and especially my mother and my father for their support and encouragement throughout my study.

Abstract

Microarray Comparative Genomic Hybridization (arrayCGH) allowed a significant advance on the diagnose of unexplained development disorders by detecting genomic copy number variations (CNVs) that were previously undetectable by other types of cytogenetic technologies. Well characterized genetic syndromes are detected and also new genomic disorders and diseases causing CNVs are being discovered through the utilization of this technique.

The pathogenicity assessment of the CNVs detected by arrayCGH are directly or indirectly related with a query of previously recorded and classified CNVs. When a CNV that has not been observed before appears, clinicians have to interpret the variant of uncertain clinical significance (VOUS), which can be very challenging for them. However, sharing classified CNVs between laboratories can be helpful when facing a VOUS in order to help its classification as benign or pathogenic.

Easily recording, querying and sharing information about CNVs is very important for clinicians and laboratories in order to accomplish the right diagnostic for their patients. In this work is presented the development of a database, using a LAMP technology (Linux, Apache, MySQL and PHP), to store arrayCGH records and of an interface that laboratory clinicians can use for querying and management of CNVs.

Resumo

A técnica de Microarray de Hibridização Genómica Comparativa (arrayCGH) permitiu um avanço significativo no diagnóstico de doenças de desenvolvimento incompreensíveis através da deteção de variações do número de cópia genómicas (CNVs) que antes eram indetetáveis por outros tipos de tecnologias citogenéticas. Através desta técnica são detetadas síndromes genéticas já identificadas mas também novos distúrbios e novas doenças genómicas causadas por CNVs.

A avaliação da patogenicidade das CNVs detetadas por arrayCGH está direta ou indiretamente relacionada com a consulta de CNVs previamente classificadas e arquivadas. Quando é detetada uma CNV anteriormente não observada, os clínicos tem de interpretar uma variação de significado clínico incerto (VOUS) o que pode ser muito complicado para eles. No entanto, a partilha de CNVs classificadas entre laboratórios pode ajudar quando uma variação destas aparece, de modo a poder classificá-la como benigna ou patogénica.

Guardar, consultar e partilhar informação sobre CNVs de uma forma fácil é muito importante para os clínicos e para os laboratórios de modo a que estes possam fazer um diagnóstico correto dos seus pacientes. Neste trabalho é apresentado o desenvolvimento de uma base de dados, usando a tecnologia LAMP (Linux, Apache, MySQL e PHP), para guardar registos de arrayCGH e para dar aos clínicos laboratoriais uma interface para consultar e gerir CNVs.

Content

Acknowledgements.....	i
Abstract.....	iii
Resumo.....	v
Content.....	vii
Acronyms.....	xi
List of Figures.....	xiii
List of Tables.....	xvi
1. Introduction.....	1
1.1. Motivation.....	1
1.2. Goals.....	2
2. Background.....	3
2.1. The arrayCGH technique.....	3
2.2. The laboratory steps of the arrayCGH technique.....	5
2.3. The interpretation of an arrayCGH result.....	9
2.4. The importance of databases in the interpretation of a CNV.....	13
2.4.1. Interpretation of CNV at <i>Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra</i>	16
2.5. Management of arrayCGH results at <i>Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra</i>	16
3. Software Development.....	19
3.1 Software Requirements.....	19
3.1.1. General requirements.....	20
3.1.2. Researcher requirements.....	21
3.1.3. Administrator requirements.....	21
3.1.4. Super Administrator Requirements.....	22
3.2. Software Design.....	23
3.2.1. Use Case Diagrams (UCD).....	23
3.2.2. Entity-Relationship Model (ERM) and Relational Model (RM).....	26

3.3. Server: LAMP technology	32
3.3.1. Linux Operating System.....	32
3.3.2. Apache Web Server	33
3.3.3. MySQL Database Server	33
3.3.4. PHP Scripting Language.....	33
3.4. Materials	34
3.4.1. Software versions and hardware	34
3.4.2. Characterization of arrayCGH results provided for the development of arrayCGH DB	34
4. Security.....	37
4.1. User accesses	37
4.2. Clinic Case management and access	38
4.3. Data Removal	38
4.4. SQL injection.....	39
4.5. Data Backup	39
4.6. Ubuntu Security.....	39
5. Results and Discussion	43
5.1. Log in and Sign in.....	43
5.2. Users Navigation Bar	45
5.3. Software Main Features	47
5.3.1. Upload of Clinic Cases	47
5.3.2. Display CNV matching	50
5.4. Other Features	53
5.4.1. Management of Clinic Cases	53
5.4.2. Management of Users.....	59
5.4.3 Management of Laboratories.....	62
5.5. Validation	64
5.5.1 Distribution of CNVs by size	64

5.5.2. Distribution of CNVs by alteration type type	66
5.5.3. Distribution of CNVs alteration type by size	66
5.5.4. Distribution of CNVs per chromosome and Distribution of CNVs alteration type per chromosome	67
5.5.5. Distribution of CNVs class	70
5.5.6. Distribution of CNVs class by size.....	71
6. ArrayCGH DB Fixes	73
6.1. Negative Size Fix.....	73
6.2. Class Fix	76
6.3. Alteration Type Fix	79
7. Overview, Conclusions and Future Perspectives	81
7.1. Overview	81
7.2. Conclusions and Future Perspectives.....	83
References.....	85

Acronyms

1NF	First Normal Form
ArrayCGH	Microarray Comparative Genomic Hybridization
CNV	Copy Number Variance
DB	Database
DNA	Deoxyribonucleic Acid
ERM	Entity-Relationship Model
FK	Foreign Key
GD	Graphics Draw
gDNA	Genomic Deoxyribonucleic Acid
GIF	Graphics Interchange Format
GNU	GNU's NOT UNIX
GPL	General Public License
HTML	HyperText Markup Language
ID	Identifier
JPEG	Joint Photographic Experts Group
LAMP	Linux Apache MySQL PHP
LTS	Long Term Service
MySQL	My Structured Query Language
NASA	National Aeronautics and Space Administration
OMIM	Online Mendelian Inheritance in Man
PHP	Hypertext Preprocessor
PK	Primary Key
PNG	Portable Network Graphics
RM	Relational Model
SQL	Structured Query Language
UCD	Use Case Diagram
UCSC	University of California, Santa Cruz
URL	Uniform Resource Locator
VOUS	Variance Of Uncertain Significance

List of Figures

Figure 1 – Schematic representation of array CGH technique.	5
Figure 2 – Laboratory workflow of the arrayCGH technique.	6
Figure 3 Fluorescence intensities of four microarrays experiments.	8
Figure 4 - Scanned images of microarrays with quality defects displayed.	9
Figure 5 – Genomic imbalances via log2 ratio.	10
Figure 6 – Requirements extracted from the researchers first request.	19
Figure 7 – Use Case Diagram of a Researcher.	24
Figure 8 – Use Case Diagram of an Administrator.	24
Figure 9 – Use Case Diagram of a Super Administrator.	25
Figure 10 - Entity-Relationship Model.	26
Figure 11 - Relational Model.	29
Figure 12 – Example: A table with non-atomic data where several values of same data type are allowed in genes column.	30
Figure 13 – Example: A table with non-atomic data where multiple columns with same data type are allowed.	30
Figure 14 – Example: A table where the ID is the PK.	31
Figure 15 – Example: PK and FK.	31
Figure 16 – Home page of Array CGH DB.	43
Figure 17 – Sign up page of Array CGH DB.	44
Figure 18 – Successful “Sing up” message.	45
Figure 19 – Log in page of Array CGH DB.	45
Figure 20 –Researcher: Home page and Navigation Bar.	46
Figure 21 – Administrator: Home page and Navigation Bar.	46
Figure 22 –Super-Administrator: Home page and Navigation Bar.	47
Figure 23 - Researcher: Upload Clinic Case page	48
Figure 24 – Researcher: Upload Report of Clinic Cases	49
Figure 25 Researcher: Upload Report of Clinic Cases.	50
Figure 26 – Researcher: CNV match with chromosome location.	51
Figure 27 - Researcher: CNV match with gene name	52
Figure 28 – Researcher: Clinic Case page.	54
Figure 29 – Super-Administrator: Clinic Case page.	55
Figure 30 – Super Administrator: Clinic Cases sorted by genre.	56

Figure 31 – Researcher: Edit Clinic Case page.....	56
Figure 32 – Researcher: View Clinic Case page.....	57
Figure 33 – Researcher: Edit CNV page.....	58
Figure 34 – Researcher: Remove Clinic Case page.....	59
Figure 35 – Super-Administrator: “User” management page.....	60
Figure 36 - Administrator: “User” management page.....	60
Figure 37 – Super-Administrator: Remove User page.....	61
Figure 38 Super-Administrator: Message displayed when changing a user from blocked to approved.....	61
Figure 39 – Super-Administrator: User approved.....	62
Figure 40 – Administrator: Message shown when an Administrator tries to remove a user he has not permission to.....	62
Figure 41 – Super-Administrator: Laboratory Management page.....	63
Figure 42 – Super-Administrator: Remove Laboratory page.....	63
Figure 44 - CNVs Size Overview.....	65
Figure 45 - CNVs stored in the arrayCGH DB alteration type overview.....	66
Figure 46 –Distribution of CNVs alteration Type by CNV size.....	67
Figure 49 – Distribution of CNVs by class.....	70
Figure 50 – Distribution of CNVs Class by size.....	71
Figure 51 – Upload report of a Clinic Case with one CNV where Start is greater than Stop.....	74
Figure 52 – Clinic Case where one CNV has a Start greater than Stop and consequently has negative size.....	74
Figure 53 – Editing of Stop from the wrong CNV of Figure 52, after consulting the respective “IntervalBasedReport” where the values were correct.....	75
Figure 54 – Clinic Case of Figure 52 after being edited.....	76
Figure 55 – Clinic Case with several CNVs with wrong class (sup).....	77
Figure 56 – 4 CNVs with wrong class (IIB) caused by a misspelling on the uploaded spreadsheet.....	78
Figure 57 – Two Clinic Cases with wrong class which are not stored on ArrayCGH DB.....	78
Figure 58 – Clinic Case with one CNV where the alteration type is Triplication.....	79
Figure 59 – CNV match where one CNV with alteration type of Triplication appears.....	80

List of Tables

Table I - Requirement of gDNA Input Amount and Volume per Microarray.	6
Table II - Example of an arrayCGH result.	12
Table III - Assessment of Pathogenicity of a CNV.....	14
Table IV – CNV classification used at Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra.	16
Table V – Characterization of arrayCGH results for “IntervalBasedReport” and “Tabela Classes” formats.....	35

1. Introduction

1.1. Motivation

Microarray Comparative Genomic Hybridization (arrayCGH) allowed a significant advance on the diagnose of unexplained development diseases by detecting genomic copy number variations (CNVs) that were previously undetectable by other types of cytogenetic technologies. Well characterized genetic syndromes are detected and also new genomic disorders and diseases causing CNVs are being discovered through the utilization of this technique.

The pathogenicity assessment of the CNVs detected by arrayCGH are directly or indirectly related with a query of previously recorded and classified CNVs. When a CNV that has not been observed before appears, clinicians have to interpret the variance of uncertain significance (VOUS), which can be very challenging for them. However, sharing classified CNVs between laboratories can be helpful when facing a VOUS in order to help its classification as benign or pathogenic.

At Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra, the arrayCGH technique is performed since 2011. From that year until the beginning of 2015, the clinicians of this laboratory saved arrayCGH results of their patients in spreadsheet files to help in future clinical diagnosis of new patients. Until the beginning of 2015, they kept a record of about 4000 arrayCGH results. However, as the number of arrayCGH results increased, the laboratory clinicians start to face added difficulties to find the results they needed to help in the decision about the pathogenicity of a CNV in recent arrayCGH results. This situation prompted them to look into a more efficient way to save and query their results.

As a solution, the clinicians chose to develop an in-house relational Database (DB) to store their results and an application which allows them to feed and query the DB and to share their results with others.

1.2. Goals

The main goal of this project is develop a Database for Array Comparative Genomic Hibridization and an application which allows uploading spreadsheets of arrayCGH results performed at *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra*, and also to query about CNVs stored on the database, in order to help on daily clinical practice.

Another goal is allowing the share of stored results with other clinicians from Portugal and also store results from other Portuguese laboratories.

2. Background

2.1. The arrayCGH technique

Microarray comparative genomic hybridization (arrayCGH) is a cytogenomic technique used to detect copy number variations (CNV) in chromosomes which are too small to be seen down a microscope. It has a higher resolution than G-banded karyotyping which has been the standard first-tier test for detection of chromosomal alterations for more than 35 years¹. Earlier, the arrayCGH technique allowed resolutions larger than 200 Kb, however its resolution level was increasing every year since its first appearance², reaching a resolution of about 1Kb (smaller than the average gene)³. The arrayCGH resolution exceed the resolution of G-banded karyotyping (with resolutions about 5-10 Mb)³ and is more precise at mapping chromosomal aberrations when compared with traditional cytogenetic techniques².

This technique was first developed as a research tool for the investigation of genomic alterations in cancer⁴ and later started to be used as a clinical diagnosis tool for patients with an unexplained development delay or multiple congenital abnormalities^{1,4}.

It is based on traditional metaphase CGH technique and it compares a patient DNA sample against a control DNA sample. Equal amounts of a patient DNA sample and a control DNA sample are labelled with two different fluorescent dyes and are competitively hybridized on a microarray. This is spotted with millions of DNA probes that have been robotically printed to represent the entire human genome or only a specific region. Each probe is spotted with millions of oligonucleotides where the DNA samples will hybridize. For regions where there is no genomic differences between the patient DNA and the control DNA, equal amounts of each DNA sample will bind to the probes. But, if the patient has a gain or a loss of genetic material, there will be a difference between the amounts of each DNA sample bound to the probes, and as a consequence, there will be different levels of fluorescence.^{4,5}

A computer imaging program evaluates the fluorescence level of each probe in the microarray and returns the correspondent logarithmic values of the ratios (\log_2 scale). These values carry the information about the proportion of the control DNA sample and the patient DNA sample bound to the probes. With this information, chromosome regions that have been amplified or deleted, known as CNV, are identified on the patient DNA and are interpreted and reported by a clinic laboratory geneticist.⁴

5

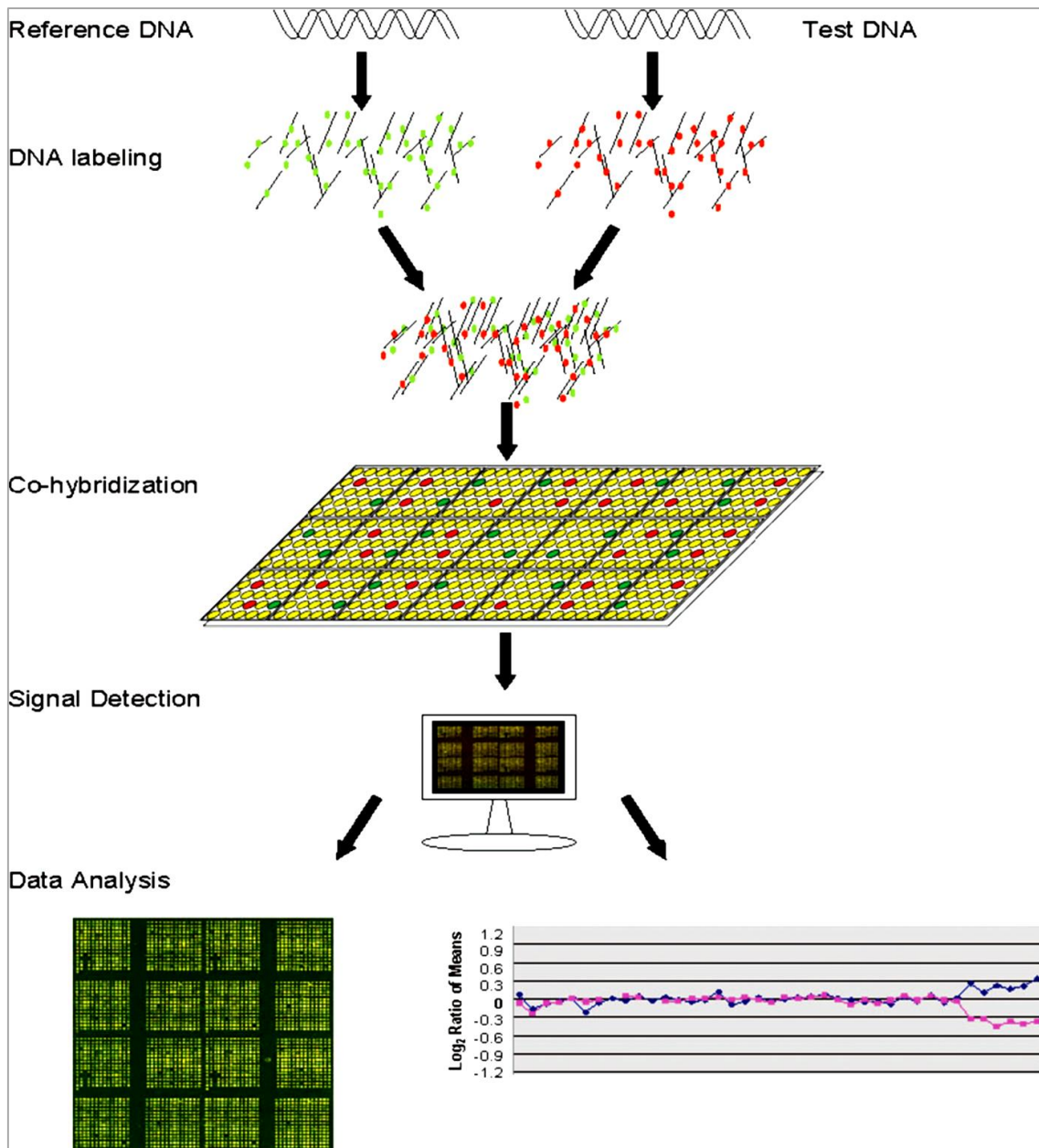


Figure 1 – Schematic representation of array CGH technique. Equal amounts of a patient DNA sample (right) and a control DNA sample (left) are labelled with two different fluorescent dyes and are competitively hybridized on a microarray. A computer imaging program evaluates the fluorescence level of each probe in the microarray (bottom left) and returns the correspondent logarithmic values of the ratios, \log_2 scale (bottom right). Figure from Bejjani, B., and Lisa, S. (2006). Application of Array-Based Comparative Genomic Hybridization to Clinical Diagnostics. *Journal of Molecular Diagnostics*, Vol. 8, N^o. 5. DOI: 10.2353/jmoldx.2006.060029

2.2. The laboratory steps of the arrayCGH technique

Figure 2 shows an overview of the laboratory workflow which takes normally six days.

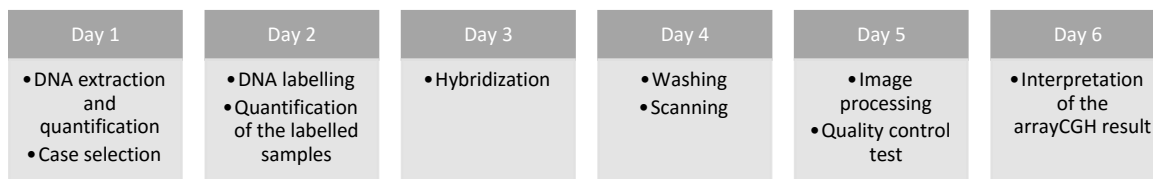


Figure 2 – Laboratory workflow of the arrayCGH technique.

Briefly, the main points of arrayCGH workflow are :

1. **DNA extraction and quantification:** the patient DNA sample can be obtained by any type of DNA extraction that yields high molecular DNA suitable for use in arrayCGH. The control DNA sample can be taken from blood lymphocytes from any karyotypically healthy normal individual. After that, a DNA quantification is performed for each sample because different microarray formats require different DNA quantities (Table I).⁵

Table I - Requirement of gDNA Input Amount and Volume per Microarray. Table from: Agilent Oligonucleotide Array-Based CGH for Genomic DNA Analysis Protocol. Version 7.3. Agilent Technologies, Inc. 5301 Stevens Creek Blvd, Santa Clara, CA 95051 USA. March 2014. Manual Part Number G4410-90010.

Microarray format	gDNA input amount requirement (µg)	Volume of gDNA with restriction digestion (µL)	Volume of gDNA without restriction digestion (µL)
1-pack	0.5 to 1.0	20.2	26
2-pack or 4-pack	0.5 to 1.0	20.2	26
8-pack	0.2 to 0.5	10.1	13

2. **Case selection:** on this step a patient DNA sample and a control DNA sample are grouped by gender. Male patient DNA samples are grouped with male control DNA samples and female patient DNA samples are grouped with female DNA control samples.²

3. **DNA labelling:** the patient DNA sample and the control DNA sample are labelled with two different fluorescent dyes. Usually, the patient sample is labelled with a red fluorescent dye (Cy5) and the control sample is labelled with a green fluorescent dye (Cy3). Later, this step will allow to determine the amount of each DNA sample bound to each probe in the microarray.⁶

4. **Quantification of the labelled samples:** the two DNA samples will have to compete for a spot in the probes of the microarray. So, it is important to have equal amounts of labelled DNA samples. Different quantities of labelled DNA samples will cause an unbalanced competition. If one of the DNA samples has more DNA labelled it will lead to an advantage in the competition for a spot of the microarray and the significance of the differences between the patient sample and the control sample will be compromised.⁶

5. **Hybridization:** the two samples are dropped in the microarray on an oven at 67 °C. This process breaks the double strand of the DNA samples into single strands. The single strands compete to a probe spot which has complementary oligonucleotides. The duration of this step depends on the microarray format.⁷

6. **Washing:** microarray is washed to remove the DNA which was not bound to any probe.^{5,6}

7. **Scanning and image processing:** microarray is scanned into image files using a microarray scanner, and the fluorescence intensities of red and green bound to each probe are measured (Figure 3). The resulting ratio of fluorescence intensities is proportional to the ratio of the copy numbers of DNA sequences in the patient and control genomes, which is converted to a \log_2 ratio and displayed for CNV analysis.²

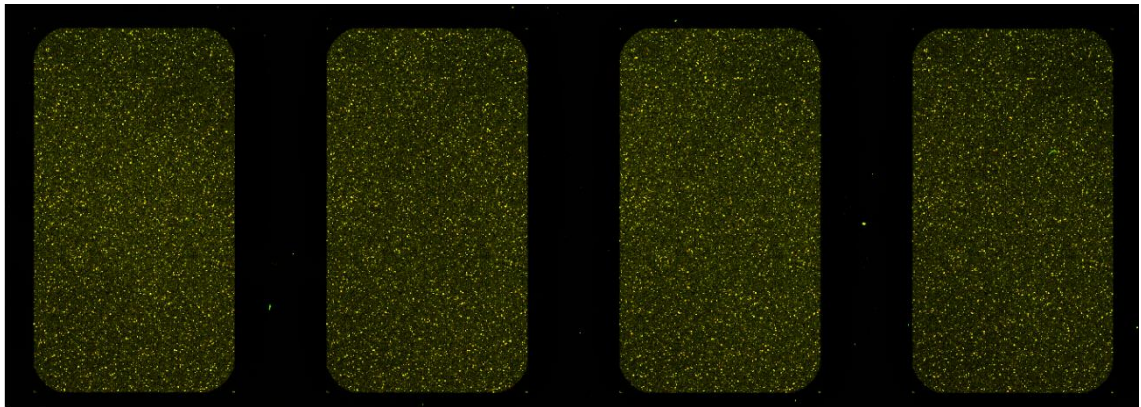


Figure 3 - Fluorescence intensities of four microarrays experiments. Figure from: Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra.

- 8. Quality control test:** this step is used to assess the relative data quality from a set of microarray by performing several tests, such as: verify the alignment of the probes positions, the intensity levels of green and red signals, the spread of red and green on the microarray and the noise level. In some cases, they can indicate potential processing errors that have occurred or suggest that the data from particular microarrays might be compromised (Figure 4).⁷

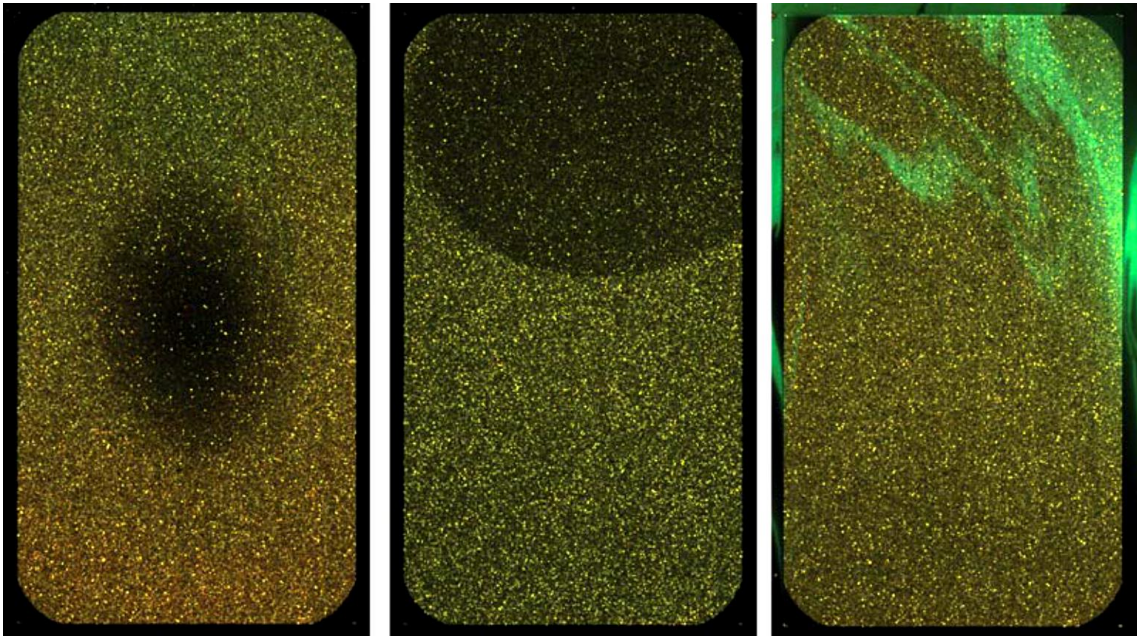


Figure 4 - Scanned images of microarrays with quality defects displayed. On the left and middle, defects resulting from large air bubbles during the hybridization process are shown. Poor hybridization is visible as the darker regions, which show reduced signal intensities of the underlying targets. On the right, typical defects introduced on washing (seen as green fluorescence) are shown. Figure from: Vermeesch, J., et al. (2012). *Genome-Wide Arrays: Quality Criteria and Platforms to be Used in Routine Diagnostics*. WILEY PERIODICALS, INC. DOI: 10.1002/humu.22076

The last step, the clinical interpretation of an arrayCGH result, requires a more detailed explanation for the comprehension of this work and is described on the next section.

2.3. The interpretation of an arrayCGH result

The interpretation of an arrayCGH result needs an understanding of the base-2 logarithmic values obtained in the fluorescence analysis of the microarray probes. Each logarithmic value corresponds to a probe and, each probe represents a specific region of the human genome where one fraction of patient DNA and one fraction of control DNA are bound. By the inverse of the logarithmic the proportions of each DNA sample can be estimated^{5,8}:

$$\log_2(\text{ratio}) = y \quad (1.1)$$

$$\text{ratio} = 2^y,$$

For a given probe, the y is one logarithmic value in the \log_2 scale, obtained from image processing, and the ratio is the proportion of each DNA sample present in the probe.

For probes where y is smaller than zero there is a loss of genetic material. In these cases, less quantity of patient DNA sample was present in the probes when comparing to the control DNA sample. A gain of genetic material occurs when the y is greater than zero, which means that exists more quantity of patient DNA sample than the control DNA sample bound to the probe. If y is zero, there is no genomic differences for that region and the same quantity of each DNA sample were hybridized in the probe.^{5,8}

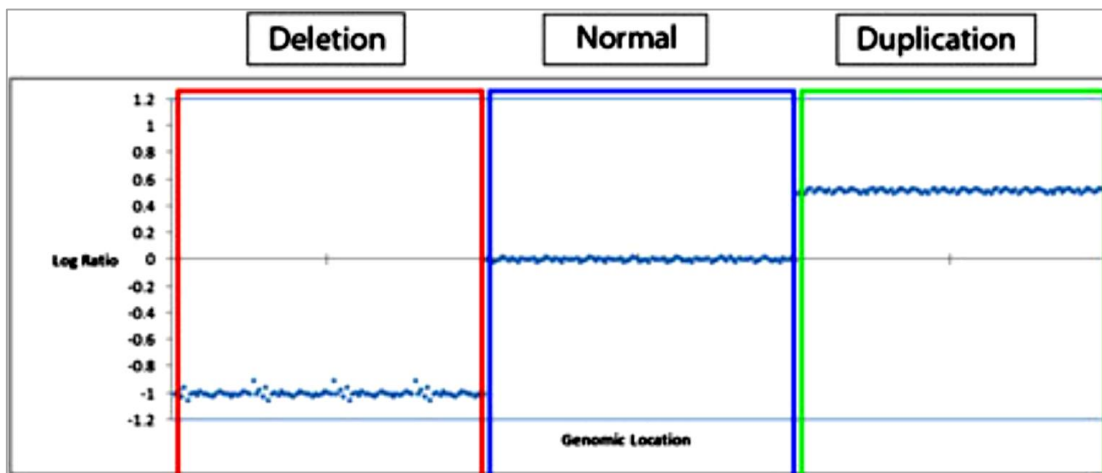


Figure 5 – Genomic imbalances via \log_2 ratio. The figure displays a typical plot of array CGH data, with probes plotted by genomic position on the x-axis and by the normalised $\log_2 R$ signal intensity ratios on the y-axis. Figure adapted from: Brady, P. and Vermeesch, J. (2012) *Genomic microarrays: a technology overview*. John Wiley & Sons, Ltd. DOI: 10.1002/pd.2933

For clinical reports only cases with a minimum of three consecutive probes with gains or three consecutive probes with losses are reported. Here, the proportions of patient and control DNA samples are calculated by equation (1.1) but y is the mean of the logarithmic values of the probes which reported the alteration.

A reported alteration is known as a copy number variation (CNV), which corresponds to a region of the genome, where alterations may range from about one kilobase to several megabases, that have been deleted or amplified on a certain chromosome.^{4,5} A set of CNVs is the result of an arrayCGH and the clinic interpretation is an interpretation of all CNVs in the set. There are two main steps when interpreting a CNV:

- The first is to analyse the “CNV *ratio*”, meaning if the CNV is a deletion or an amplification in a certain chromosome.
- Then, it is necessary to determine if the CNV has an impact on the patient life, in other words, if the CNV is benign or pathogenic.

An example of a possible arrayCGH result and the first step in its interpretation is shown below.

Table II - Example of an arrayCGH result. Each line is a CNV and each CNV is characterized by the chromosome where the variation appears, a cytoband which is a pattern capable of identifying a location in a chromosome, a start and a stop reference, a size of the alteration, the number of probes which reported the alterations and a $\log_2(\text{ratio})$, a mean of the all $\log_2(\text{ratio})$ registered by all the probes involved in a certain CNV.

Chromosome	Cytoband	Start (Kb)	Stop (Kb)	Size (Kb)	Probes	$\log_2(\text{ratio})$
5	p15.33	722994	820565	97572	4	-1.1005
6	p25.3	255150	362431	107282	9	0.5587

Determine a “CNV ratio” only requires the application of the \log_2 inverse (1.1):

$$\text{Chromosome 5: ratio} = 2^{-1.1005} \approx 0,5 = \frac{1}{2}$$

$$\text{Chromosome 6: ratio} = 2^{0.5587} \approx 1,5 = \frac{3}{2}$$

The first CNV of the Table II has a *ratio* of $1/2$ which means the control DNA sample has 2 copies of that region for each copy of the patient DNA sample. One portion of this region is missing on the patient DNA so the patient has one deletion of that region when compared with the control DNA sample. The second CNV of the Table II has a ratio of $3/2$. Here the patient DNA sample has three copies of that region but the control only has two copies. In this case the patient shows one duplication of that region.

The first step of the interpretation of an arrayCGH result is accomplished but what is the impact of the regions that are deleted or amplified in the patient life? Are they really pathogenic or are they harmless?

2.4. The importance of databases in the interpretation of a CNV

After determining a “CNV ratio”, the clinicians need to decide about the pathogenicity of a CNV. On Table III the main criteria of how to assess the pathogenicity of a CNV are presented.

Table III - Assessment of Pathogenicity of a CNV. Table from: Miller, D., et al. (2010). Consensus Statement: Chromosomal Microarray is a First-Tier Clinical Diagnostic Test for Individuals with Development Disabilities or Congenital Anomalies. *The American Journal of Human Genetics*. ISSN: 86, 749-764.

Primary Criteria	Pathogenic	Benign
1a. Identical CNV inherited from a healthy parent ^a		X
1b. Expanded or altered CNV inherited from a parent	X	
1c. Identical CNV inherited from an affected parent	X	
2a. Similar to a CNV in a healthy relative		X
2b. Similar to a CNV in an affected relative	X	
3. CNV is completely contained within genomic imbalance defined by a high-resolution technology in a CNV database of healthy individuals		X
4. CNV overlaps a genomic imbalance defined by a high-resolution technology in a CNV database for patients with development delay or congenital anomalies	X	
5. CNV overlaps genomic coordinates for a known genomic-imbalance syndrome (i.e., previously published or well-recognized deletion or duplication syndrome)	X	
6. CNV contains morbid OMIM genes ^b	X	
7a. CNV is gene rich	X	
7b. CNV is gene poor		X
General Findings ^c		
1a. CNV is a deletion	X	
1b. CNV is a homozygous deletion	X	
2a. CNV is a duplication (no known dosage-sensitive genes)		X
2b. CNV is an amplification (greater than 1 copy gain)	X	
3. CNV is devoid of known regulatory elements		X

Note: ^aAn inherited deletion from an unaffected parent could harbour an OMIM (Online Mendelian Inheritance in Man, an online catalogue of Human Genes and Genetic Disorders) morbid gene that is recessive and could be pathogenic in conjugation with a point mutation on the trans allele inherited from other parent. ^bCNV should produce the same type of mutation that is known to cause the OMIM disease and the phenotype produced should be that expected for the OMIM disease. ^cExceptions to each case have been seen

As shown in Table III, the majority of criteria are directly or indirectly related with a query of previously recorded CNVs, and this query is often performed on data contained in a database. When clinicians make a query for a certain CNV and find a large number of matches, the CNV assessment of pathogenicity will be more reliable and more accurate.¹

Sometimes, clinicians get a CNV matching result with a poor number of matches or even none when querying databases about a certain CNV. This situation is always a challenge for clinicians because they have to interpret variants of uncertain clinical significance (VOUS). In this situation, the assessment of pathogenicity sometimes is not possible and when it is possible it will be a less reliable and less accurate decision than one with a large number of matches. This problem can be addressed through an increased data sharing among laboratories and clinicians, and a database will be a good solution for registering and sharing information.¹

2.4.1. Interpretation of CNV at *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra*

Table IV describes the CNV classification used at *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra*.

Table IV – CNV classification used at Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra.

Class	Description
I	Deletion or duplication in a region associated with a microdeletion or microduplication syndrome.
II	Deletions or duplications that are not described in the DGV until to date and involve known coding genes.
IIIA	Deletions or duplications that, until to date, are reported in low frequency in the DGV or are not fully overlapping with those described in DGV.
IIIB	Deletions or duplications that, until to date, are reported in low frequency in the DGV or are not fully overlapping with those described in DGV, not involving known coding genes.
IV	Deletions or duplications reported in normal subjects in DGV.

DGV - Database of Genomic Variants (<http://projects.tcag.ca/variation/>)

2.5. Management of arrayCGH results at *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra*

The arrayCGH technique is performed at *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra* since 2011. From that year until the beginning of 2015, the clinicians of this laboratory saved the arrayCGH results of their patients in spreadsheet files for to help in future clinical diagnosis of new patients. Until the beginning of 2015, they kept a record of about 4000 arrayCGH results.

However, as the number of arrayCGH results increased, the clinicians start to face added difficulties to find the results they needed to help in the decision about the

pathogenicity of a CNV in recent arrayCGH results. This situation prompted them to look into a more efficient way to save and query their results.

As a solution, the clinicians chose to develop an in-house relational Database (DB) to store their results and an application which allows them to feed and query the DB and to share their results with others. The next sections are the report of the development of this solution.

3. Software Development

3.1 Software Requirements

The first step of a software development starts with a request from the researchers. Normally it is a small statement where some global goals are pointed out⁹. The original request for the software was:

- Develop and implement a relational database which should contain arrayCGH results from patients with cognitive deficit, and also to allow a set of queries defined by the laboratory clinicians in a way that can answer to questions or hypotheses related with new arrayCGH results.

This information needs to be analysed in order to break out some requirements from what the researchers are asking for. A requirement is a single task that the software has to do⁹. Figure 6 is a set of requirements which can be written with the information from the first request.

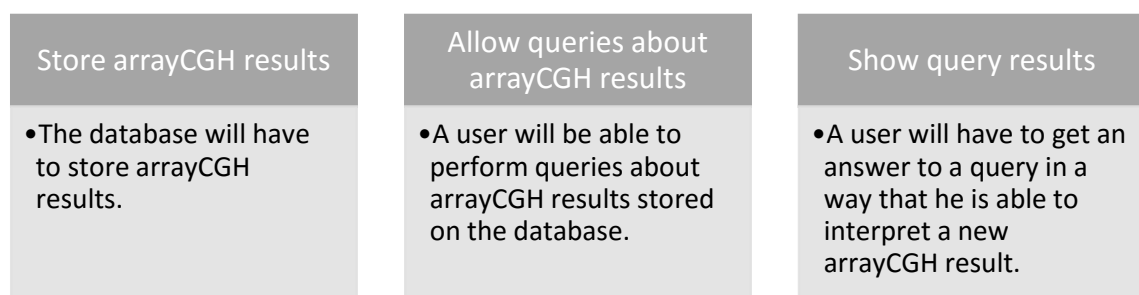


Figure 6 – Requirements extracted from the researchers first request.

These requirements give some information about some software functionalities. However, only with this information, there are a lot of gaps in the understanding of what the software is supposed to do. It is necessary to talk with the researchers in order to

get more details about the requirements and also to find out about additional requirements the researchers did not express in the first request. Before start coding a software, it is necessary for the researchers to think about everything they want the software to realise. There are a lot of techniques which can help accomplish that goal such as brainstorming, role playing and observation. These techniques were used several times, through an iterative process, in order to better shape the requirements to the researchers ideas and to improve or add new functionalities that they were not conscious about in the first request of the project. In the end of requirements gathering, they must be researcher-oriented. In other words, a requirement should be written from the researcher's perspective, describing what the software is going to do for them. This is known as user stories. Both the programmer and the researchers should understand the meaning of the user stories. In that way, both will be capable to ask about any requirement and communicate with each other at any time of the software development.⁹ A set of researcher-oriented requirements developed for this work are presented next and they are grouped by general, researcher, administrator and super administrator requirements.

3.1.1. General requirements

<p>Add a new user</p> <ul style="list-style-type: none"> •A new user will be able to sign up. 	<p>Log in</p> <ul style="list-style-type: none"> •An approved user will be able to log in. 	<p>Log out</p> <ul style="list-style-type: none"> •A logged in user will be able to log out. 	<p>Look for a CNV match</p> <ul style="list-style-type: none"> •A user will be able to search for a CNV match by giving a localization or by specifying a gene name.
<p>Show CNV match</p> <ul style="list-style-type: none"> •A user will be able to see a bar graph which represents the CNVs that match his search. 	<p>View a clinic case of a CNV</p> <ul style="list-style-type: none"> •A user will be able to view any clinic case of a CNV that appeared on a CNV match. 	<p>Look for a CNV match in other DB</p> <ul style="list-style-type: none"> •A user will be able to look for a CNV match in the DB of UCSC, of any CNV in a recorded arrayCGH result. 	<p>Look for a gene information</p> <ul style="list-style-type: none"> •A user will be able to look for information about any gene contained on a arrayCGH result in the DB of GeneCards.

<p>Allow different kind of users</p> <ul style="list-style-type: none"> • A user should be one of those kinds: Researcher, Administrator or Super Administrator 	<p>Show a User Interface</p> <ul style="list-style-type: none"> • Show a UI which allows users to access the software functionalities according to their kind.
--	---

(Legend: **DB** – Database; **UCSC** – University of Carolina, Santa Cruz, genome browser at <https://genome.ucsc.edu/>; **GeneCards** – Human Gene Database at <http://www.genecards.org/>; **UI** – User Interface)

3.1.2. Researcher requirements

<p>Add a new clinic case</p> <ul style="list-style-type: none"> • A researcher will be able to upload a spreadsheet with the information about an arrayCGH result. 	<p>Show clinic cases</p> <ul style="list-style-type: none"> • A researcher will be able to see all arrayCGH results added by him. 	<p>Edit a clinic case</p> <ul style="list-style-type: none"> • A researcher will be able to update any information about any arrayCGH result added by him. 	<p>Remove a clinic case</p> <ul style="list-style-type: none"> • A researcher will be able to remove any clinic case added by him.
<p>View a Clinic Case</p> <ul style="list-style-type: none"> • A user will be able to see any arrayCGH result added by him. 			

3.1.3. Administrator requirements

<p>Show users</p> <ul style="list-style-type: none"> • An administrator will be able to see all information about all researchers from his laboratory. 	<p>Change a user state</p> <ul style="list-style-type: none"> • An administrator will be able to block or approve the access to the software of any researcher from his laboratory. 	<p>Remove a user</p> <ul style="list-style-type: none"> • An administrator will be able to remove a researcher from his laboratory. 	<p>Search for a user</p> <ul style="list-style-type: none"> • An administrator will be able to search for a researcher from his laboratory by the researcher name.
---	--	--	---

<p>Show clinic cases</p> <ul style="list-style-type: none"> •An administrator will be able to see the information about all arrayCGH results linked to his laboratory. 	<p>Change a clinic case state</p> <ul style="list-style-type: none"> •An administrator will be able to change the visibility state of a clinic case linked to his laboratory. 	<p>View a clinic case</p> <ul style="list-style-type: none"> •An administrator will be able to see all information of any clinic case linked to his laboratory. 	<p>Search a clinic case</p> <ul style="list-style-type: none"> •An administrator will be able to search for any clinic case linked to his laboratory by its name.
<p>Remove a clinic case</p> <ul style="list-style-type: none"> •An administrator will be able to remove any clinic case linked to his laboratory 			

3.1.4. Super Administrator Requirements

<p>Show laboratories</p> <ul style="list-style-type: none"> •A Super Administrator will be able to see the information about all laboratories. 	<p>Edit laboratory</p> <ul style="list-style-type: none"> •A Super Administrator will be able to update all information about a laboratory. 	<p>Add a new laboratory</p> <ul style="list-style-type: none"> •A Super Administrator will be able to add a new laboratory to the DB. 	<p>Remove a laboratory</p> <ul style="list-style-type: none"> •A Super Administrator will be able to remove a laboratory.
<p>Search for a laboratory</p> <ul style="list-style-type: none"> •A Super Administrator will be able to search for a laboratory by its laboratory name. 	<p>Show users</p> <ul style="list-style-type: none"> •A Super Administrator will be able to see all information about all researchers and administrators. 	<p>Change a user state</p> <ul style="list-style-type: none"> •A Super Administrator will be able to block or approve the access to the software of any researcher or administrator. 	<p>Remove a user</p> <ul style="list-style-type: none"> •A Super Administrator will be able to remove a researcher or an administrator.
<p>Search for a user</p> <ul style="list-style-type: none"> •A Super Administrator will be able to search for a researcher or an administrator by his name. 	<p>Show clinic cases</p> <ul style="list-style-type: none"> •A Super Administrator will be able to see the information about all arrayCGH results. 	<p>Change a clinic case state</p> <ul style="list-style-type: none"> •A Super Administrator will be able to change the visibility state of any clinic case. 	<p>View a clinic case</p> <ul style="list-style-type: none"> •A Super Administrator will be able to see all information of any clinic case.

Remove a clinic case

- A Super Administrator will be able to remove any clinic case.

Search a clinic case

- A Super Administrator will be able to search for any clinic case by its name.

3.2. Software Design

After knowing the software requirements the design process begins, which will help a programmer to plan and structure the researcher needs and, later on, in software coding and implementation. Three design processes were used in the development of this work: a Use Case Diagram (UCD), an Entity-Relationship Model (ERM) and a Relational Model (RM). The first one allows to set the system dynamic behaviour. The second diagram defines the conceptual view of a DB. The RM is a mapped relational schema based on the ERM and is a key diagram when setting up a DB.

3.2.1. Use Case Diagrams (UCD)

To model a system it is important to know its dynamic behaviour, the evolution of the system when it is running. A dynamic behaviour is made of interactions between the system and the users or other external systems. UCD can help to model these interactions and to map the system requirements.¹⁰

A UCD is made of actors, use cases and their relationships, in order to model a system. Normally, an actor represents a human user, a use case represents some functionality of the system and a relationship is an action between actors and use cases.¹⁰

So, to model a system, several UCD are used. The Figures 7, 8 and 9 represent simplified UCD of Researcher, Administrator and Super Administrator in this work.

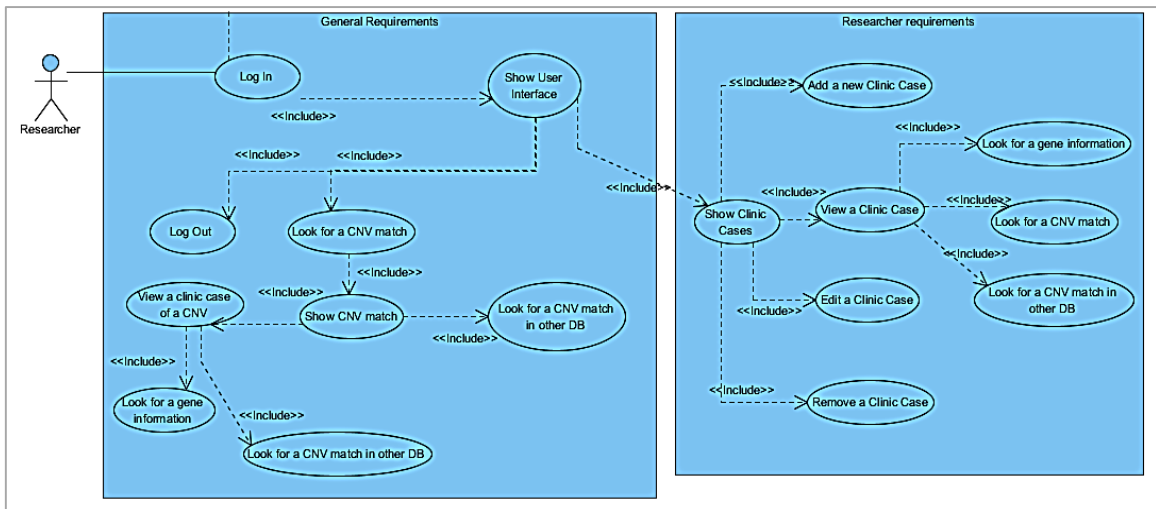


Figure 7 – Use Case Diagram of a Researcher.

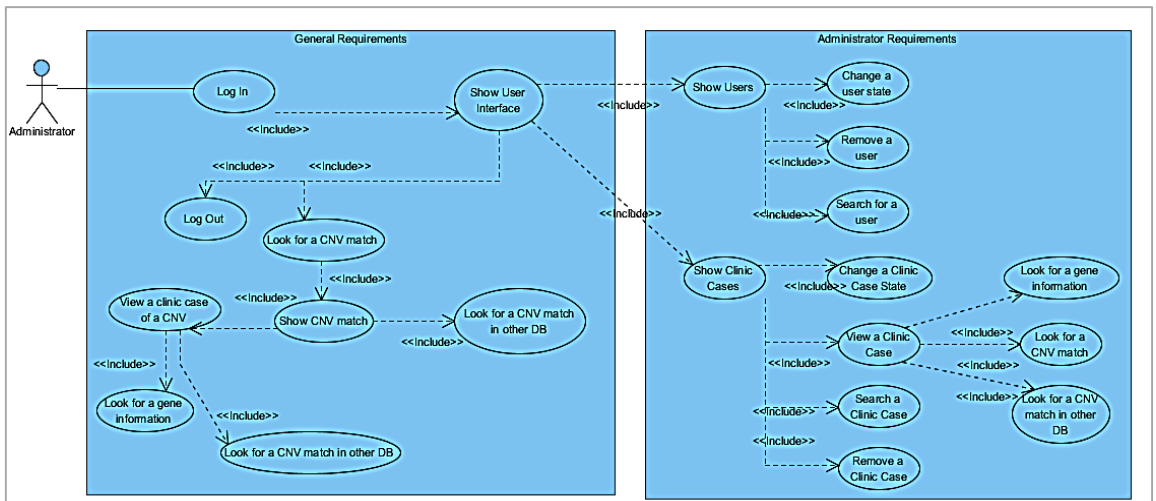


Figure 8 – Use Case Diagram of an Administrator.

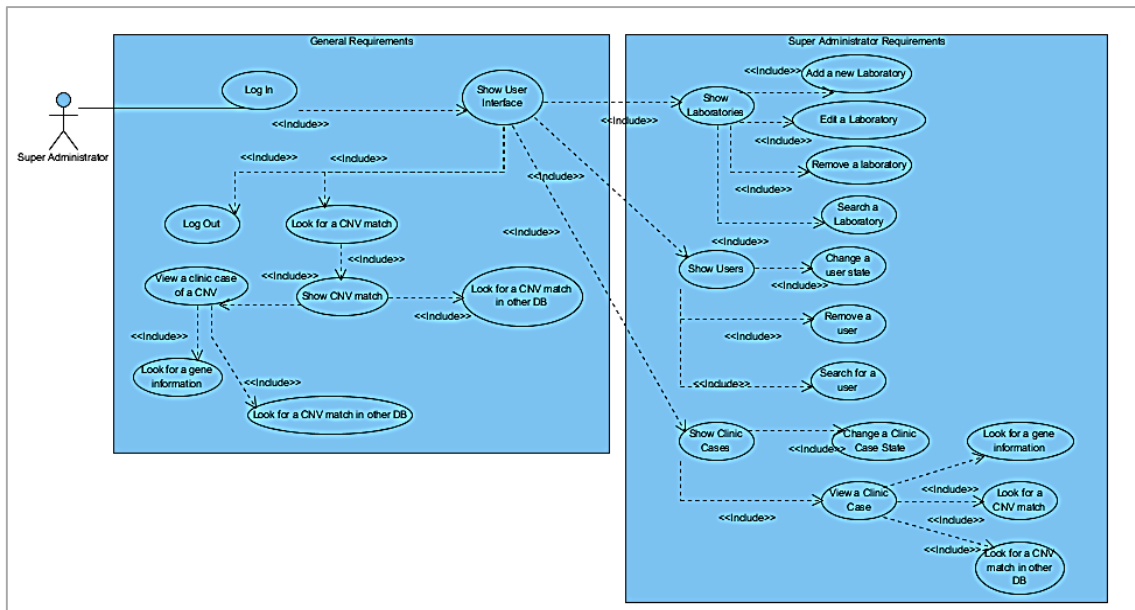


Figure 9 – Use Case Diagram of a Super Administrator.

Notes about UCD:

A use case is a situation where a system is used to fulfil one or more of the researcher-oriented requirements and it captures a piece of functionality that the system provides. Use cases describe the system’s requirements strictly from the outside and they specify the value/service that the system delivers to the users.

After capturing an initial set of actors that interact with the system it is possible to start assembling relationships between the actors and the requirements. The purpose of use case relationships is to provide the designers of the system with some architectural guidance in order to help them break down the system requirements into manageable pieces.

Figures 7, 8 and 9 only show relationships of the include type. The include relationship declares that the use case at the head of the dotted arrow completely follows all the steps from the use case being included. The use cases have the name of the requirements and they are specified in the Software Requirements, which could help with the UCD interpretation. The use case for the requirement “Add a new user” is not represented by a UCD, but essentially, a new user will sign up by filling a form with some

basic information and by choosing a role (Researcher or Administrator), a laboratory and a username and a password in order to log in after being approved.¹⁰

3.2.2. Entity-Relationship Model (ERM) and Relational Model (RM)

An ERM defines the conceptual view of a DB and usually is made of entities, attributes and relationships. An entity is a real world object which is defined by a set of properties, the attributes. Attributes store some kind of information about an entity. Between entities occur relationships, which are a description about an interaction between two entities. Figure 10 shows a representation of the ERM model developed for this work.¹¹

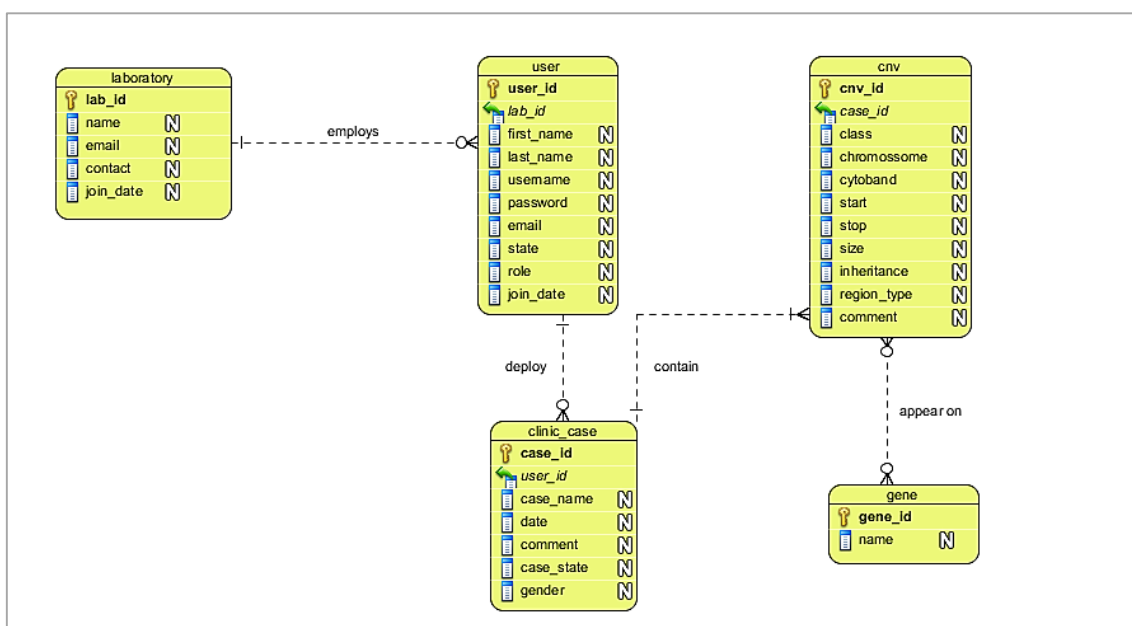


Figure 10 - Entity-Relationship Model.

Notes about ERM:

The ERM has five entities: “laboratory”, “user”, “clinic case”, “cnv” and “gene”, each one with their attributes. But what is the difference between entities and attributes

and why they are linked like that? It is not easy to distinguish their roles in database modelling, but, briefly, entities should contain descriptive information and attributes should be linked to the entities they most directly describe. If there is descriptive information about a data element, the data element should be classified as an entity. If a data element requires only an identifier and does not have relationships, it should be classified as an attribute. With “user”, for example, there are some descriptive information such as “first name” and “email”, then “user” should be an entity. If the first name and the last name are needed to identify a user, then “first name” and “last name” should be classified as attributes associated with the “user” entity.

The relations between two entities can be described by its connectivity and its existence. Connectivity defines the relation degree between two entities. In other words, it defines how many records of a certain entity are allowed to relate with other entity records. The connectivity between two entities can be one of the following:^{11, 12}

- **One to one (1:1):** a record in entity A can have exactly one matching record on entity B, and a record in entity B only can have one matching record in entity A too.
- **One to many (1:N):** a record in entity A can have many matching records in entity B, but a record in entity B only can match one record in entity A.
- **Many to Many (N:N):** a record in entity A can have many matching records in entity B and a record in entity B can have many matching records in entity A, too.

Existence defines the obligation of a relationship of an entity record to another entity record, and it can be:

- **Mandatory:** If entity A has mandatory existence in entity B, all entity A records have to be references in records in entity B.
- **Optional:** If entity A has optional existence in entity B, there could exist records of entity A which have no references on entity B records.

ERM Relationships:

- **Laboratory - User:**

Connectivity 1:N: One laboratory record can have many user records. One user record only can match exactly one laboratory record.

Existence: One laboratory record can have no reference to a user record (optional). One user record always have to be one reference to a laboratory record (mandatory).

- **User – Clinic Case:**

Connectivity 1:N: One user record can have many clinic cases records. One clinic case record can have only one user record.

Existence: One user record can have no reference to a clinic case record (optional). One clinic case record always have to be one reference to a user record (mandatory).

- **Clinic Case - CNV:**

Connectivity 1:N: One clinic case record can have many CNV records. One CNV record can have only one clinic case record.

Existence: One clinic case record have to be always a reference to a CNV record (mandatory). One CNV record always have to be a reference to one clinic case record (mandatory).

- **CNV – Gene:**

Connectivity N:N: One CNV record can have many genes records. One gene record can have many CNV records.

Existence: One CNV record can have no reference to a gene record (optional). One gene record can have no reference to a CNV record (optional).

After setting up an ERM it is easier to develop an RM. ERM entities became RM tables, ERM entity's attributes become RM columns (fields) of tables with their respective data types. ERM relationships help at mapping interactions between the tables of an RM.¹¹

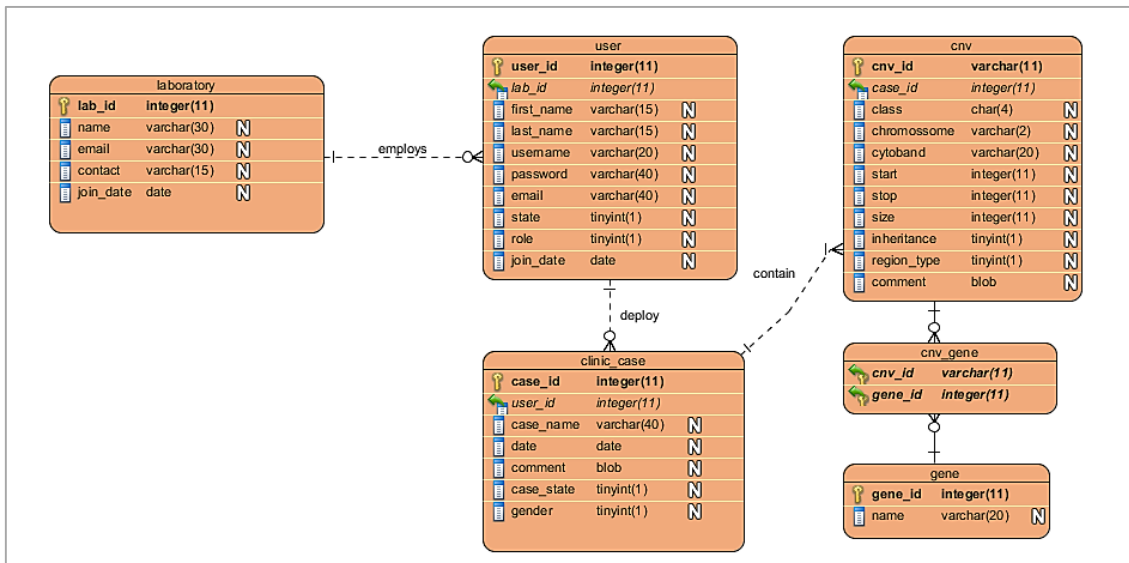


Figure 11 - Relational Model.

Notes about RM:

The connectivity and the existence describe the relation between two tables in the same way as in ERM, but those two concepts are not enough to model a database. An important thing to address when setting up a database is how the data contained in tables will be used. The goal is to use the columns in a way that makes getting the information out of it easy. It is simpler to accomplish that goal when tables are normalized, and databases with normal tables have huge benefits:¹³

- Normal tables won't have duplicate data, which will reduce the size of the database.
- With less data to search through, queries will be faster.

In order to be normal, a table has to follow some rules:

1. First Normal Form (1NF):

- **Each row of data must contain atomic data:** small pieces of data that cannot or should not be divided. In other words, a column with atomic data cannot have several values of the same data type in that column and a table with atomic data cannot have multiple columns with the same data type. Figures 12 and 13 show two examples where this rule is violated:

Table: CNV			
Chromosome	Start (pb)	Stop (pb)	Genes
2	12314	423544	ABCA12, CACNB4, DCTN1
X	33454	52343	EMD, DCX, FMR1

Figure 12 – Example: A table with non-atomic data where several values of same data type are allowed in genes column. This table makes queries about genes much harder.

Table: Laboratory			
Laboratory Name	user1	user2	user3
GeneticLab	Maria	João	Raquel
CitoLab	Sara	Rui	Pedro

Figure 13 – Example: A table with non-atomic data where multiple columns with same data type are allowed. This table makes queries about users from a laboratory more difficult.

- **Each row of data must have a unique identifier, known as a Primary Key (PK):** a primary key is a column of a table that makes each record unique. A PK cannot be NULL, a value must be given when the record is inserted, must be compact and its value cannot be changed. Figure 14 shows an example of a table that follows this rule:

Table: CNV			
ID	Chromosome	Start (pb)	Stop (pb)
1	2	12314	423544
2	X	33454	52343

Figure 14 – Example: A table where the ID is the PK.

When a table has non-atomic columns they need to be moved into new tables and the new tables have to be designed in order to obey to 1NF. If those tables have records that can be related to the records of the table from where they are moved (the parent table), they will need a reference to the parent table. This reference is known as Foreign Key (FK). A FK is a column in a table that references the PK of another table. A FK is also a table constraint because a FK can only take values that exist in the table where the key came from, the parent table, which is known by referential integrity. Figure 15 illustrates an example about the utility of a FK.

Table: User			
User ID (PK)	First Name	Last Name	Role
3	Ana	Correia	Administrator
4	Tiago	Amorim	Researcher
5	Isabel	Silva	Researcher

Table: Clinic Case		
Clinic Case ID (PK)	User ID (FK)	Case Name
1	5	D233M
2	4	D456M
3	5	D245F

Figure 15 – Example: PK and FK. The User ID on the Clinic Case table is a foreign key because it references a primary key from User table. This allows to link clinic cases to researchers from the user table.

On the ERM a many-to-many connectivity happens between “cnv” entity and “gene” entity. On ER model many-to-many connectivities cannot happen because it ends up with duplicate data on one of the linked tables which breaks the 1NF. There is no good reason to violate the 1NF, and many good reasons not to. One of the most important is that it takes a much larger time when querying tables with all the repeated data. Another one is to preserve data integrity on delete and update processes. So, to solve this issue, a junction table “cnv_gene” is created between the tables “cnv” and “gene”. A junction table holds a PK from each table, which allows to relate records between the tables.

3.3. Server: LAMP technology

Now as the software requirements and the software design are defined, it is necessary to set up a system capable of implementing what was projected, a system capable of analysing requests, storing information and delivering responses according to the requests. A system with these requirements is known as a Server. There are a lot of solutions to implement a server; in this work a server based on LAMP technology was the chosen option.¹⁴

LAMP, which stands for Linux, Apache, MySQL and PHP, is a robust set of software that works well as a system that has a proven track record of being efficient, secure and always on the leading edge of the Internet technology. Also, the LAMP solution has the advantage of being free and fast. Each component of LAMP exhibits benchmarks that far exceed those of their open source competitors. The Linux/Apache combination is capable of serving more pages to its users than other commercial or open source solutions. MySQL is the fastest open source database available and is utilized by several well-known companies (<http://www.mysql.com/customers/>). PHP is a very solid server-side scripting language.¹⁴

The major advantages of LAMP were already presented, but their elements need a brief explanation in order to understand their importance on LAMP technology. After this, the hardware and the software versions used to develop this work are presented.¹⁴

3.3.1. Linux Operating System

Linux is an operating system that runs applications. It is specifically noted for its speed, minimal hardware requirements, security and remote administration capability. It is a fully featured operating system that does not cost anything to use.¹⁴

The core of the Linux operating system (the kernel) is under GNU General Public License (GPL). The reason Linux is under GPL is simple: people are authorized to make modifications of the software, and in turn, release their versions to the public, as long

as they release the source code along with it; this allows other people to use and modify the work that others have done. GPL it's a "remember your roots" type of license, which allows this open source to continuously evolve.¹⁴

3.3.2. Apache Web Server

Apache is an open source web server solution packed with features, extremely fast, and works well with the Linux operating systems. With the Apache Web server it is possible to create virtual hosts that allow to run multiple web sites on a single server.¹⁴

Apache is a well-known web server solution and is currently the most used number web server system worldwide according to Netcraft surveys (<http://news.netcraft.com/archives/category/web-server-survey.html>).

3.3.3. MySQL Database Server

MySQL is a powerful, robust database manager that enables to store and retrieve data with a script language, such as PHP. Using a database is important for creating dynamic sites, being able to use a single page of code to deliver different information based on a user's interaction. This would be impossible without the use of a database and a scripting language, such as PHP, to manipulate data.¹⁴

3.3.4. PHP Scripting Language

PHP is a recursive acronym that stands for: Hypertext Preprocessor. It is used for Web Development and can be embedded into HTML. It is a simple scripting language that enhances a website by allowing interactions with the users.¹⁴

3.4. Materials

3.4.1. Software versions and hardware

In this work, the following software versions have been used:

- Ubuntu 14.04 LTS (Ubuntu is an operating system based on Linux)
- Apache 2.4.7
- MySQL 5.5.41
- PHP 5.5.9

The hardware has the following specifications:

- Processor: 2x Intel(R) Pentium(R) 4 CPU 3.20GHz
- Memory: 1024MB
- Hard Disk: ATA ST3808110AS (80 GB)

3.4.2. Characterization of arrayCGH results provided for the development of arrayCGH DB

In order to develop this project, the researchers of *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra* provided a set of arrayCGH results, which they considered representative for the majority of arrayCGH results they have stored on spreadsheets since 2011. The set given, contained results from 2013, 2014 and 2015, and has two types of formats: “Tabela Classes”, a handmade report which is delivered to clinicians, and “IntervalBasedReport”, a report exported by CytoGenomics, a software for CNV analyses. On Table V are characterized the data contained in arrayCHG results for those two format types.

Table V – Characterization of arrayCGH results for “IntervalBasedReport” and “Tabela Classes” formats.

	Case Name	Genre	Chromosome	Cytoband	Start	Stop	Size	Type	Classification	Gene Names
Interval Based Report	Data Type	String	String	String	Integer	Integer	Integer	String	String	String
	Values or Description	Case Name and Genre separated by underscores	chr1-22,X or Y	Cytoband expression	Maximum 11 digits	Maximum 11 digits	Maximum 11 digits	Not provided or Duplicação, Deleção, Amplificação or Variação if added by Researchers	Classe I, II, IIIA, IIIB or IV	One or more gene names separated by commas or empty
	Example	S123_14M_Female	chr4	q11.1-q11.21	13199589	13992363	792775	Duplicação	Classe IV	ZNF705D, FAM66D,
Tabela Classes	Data Type	String	String	String	Integer	Integer	Integer	String	String	Not provided
	Values or Description	Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra format	Cytoband with chromosome	Cytoband with chromosome	Maximum 11 digits	Maximum 11 digits	Maximum 11 digits	Duplicação, Deleção, Amplificação or Variação	I, II, III or IV	
	Example	S123/14M	14q21.1	14q21.1	13199589	13992363	792775	Deleção	II	

4. Security

All data is stored on a relational database and the accesses are regulated by user-level permissions. Users are allowed to share and manage their own data with different users/laboratories. In order to guarantee data integrity and data security, all options available for a registered and approved user are only accessible after log in. Clinic Case anonymity is ensured as there is no storage of personal information, such as patient names. Clinic Cases identification relies on laboratory Clinic Case identification.

4.1. User accesses

Any given Researcher is only able to access the data stored after an approved registration by a high level user, a Super Administrator or an Administrator from the same laboratory as the Researcher. A registration of an Administrator only can be approved by a Super Administrator. After that, the access can be denied at any time, a Super Administrator can block any user, and an Administrator can only block their laboratory Researchers.

For the user passwords is used a MySQL function that provides a one way encryption. This means that it is not possible to decrypt them and it adds an additional degree of security for this type of data. However, this does not mean that it is not possible to discover the real passwords through brute force attacks, but it makes it more difficult.

A Super Administrator is able to remove any low level user. An Administrator can only remove Researchers from its own laboratory.

When a user asks for some content on the server, an authorization process verifies if the user has permission to access the content he is asking for, thanks to *session* and *cookies* variables set up on user log in.

Log Out will clear the *session and cookies* variables.

4.2. Clinic Case management and access

An approved Researcher is able to upload Clinic Cases that report CNVs. All the data upload process is verified, and wrong Clinic Case formats or unexpected data are rejected, thanks to a validation process.

All uploaded Clinic Cases have their state set to hidden, which means that only users from the same laboratory as the Researcher that had made the upload can access the uploaded information. A Researcher can edit some information of Clinic Cases linked to him, such as Clinic Case comments, Clinic Case Gender, CNV class, CNV chromosome, CNV genes, CNV inheritance, etc., in order to provide more details about the case or to correct some information that was wrong. A Researcher can also remove their Clinic Cases.

Administrators can change the state, to hidden or to public, and remove Clinic Cases which are linked to their laboratory.

Super-Administrators can change the state, to hidden or to public, remove and view all Clinic Cases.

4.3. Data Removal

A Super Administrator is able to remove Laboratories and Users. Administrators are able to remove Researchers and Clinic Cases from their Laboratories. Researchers can only remove their Clinic Cases.

Deletes are on cascade, which means that if a Super Administrator decides to remove a laboratory he will remove all the data linked to the laboratory, such as Users, Clinic Cases and CNVs. This is crucial to safeguard the data referential integrity and will prevent errors related to this subject. This is an ultimate option to guarantee the database robustness. Before proceeding to a data removal, alternatives like blocking a

user or hiding a Clinic Case should be considered. Anyway, before a data removal the user will be informed of what he will remove and if he is sure about that, he will be able to confirm the process and the data will be removed.

4.4. SQL injection

Some precautions were taken in order to avoid SQL injection. Data sent by users are not used directly when building queries. Instead, data validation processes are used through PHP functions and PHP MySQL functions in order to inhibit this kind of database attacks.¹⁵

4.5. Data Backup

The solution developed relies on MySQL and stores its data in a MySQL database, so a database backup will allow restore all information stored in case of any disaster, such as manual mistake, software errors, hardware errors, server compromise, *etc.* It was used *mysqldump*, a command to dump the database (dev.mysql.com/doc/refman/5.1/en/mysqldump.html) written on a file bash script which is ran every day through cron, a system daemon used to execute desired tasks (in the background) at designated times (help.ubuntu.com/community/CronHowto).

4.6. Ubuntu Security

Ubuntu is an open source software that is potentially vulnerable until patches and updates become available. The Ubuntu security can be secured by a set of measures, such as updating the system and by configuring the firewall.¹⁶

Staying up to date is one of the ways to keep a system secure. There are new patches and fixes for Ubuntu frequently. The open source community is constantly trying to find exploits and bugs in its software so that they can release patches and make

the environment as secure as possible.¹⁶ There are some commands to effectively update the Ubuntu system:

1. *apt-get update*, used to re-synchronize the package index files from their sources. The indexes of available packages are fetched from the location(s) specified in */etc/apt/sources.list*. It does not install new versions of software. An update should be always performed before an upgrade.
2. *apt-get upgrade*, used to install the newest versions of all packages currently installed on the system from the sources enumerated in */etc/apt/sources.list*. Packages currently installed with new versions available are retrieved and upgraded; under no circumstances are currently installed packages removed, nor are packages that are not already installed retrieved and installed. New versions of currently installed packages that cannot be upgraded without changing the install status of another package will be left at their current version.
3. *apt-get dist-upgrade*, in addition to performing the function of upgrade, this option also intelligently handles changing dependencies with new versions of packages; apt-get has a "smart" conflict resolution system, and it will attempt to upgrade the most important packages at the expense of less important ones, if necessary. The */etc/apt/sources.list* file contains a list of locations from which to retrieve desired package files.

Configuration of the firewall is another step that should be done to maintain the system security. This system belongs to a set of systems with a shared firewall set up by the informatics maintenance of the *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra*. Because of this, no additional security was added on the system firewall. However, if the system does not have a shared firewall, *iptables* should be used in order to optimize the Ubuntu firewall. *iptables* is a command-line packet filter utility that uses policy chains to allow or block traffic. When a system tries a connection with an Ubuntu system, *iptables* looks for a rule in its list to match it to. If it does not find one specific rule for that connection, the default rule is used. With

iptables it is possible, for example, to reject the communication of a certain IP with an Ubuntu system.¹⁶

5. Results and Discussion

In this section, a set of screenshots about the developed solution is shown together with a brief discussion of each one. At the end, a set of graphics is presented that will give a brief overview of the CNVs information already existing in the database.

5.1. Log in and Sign in

When a user accesses the Array CGH DB, the Home page of the application (Figure 16) is shown. There he will be able to sign up if he is a new user or to log in if he is already registered. When a user select “Sign up”, a form is displayed which asks for some user information, such as first name, last name and an email, a username and a password which will be used to log in the user, a role (Researcher or Administrator) and the laboratory where the user is affiliated (Figure 17).



Figure 16 – Home page of Array CGH DB. A user can choose: Log in, if he is already registered, or Sign up, for a new user registration.

Array CGH DB - Sign up - Mozilla Firefox

Array CGH DB - Sign up

localhost/project/signup.php

Array CGH DB - Sign up

Home Log in Sign up

Please enter all the information to sign up to Array CGH DB

User Information

First Name:

Last Name:

Username:

Password:

Password(retype):

Email:

Laboratory:

Role:

footer

Figure 17 – Sign up page of Array CGH DB. In order to “Sign up” a user has to fill some information about himself, choose a username, a password, a role and the affiliation laboratory.

After “Sign up”, a new user has to be confirmed by a Super-Administrator or by an Administrator from the same laboratory in order to be able to log in (Figure 18). After being confirmed, when a user selects “Log in”, he is asked about a username and a password. For a successful “Log in”, these two have to be recognized (the same as chosen when he signed up). (Figure 19)

When a user “Logs in”, *Session* and *Cookies* variables are set. This process allows to control and make sure that users will only get access to functionalities and information according to their identification (Role, Laboratory, User ID).

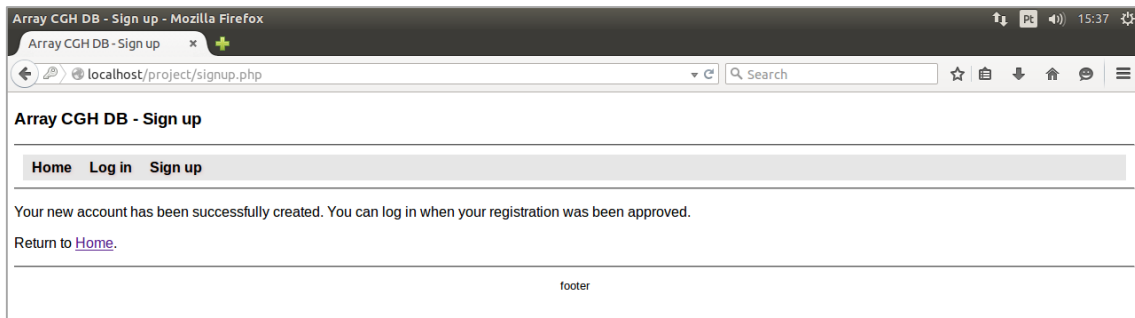


Figure 18 – Successful “Sing up” message.

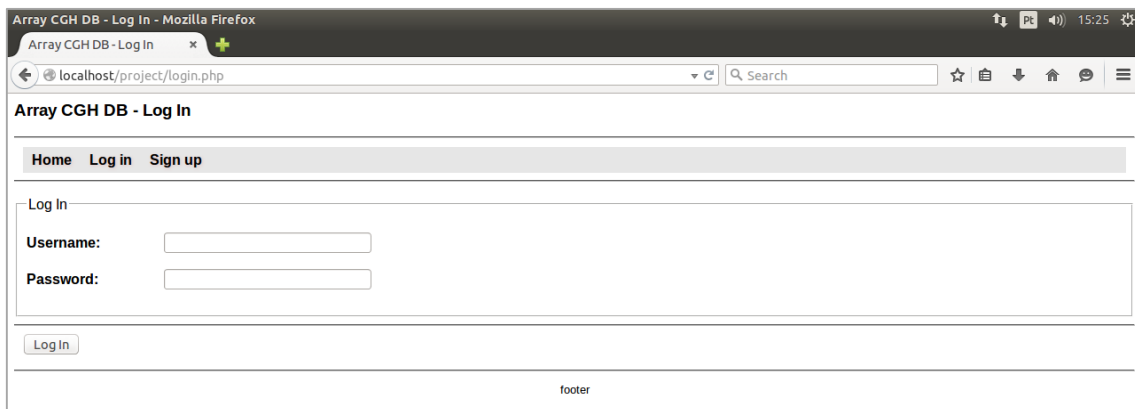


Figure 19 – Log in page of Array CGH DB. In order to “Log in”, a user has to insert a username and a password that match.

5.2. Users Navigation Bar

After “Log in”, a user will be redirected to the Home page. There he will see a Navigation Bar which allows him to access several functionalities according to his role. A Researcher is able to select “Clinic Case” to add new Clinic Cases or to manage Clinic Cases, or “CNV” to perform a CNV matching (Figure 20). An Administrator can select “User” to manage users, “Clinic Case” to manage Clinic Cases and “CNV” to perform a CNV matching (Figure 21). A Super-Administrator is able to select “Laboratory” to add new laboratories, “User” to manage users, “Clinic Case” to manage clinic cases and “CNV” to execute a CNV matching (Figure 22).

Although some Navigation Bar options have the same name for different roles its functionally can be different for each role. For Researchers and Administrators,

“CNV” allows a CNV matching with CNVs with public or hidden state linked to their laboratory, and with CNVs with public state linked to a different laboratory. For Super-Administrator “CNV” permits a CNV matching with all CNVs recorded on database. For “User”, a Super-Administrator is able to approve or block the access to the software of any Researcher or Administrator registered on the database, but an Administrator is only authorized to approve or block the access of Researchers from the same laboratory as his. When accessing “Clinic Case”, a Super-Administrator can manage all clinic cases recorded on database, an Administrator is able to perform modifications on clinic cases linked to his laboratory and a Researcher can only manage clinic cases added by him.

“Home” and “Log out” are two functionalities available and with the same effect to any logged in user. “Home” redirects users to his Home page and “Log out” take users out of Array CGH DB by removing Session and Cookies variables defined when logged in.

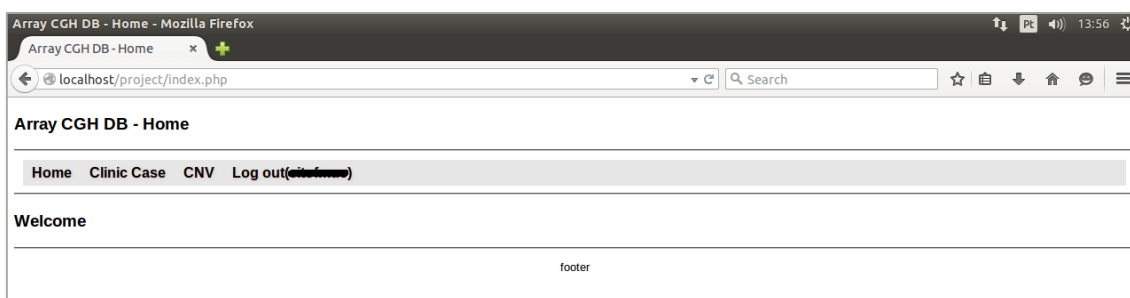


Figure 20 –Researcher: Home page and Navigation Bar. The Navigation Bar allows to select “Home”, “Clinic Case”, “CNV” or “Log out”. “Home” redirects a Researcher to this page, “Clinic Case” allows him add and manage clinic cases added by him, “CNV” is for matching CNVs and “Log out” to quit from Array CGH DB.

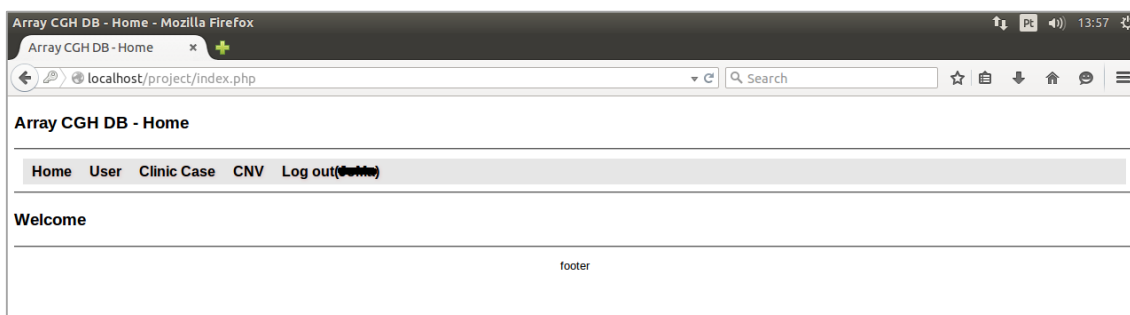


Figure 21 – Administrator: Home page and Navigation Bar. The Navigation Bar allows to select “Home”, “User”, “Clinic Case”, “CNV” or “Log Out”. “Home” redirects an Administrator to this page, “User” is to manage users linked to his laboratory, “Clinic Case” allows manage clinic cases linked to his laboratory, “CNV” is for matching CNVs and “Log out” to quit from Array CGH DB.

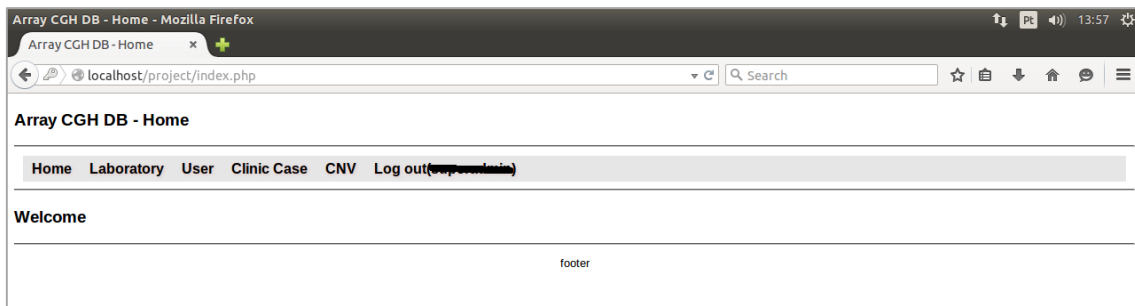


Figure 22 –Super-Administrator: Home page and Navigation Bar. The Navigation Bar allows to select “Home”, “Laboratory”, “User”, “Clinic Case”, “CNV” or “Log Out”. “Home” redirects a Super-Administrator to this page, “Laboratory” lets a Super-Administrator to manage and create new laboratories, “User” allows to manage Administrators and Researchers, “Clinic Case” allows him to manage all clinic cases, “CNV” is for matching CNVs and “Log out” to quit from Array CGH DB.

5.3. Software Main Features

There are two main features in the software. The first allows the upload of new clinic cases to be stored in the database, and the second the query of the database about a CNV match. The query results are presented in a way that makes it easy to be interpreted by clinicians.

5.3.1. Upload of Clinic Cases

The upload of Clinic Cases can be done by submitting a form where a Researcher has to specify the Clinic Cases format, and by uploading a *.zip* file with a set of spreadsheets (*.xls*) containing information about arrayCGH reports. There are two possible choices of format for arrayCGH reports: “*IntervalBasedReport*” and “*TabelaClasses*”. Those formats are used and known at *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra*. The “*TabelaClasses*” is a handmade report which is delivered to clinicians and the “*IntervalBasedReport*” is a report exported by CytoGenomics, a software for CNV analyses.

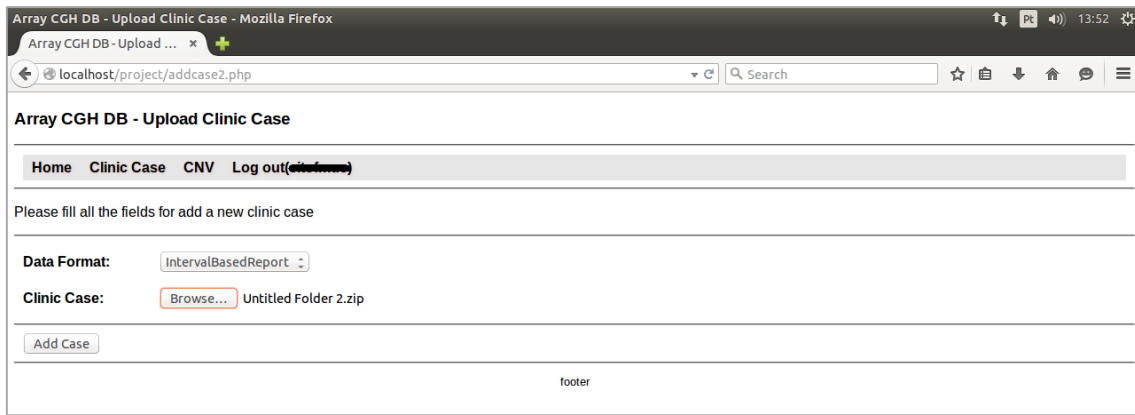


Figure 23 - Researcher: Upload Clinic Case page. On this example is submitted a .zip file with “IntervalBasedReport” format.

Those two formats do not provide all the information about CNV and Genes needed in the database. The “*TabelaClasses*” format does not provide information about the genes contained on a certain CNV and the “*IntervalBasedReport*” format does not have the “type” field for CNVs.

If a set of arrayCGH results is uploaded in the “*TabelaClasses*” format, the “gene” field is not there and reports are uploaded without this information. For “*IntervalBasedReport*” format there are two ways for get the “type” field information:

1. Researchers can introduce a column on the “*IntervalBasedReport*” spreadsheet with the “type” information.
2. Researchers can provide the spreadsheets of “*TabelaClasses*” and “*IntervaleBasedReport*” on the same .xls file. The software will detect the missing field on “*IntervalBasedReport*” and will try a match between those two spreadsheets in order to get the “type” information for each CNV on the “*IntervalBasedReport*”.

After an upload of a .zip file, for each .xls file in the set, the software tries to read it, and looks for a spreadsheet with the name of the format indicated. This is performed with *PHPExcel*, an open source set of classes for the PHP programming language, which

allows to write to and to read from different spreadsheet formats. Then it verifies if all the fields required are present and if every CNV record has all the fields filled.

Subsequently, a validation process is performed which verifies if the uploaded data have the expected format. If one file fails, the corresponding Clinic Case is not stored and the software continues with the next *.xls* file. In the end, the user gets a report of the upload process, *i. e.* successfully stored Clinic Cases or errors that occurred during the process are displayed. The errors displayed are descriptive in order to help Researchers to fix them for a successful re-upload.

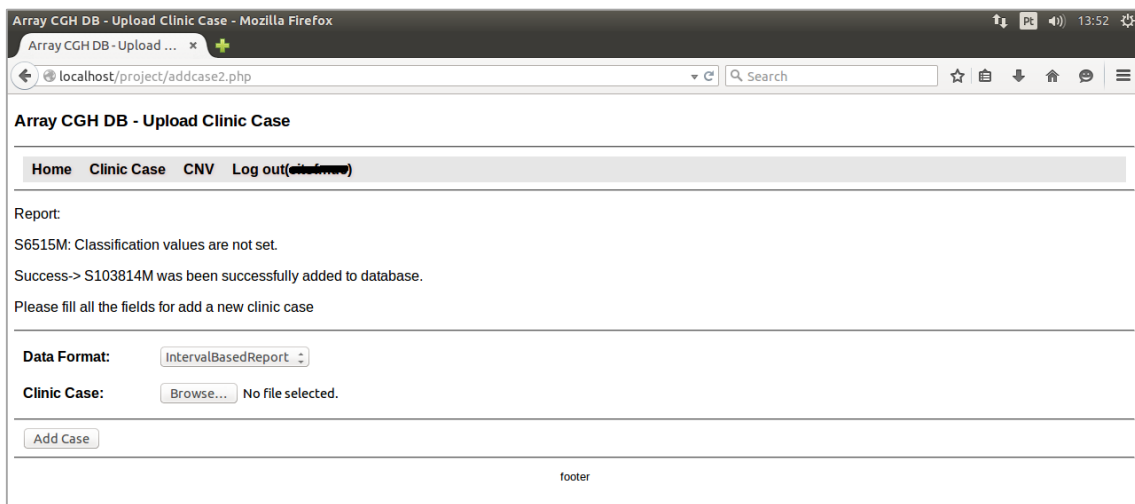


Figure 24 – Researcher: Upload Report of Clinic Cases. On the report is showed information about the upload of two clinic cases, S6515M and S103814M. The first one, is not recorded on DB because it hasn't the "classification" field for CNVs. S103814M passes on validation and is successfully added to DB.

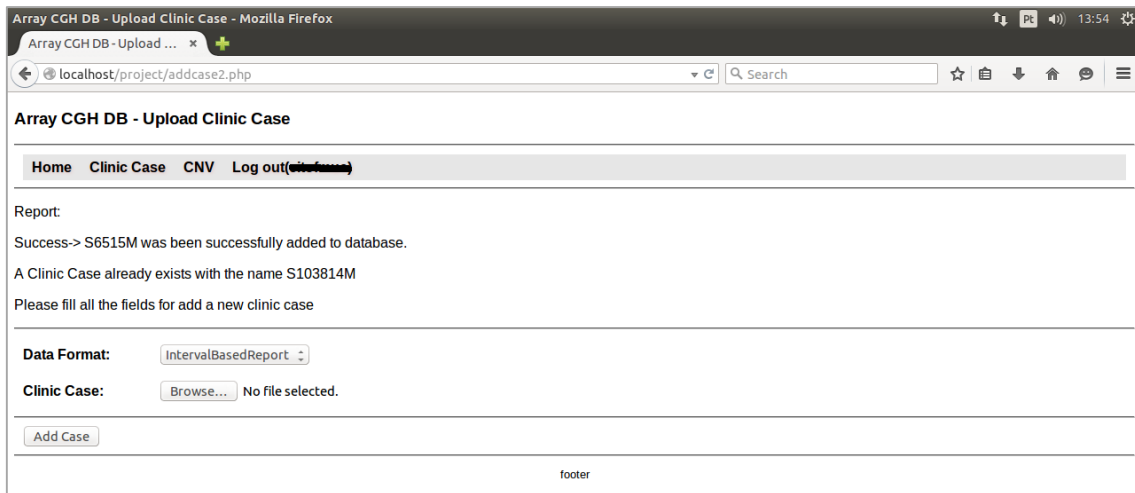


Figure 25 Researcher: Upload Report of Clinic Cases. In this example, the clinic case S6515M was successfully recorded after being added the “classification” field. The clinic case S103814 already exists on DB and wasn’t re-added.

For uploaded .zip files there is a maximum file size of 5Mb, and after the information storage the uploaded files are deleted. The objectives are not to overload the server memory and to keep the hard disk storage free of unwanted files.

5.3.2. Display CNV matching

The CNV matching is performed by sending a form with the information of a chromosome location or a gene name. After that, the software queries the database with the user request settings and builds a graph bar with the query results. Each bar of the graph represents a CNV matching the user request, in a way that all CNVs are horizontally aligned, taking as reference the request settings from the user, and the CNV attributes for start, stop and size. (Figure 26)

The bars have colours, each colour representing the CNV alteration type: red for deletion, blue for duplication, black for variations and brown for unknown. When placing the mouse over a CNV, the attributes start, stop and the alteration type are displayed (Figures 26 and 27).

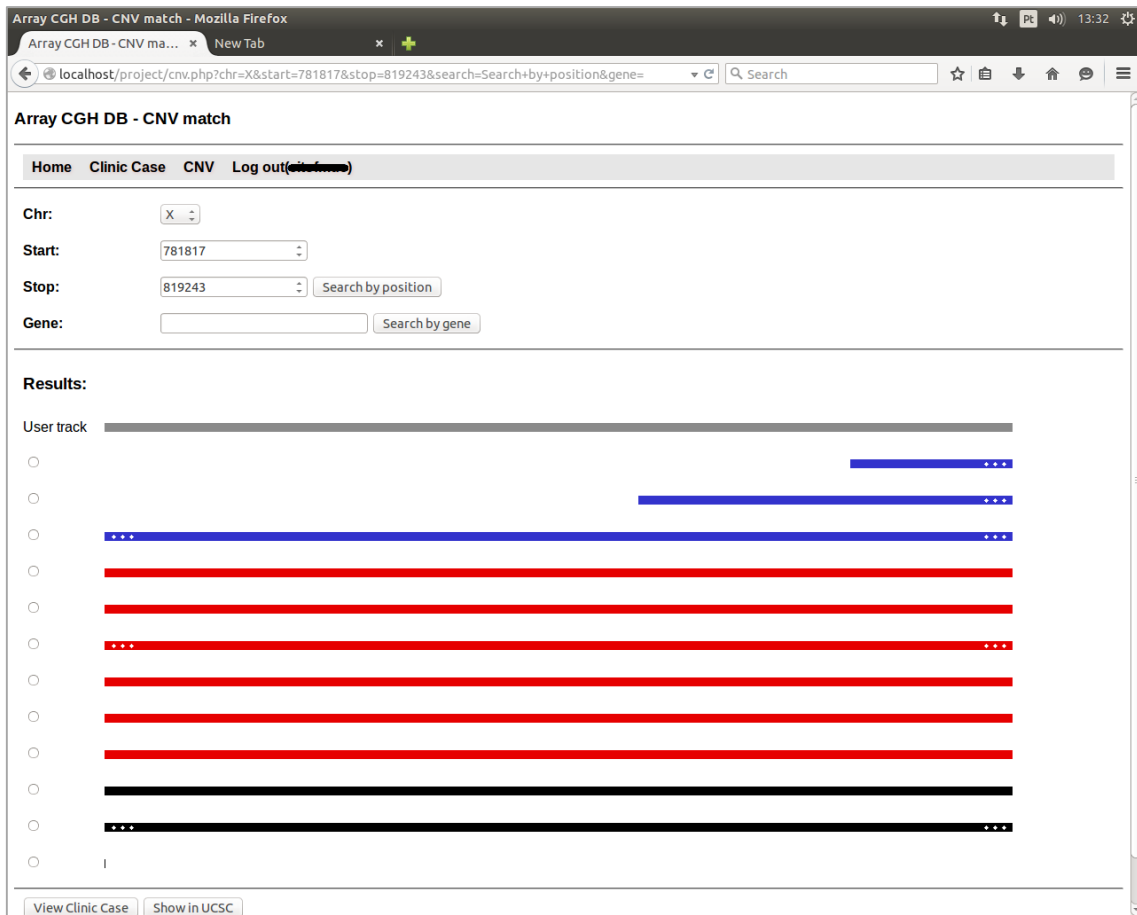


Figure 26 – Researcher: CNV match with chromosome location. It is asked for a CNV matching for chromosome X with positions starting on 781817(bp) and stopping on 819243(bp). Results show the User Track bar and the CNVs that match the User Track. CNVs larger than the User Track are represented with white dots in the end.

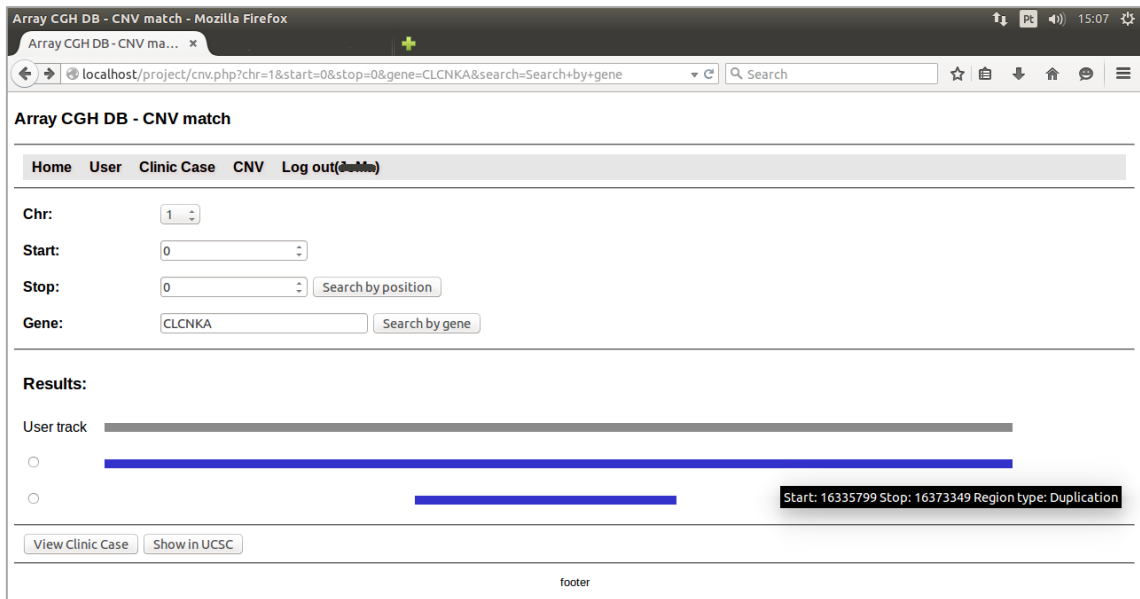


Figure 27 - Researcher: CNV match with gene name. It is asked for a CNV matching for the gene CLCNKA. Results showed the User Track bar, the biggest CNV size in the database linked to CLCNKA gene. All the other CNVs linked to the same gene are showed horizontally aligned with the User Track.

Each CNV is also available for a new CNV matching with the laboratory records or with the records on the UCSC Genome Database, a database with a large record collection of CNVs. If a user selects a CNV matching with the records of the UCSC Genome Database (<https://genome.ucsc.edu>), he will be redirected to the UCSC site and will get a CNV matching according to his request. This is a very simple process but very useful for clinicians, performed by passing the information of the selected chromosome location via URL (Uniform Resource Locator) for a CNV matching on UCSC Genome Database. Another important aspect about this functionality is the capacity of arrayCGH to communicate with other well established solutions in the assessment of pathogenicity of CNVs.

To draw bars, a graphical library called GD (Graphics Draw) is used, which allows PHP scripts to dynamically generate images in popular web formats, such as GIF, JPEG, and PNG, and either return them to the web browser for display or write them to a file on the server.

5.4. Other Features

Besides the upload of Clinic Cases and the display of CNV matches, the arrayCGH DB allows the management of Clinic Cases for Researchers, Administrators and Super-Administrators, management of Users by Administrators and Super-Administrators and management of Laboratories by Super-Administrator. In the next section these functionalities are presented in detail.

5.4.1. Management of Clinic Cases

Researchers, Administrators and Super-Administrators are able to manage Clinic Cases. However this management is different according to each role, as defined in the Software Requirements section.

When accessing to “Clinic Case” through the Navigation Bar, the available Clinic Cases are shown according to the user role, laboratory and ID. The Researcher can select view, edit or remove Clinic Cases added by him (Figure 28), the Administrator is able to view, remove or change the state, to be hidden or public, of Clinic Cases which are linked to his laboratory, and the Super-Administrator is able to perform the same tasks of an Administrator but for all Clinic Cases on the DB (Figure 29). On this page all users are able to sort Clinic Cases by Case Name, Gender, State or Join Date, and to search by Case Name, providing them an easy mechanism to find the information they need (Figure 30).

5. Results and Discussion

The screenshot displays a web browser window with the title 'Array CGH DB - Clinic Case - Mozilla Firefox'. The address bar shows 'localhost/project/cliniccase.php?sort=4'. The page content includes a navigation menu with 'Home', 'Clinic Case', 'CNV', and 'Log out(████████)'. Below the menu is a link to 'Add new clinic case'. The main area contains a table of clinic cases:

Case Name	Gender	State	Join Date	Comment
<input type="radio"/> ██████████	Female	Hidden	2015-04-29	
<input type="radio"/> ██████████	Female	Hidden	2015-04-29	
<input type="radio"/> ██████████	Female	Hidden	2015-05-05	
<input type="radio"/> ██████████	Female	Hidden	2015-05-05	
<input type="radio"/> ██████████	Female	Hidden	2015-05-05	
<input type="radio"/> ██████████	Male	Hidden	2015-04-29	
<input type="radio"/> ██████████	Male	Hidden	2015-04-29	
<input type="radio"/> ██████████	Male	Hidden	2015-04-29	
<input type="radio"/> ██████████	Male	Hidden	2015-05-05	
<input type="radio"/> ██████████	Male	Hidden	2015-04-29	
<input type="radio"/> ██████████	Male	Hidden	2015-04-29	
<input type="radio"/> ██████████	Male	Hidden	2015-04-29	
<input type="radio"/> ██████████	Unknown	Hidden	2015-04-29	
<input type="radio"/> ██████████	Unknown	Hidden	2015-05-05	
<input type="radio"/> ██████████	Unknown	Hidden	2015-04-29	
<input type="radio"/> ██████████	Unknown	Hidden	2015-05-05	
<input type="radio"/> ██████████	Unknown	Hidden	2015-04-29	
<input type="radio"/> ██████████	Unknown	Hidden	2015-05-05	
<input type="radio"/> ██████████	Unknown	Hidden	2015-05-05	
<input type="radio"/> ██████████	Unknown	Hidden	2015-04-29	
<input type="radio"/> ██████████	Unknown	Hidden	2015-05-05	
<input type="radio"/> ██████████	Unknown	Hidden	2015-04-29	
<input type="radio"/> ██████████	Unknown	Hidden	2015-05-05	
<input type="radio"/> ██████████	Unknown	Hidden	2015-04-29	
<input type="radio"/> ██████████	Unknown	Hidden	2015-05-05	

At the bottom of the page, there are buttons for 'View', 'Edit', and 'Remove', and a search box labeled 'Search by Case Name'. The footer contains the text 'booter'.

Figure 28 – Researcher: Clinic Case page. Clinic Cases added by a Researcher are shown on this page and he is able to select one of them to view, edit or remove. A Researcher is also able to sort the displayed information by Case Name, Gender, State or Join Date, and to perform a search by Case Name.

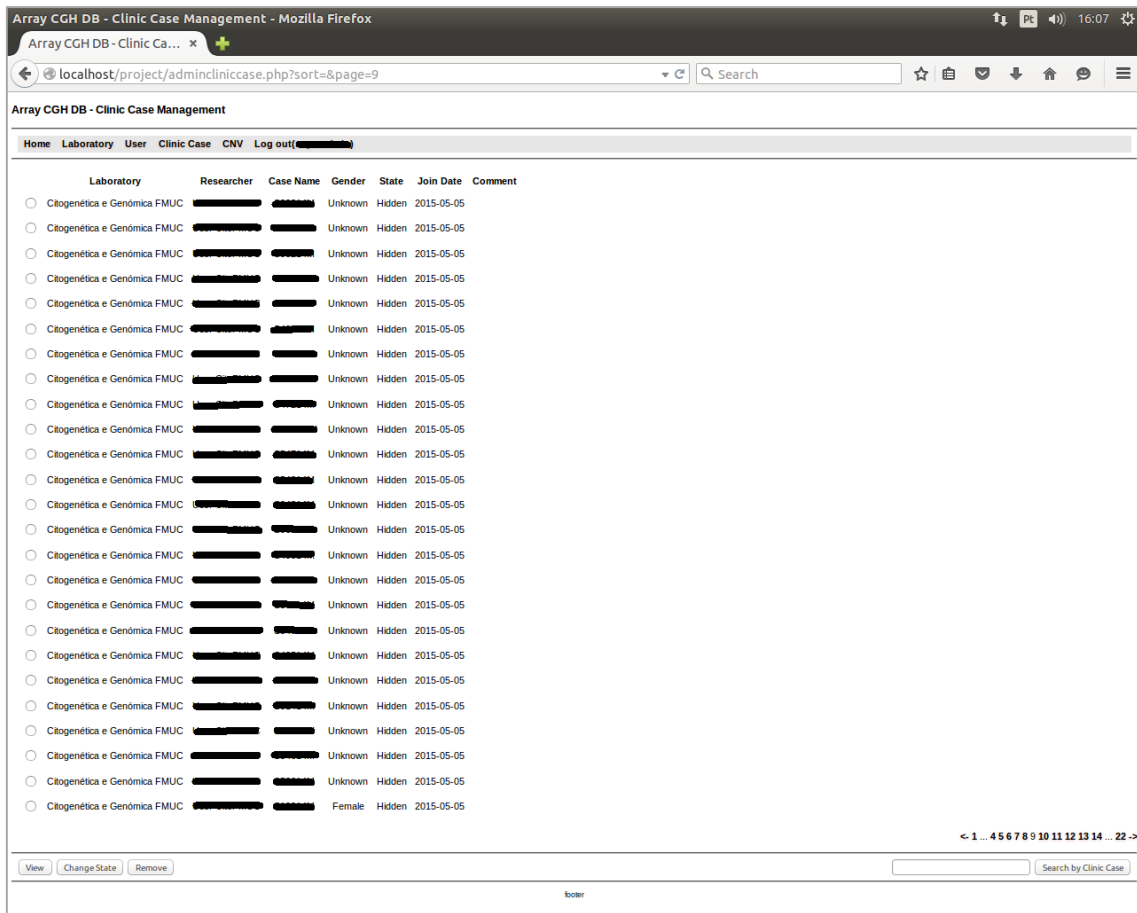
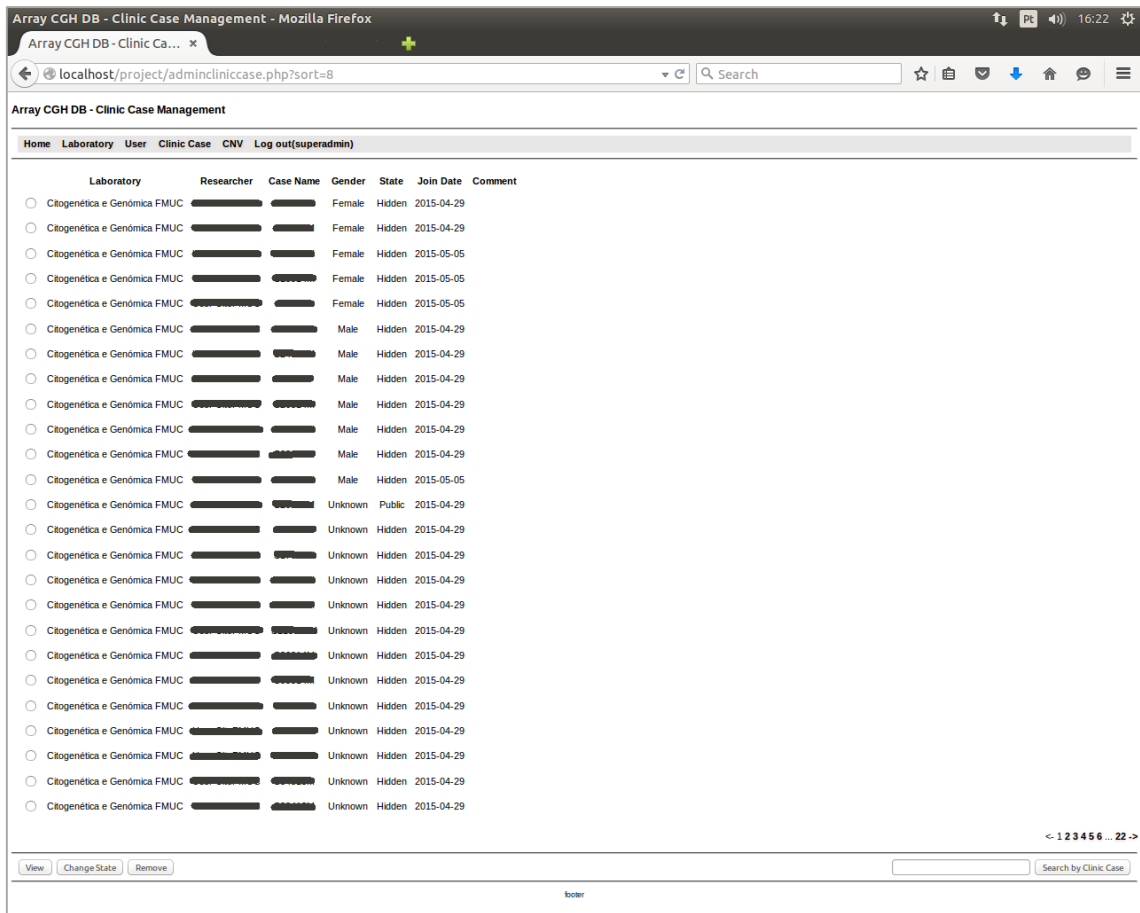


Figure 29 – Super-Administrator: Clinic Case page. All Clinic Cases added are shown on this page and Super Administrator is able to select one of them to view, change state or remove. A Super Administrator is also able to sort the displayed information by Laboratory, Researcher, Case Name, Gender, State or Join Date, and to perform a search by Case Name.

5. Results and Discussion

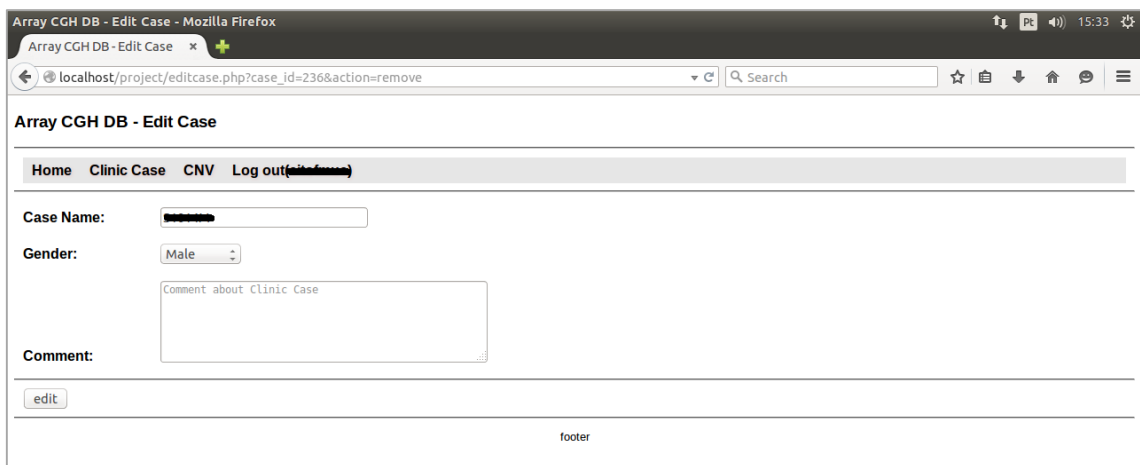


The screenshot shows a web browser window titled "Array CGH DB - Clinic Case Management - Mozilla Firefox". The address bar shows the URL "localhost/project/admincliniccase.php?sort=8". The page content includes a navigation menu with "Home", "Laboratory", "User", "Clinic Case", "CNV", and "Log out(superadmin)". Below the menu is a table with the following columns: "Laboratory", "Researcher", "Case Name", "Gender", "State", "Join Date", and "Comment". The table contains 25 rows of data, all with "Laboratory" as "Citogenética e Genómica FMUC". The "Gender" column shows a mix of "Female", "Male", and "Unknown", and the "State" column shows "Hidden" and "Public". At the bottom of the table, there are buttons for "View", "Change State", and "Remove", and a search bar labeled "Search by Clinic Case".

Laboratory	Researcher	Case Name	Gender	State	Join Date	Comment
Citogenética e Genómica FMUC			Female	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Female	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Female	Hidden	2015-05-05	
Citogenética e Genómica FMUC			Female	Hidden	2015-05-05	
Citogenética e Genómica FMUC			Female	Hidden	2015-05-05	
Citogenética e Genómica FMUC			Male	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Male	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Male	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Male	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Male	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Male	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Male	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Male	Hidden	2015-05-05	
Citogenética e Genómica FMUC			Unknown	Public	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	
Citogenética e Genómica FMUC			Unknown	Hidden	2015-04-29	

Figure 30 – Super Administrator: Clinic Cases sorted by genre.

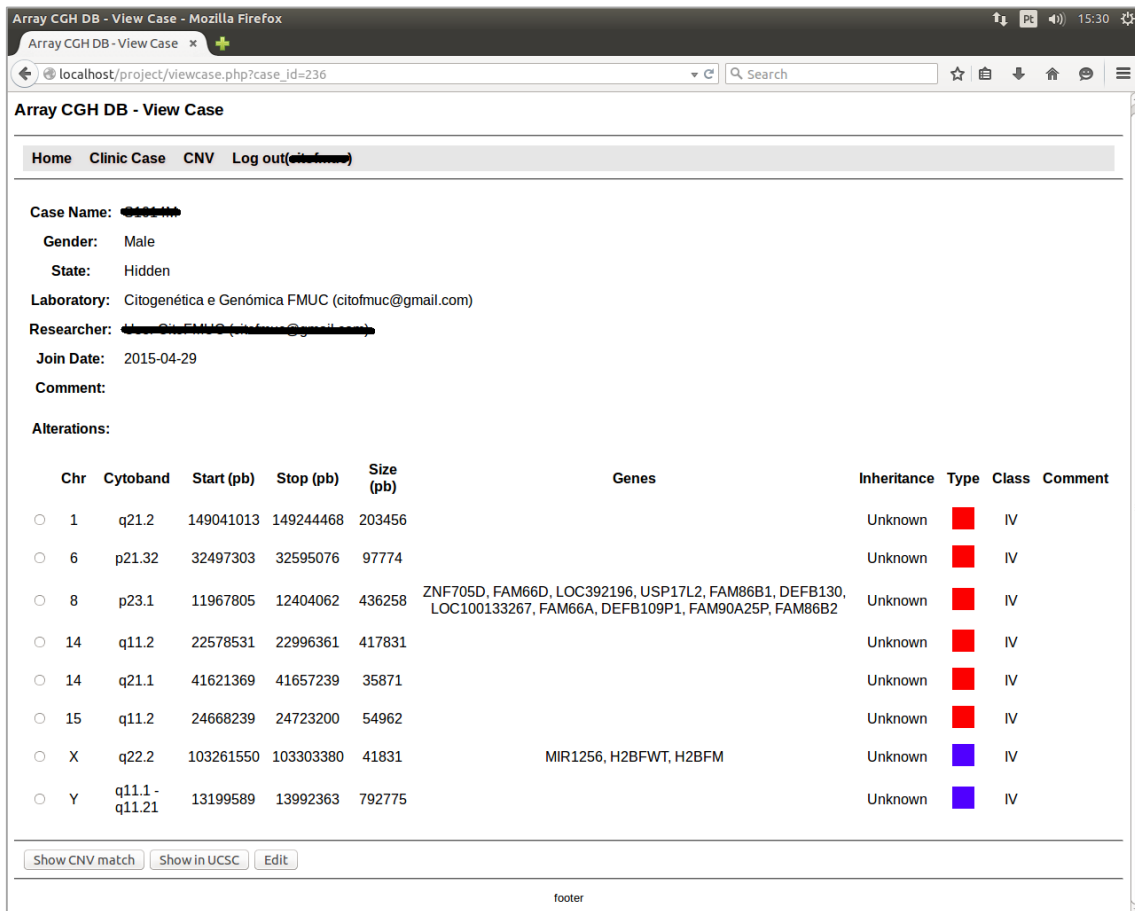
By select “Edit” on Clinic Case page, a Researcher is able to update the information of the Clinic Case selected (Figure 31).



The screenshot shows a web browser window titled "Array CGH DB - Edit Case - Mozilla Firefox". The address bar shows the URL "localhost/project/editcase.php?case_id=236&action=remove". The page content includes a navigation menu with "Home", "Clinic Case", "CNV", and "Log out(Researcher)". Below the menu is a form with the following fields: "Case Name" (text input), "Gender" (dropdown menu with "Male" selected), and "Comment" (text area). At the bottom of the form, there is an "edit" button. The footer of the page contains the word "footer".

Figure 31 – Researcher: Edit Clinic Case page. The Case Name, Gender and Comment of a certain Clinic Case are eligible to be edited by Researchers.

When the option “View” is selected for a certain Clinic Case, the information linked to this case is shown, such as Case Name, Gender, State, Laboratory (name and email), the Researcher linked to the Clinic Case and his email, any Comment and all CNVs of the Clinic Case. Any CNV displayed on this page is available for a CNV match with the arrayCGH DB or UCSC data, and can be examined on section 5.3.2 Display CNV matching. If a user clicks on a gene name, he will be redirected to GeneCards Human Gene Database (<http://www.genecards.org>), in order to get more information about the gene selected. Through this page, the Researcher is able to edit CNVs if the Clinic Case is linked to him. (Figures 32 and 33)



The screenshot shows a web browser window titled "Array CGH DB - View Case". The address bar shows the URL "localhost/project/viewcase.php?case_id=236". The page content includes a navigation menu with "Home", "Clinic Case", "CNV", and "Log out(anonymous)". Below the menu, the case information is displayed:

Case Name: ██████████
 Gender: Male
 State: Hidden
 Laboratory: Citogenética e Genómica FMUC (citofmuc@gmail.com)
 Researcher: ██████████ (citofmuc@gmail.com)
 Join Date: 2015-04-29
 Comment:

Alterations:

Chr	Cytoband	Start (pb)	Stop (pb)	Size (pb)	Genes	Inheritance	Type	Class	Comment
<input type="radio"/>	1	q21.2	149041013	149244468	203456		Unknown	IV	
<input type="radio"/>	6	p21.32	32497303	32595076	97774		Unknown	IV	
<input type="radio"/>	8	p23.1	11967805	12404062	436258	ZNF705D, FAM66D, LOC392196, USP17L2, FAM86B1, DEFB130, LOC100133267, FAM66A, DEFB109P1, FAM90A25P, FAM86B2	Unknown	IV	
<input type="radio"/>	14	q11.2	22578531	22996361	417831		Unknown	IV	
<input type="radio"/>	14	q21.1	41621369	41657239	35871		Unknown	IV	
<input type="radio"/>	15	q11.2	24668239	24723200	54962		Unknown	IV	
<input type="radio"/>	X	q22.2	103261550	103303380	41831	MIR1256, H2BFWT, H2BFM	Unknown	IV	
<input type="radio"/>	Y	q11.1 - q11.21	13199589	13992363	792775		Unknown	IV	

At the bottom of the table, there are three buttons: "Show CNV match", "Show in UCSC", and "Edit".

Figure 32 – Researcher: View Clinic Case page. All the Clinic Case information is shown, including the information of the CNVs linked to the case. If the Clinic Case is linked to the Researcher, any CNV showed can be edited. All users are able to use “Show CNV match” or “Show in UCSC” and by clicking on a gene name more information about that gene will be provided on Gene Cards Human Gene Database (<http://www.genecards.org>).

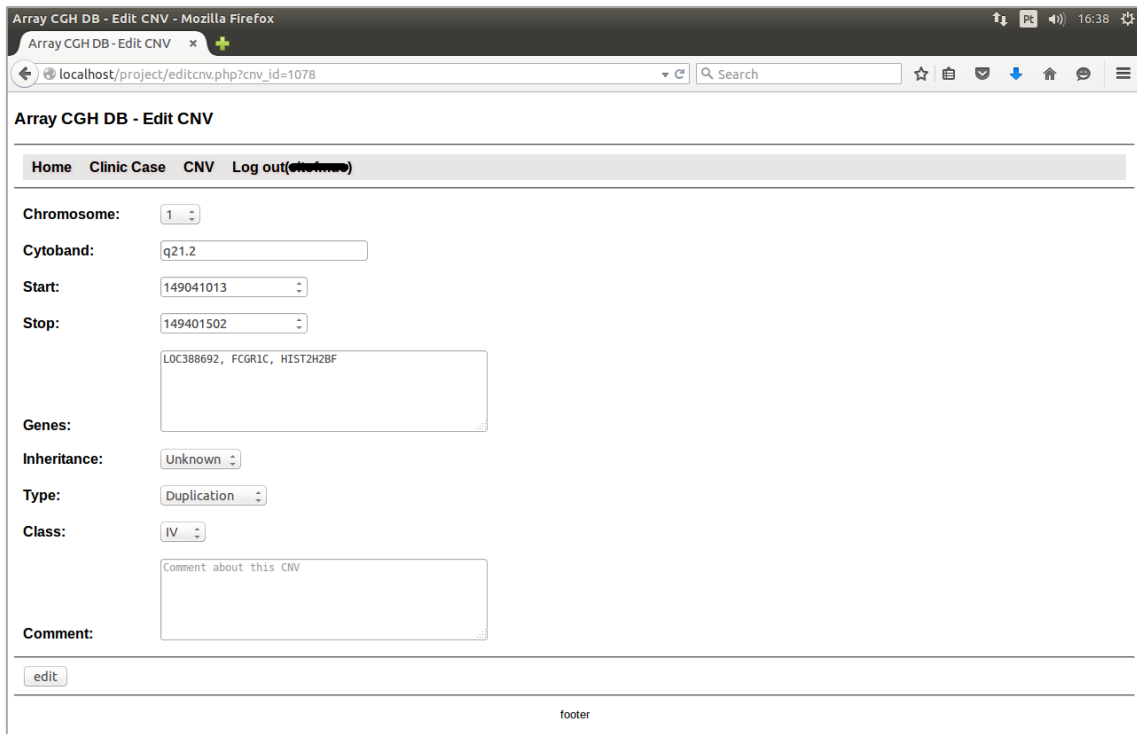


Figure 33 – Researcher: Edit CNV page. Researcher is capable to edit CNV information, such as Chromosome, Cytoband, Start, Stop, Genes, Inheritance, Type, Class and Comment, of Clinic Cases linked to him.

When a user selects “Remove” from the “Clinic Case” page, he is informed of the information he is about to delete and has to confirm by selecting “Yes” and then “Remove Clinic Case” in order to remove the Clinic Case (Figure 34).

Array CGH DB - View Case

Home Clinic Case CNV Log out(**citofmuc**)

Case Name: **citofmuc**
 Gender: Male
 State: Hidden
 Laboratory: Citogenética e Genómica FMUC (citofmuc@gmail.com)
 Researcher: **citofmuc** (citofmuc@gmail.com)
 Join Date: 2015-04-29
 Comment:

Alterations:

Chr	Cytoband	Start (pb)	Stop (pb)	Size (pb)	Genes	Inheritance	Type	Class	Comment
1	q21.2	149041013	149244468	203456		Unknown	■	IV	
6	p21.32	32497303	32595076	97774		Unknown	■	IV	
8	p23.1	11967805	12404062	436258	ZNF705D, FAM66D, LOC392196, USP17L2, FAM86B1, DEFB130, LOC100133267, FAM66A, DEFB109P1, FAM90A25P, FAM86B2	Unknown	■	IV	
14	q11.2	22578531	22996361	417831		Unknown	■	IV	
14	q21.1	41621369	41657239	35871		Unknown	■	IV	
15	q11.2	24668239	24723200	54962		Unknown	■	IV	
X	q22.2	103261550	103303380	41831	MIR1256, H2BFWT, H2BFM	Unknown	■	IV	
Y	q11.1 - q11.21	13199589	13992363	792775		Unknown	■	IV	

Yes No

footer

Figure 34 – Researcher: Remove Clinic Case page. All information about to be deleted is shown, and in order remove the Clinic Case, users have to confirm by selecting “Yes” and then “Remove Clinic Case”.

5.4.2. Management of Users

Super-Administrators and Administrators are able to manage users by selecting “Users” in the Navigation Bar. Super-Administrators can view information about any user (Super-Administrators, Administrators and Researchers), such as Name, Username, Role, State, Laboratory, Email and Join Date, and are capable to remove or change the state, to block or approve, of any Researcher or Administrator (Figure 35). Administrators can perform the same tasks of Super-Administrators but only for Researchers linked to their laboratory (Figure 36). Super-Administrators or Administrators are also able to search any user by name or sort any information displayed (Figures 35 and 36).

5. Results and Discussion

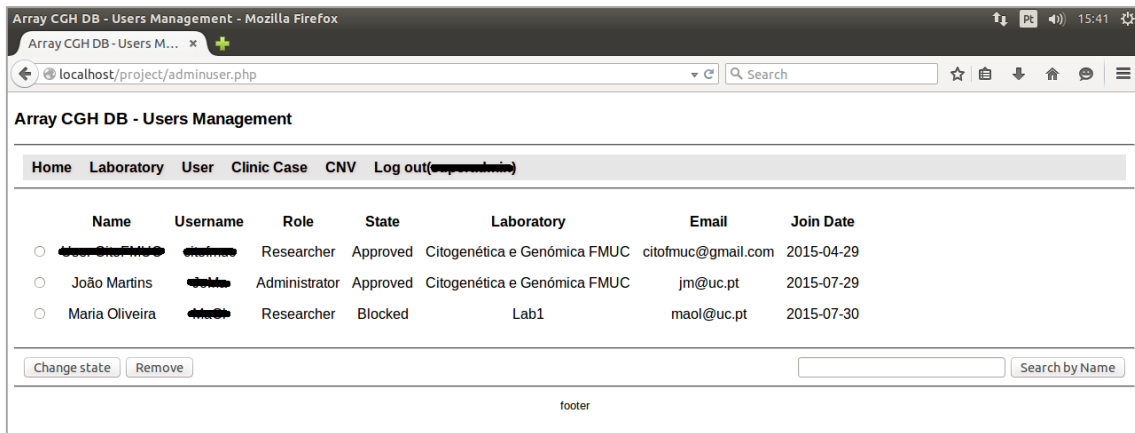


Figure 35 – Super-Administrator: “User” management page. Name, Username, Role, State, Laboratory, Email and Join Date of all users are shown, and any Researcher or Administrator can be removed or have his state set to Blocked or Approved by a Super-Administrator. A Super-Administrator is also able to search any user by Name or sort any information shown.

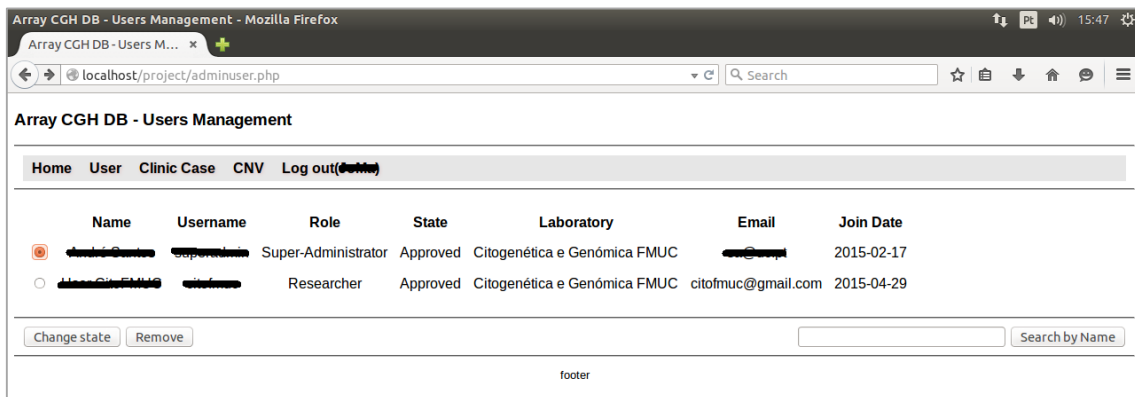


Figure 36 - Administrator: “User” management page. Name, Username, Role, State, Laboratory, Email and Join Date of all users are shown, and any Researcher can be removed or have his state set to Blocked or Approved by an Administrator of the same laboratory. Administrators are also able to search any user by Name or sort any information shown.

When a user is selected for removal, a page with all the information about to be removed is shown and in order to delete it, the Administrator or Super-Administrator, need to confirm the operation by selecting “Yes” and then “Remove”. If the user selected for removal is a Researcher, all Clinic Cases linked to him will be removed, too. (Figure 37)

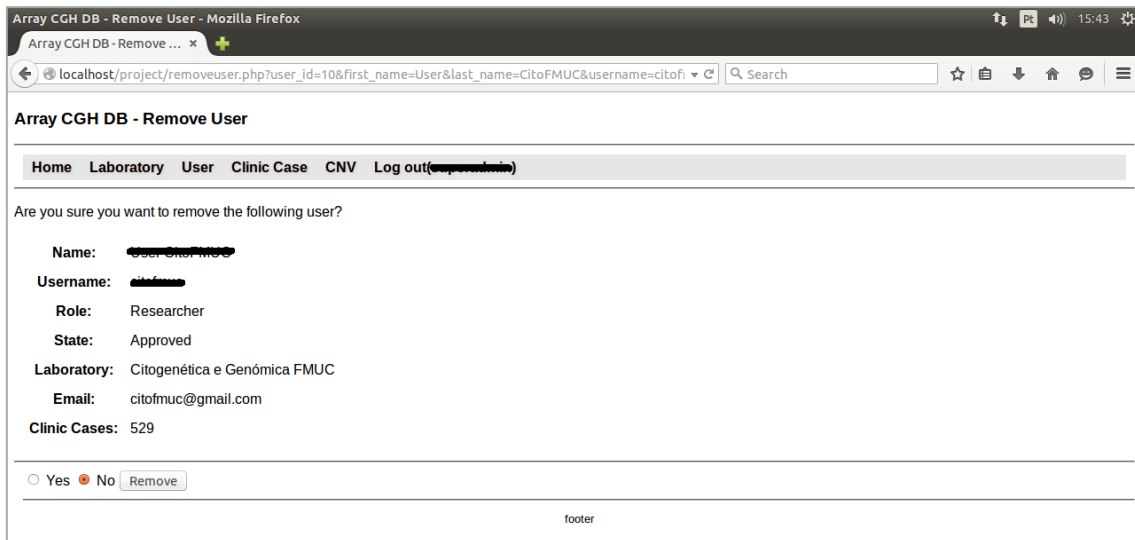


Figure 37 – Super-Administrator: Remove User page. All information about the user about to be removed is shown and if he was linked to clinic cases, they will be removed, too.

To change a user state, Super-Administrators or Administrator have to select the user through the “User” option on Navigation Bar and then choose “Change State”. After that, a message is shown and the user state is altered. (Figures 38 and 39)



Figure 38 Super-Administrator: Message displayed when changing a user from blocked to approved. The state of user Maria Oliveira is changed to approved.

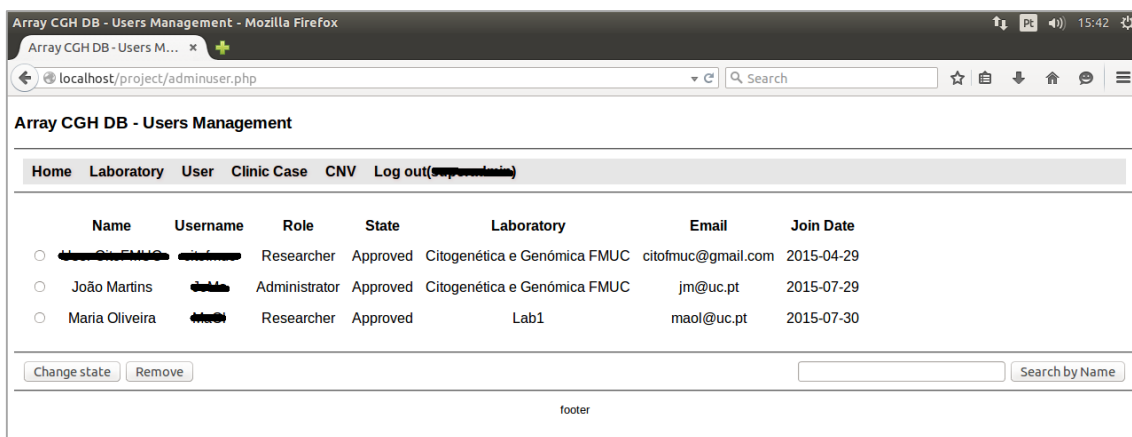


Figure 39 – Super-Administrator: User approved. On Figure 35, user Maria Oliveira is blocked, but after applying the “Change State” option her state is changed to approved.

If Super-Administrators or Administrators try to “Remove” or “Change State” of a user they are not allowed to, a message is shown and the action is not performed (Figure 40).



Figure 40 – Administrator: Message shown when an Administrator tries to remove a user he has not permission to. Although an administrator is able to view a Super-Administrator (Figure 36), he is not able of remove him.

5.4.3 Management of Laboratories

Super-Administrators are able to add new laboratories and remove or edit the existing laboratories through the “Laboratory” option on the Navigation Bar. When this option is selected, all information about the registered laboratories is shown (Name, Email, Contact, and Join Date). Super-Administrators are also able of sort the

laboratories by any of attribute shown or search a laboratory by its Name. (Figures 41, 42 and 43)

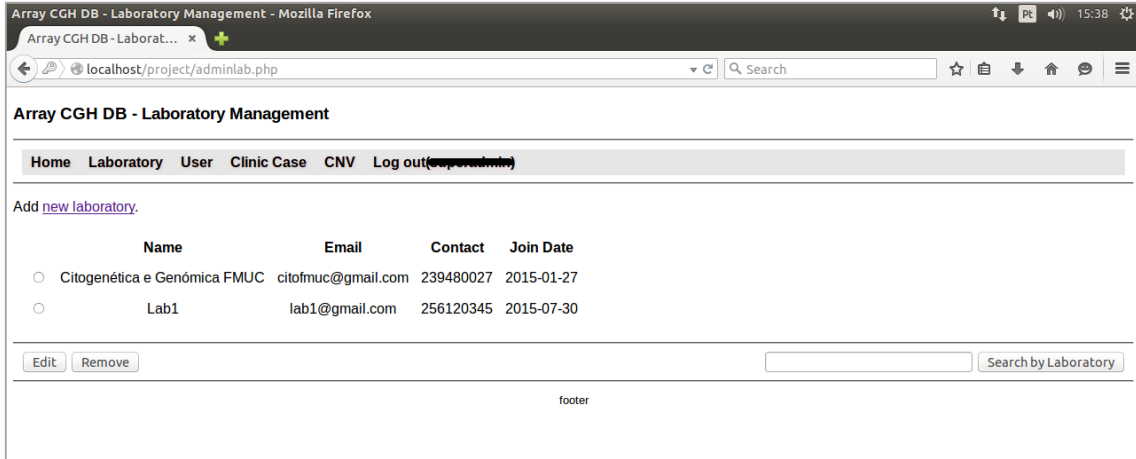


Figure 41 – Super-Administrator: Laboratory Management page. All information about the registered laboratories is shown (Name, Email, Contact, and Join Date). Super-Administrator is able to remove or edit any laboratory shown, or add a new one. He is also able to sort the laboratories by any of the attributes shown or search a laboratory by its Name.

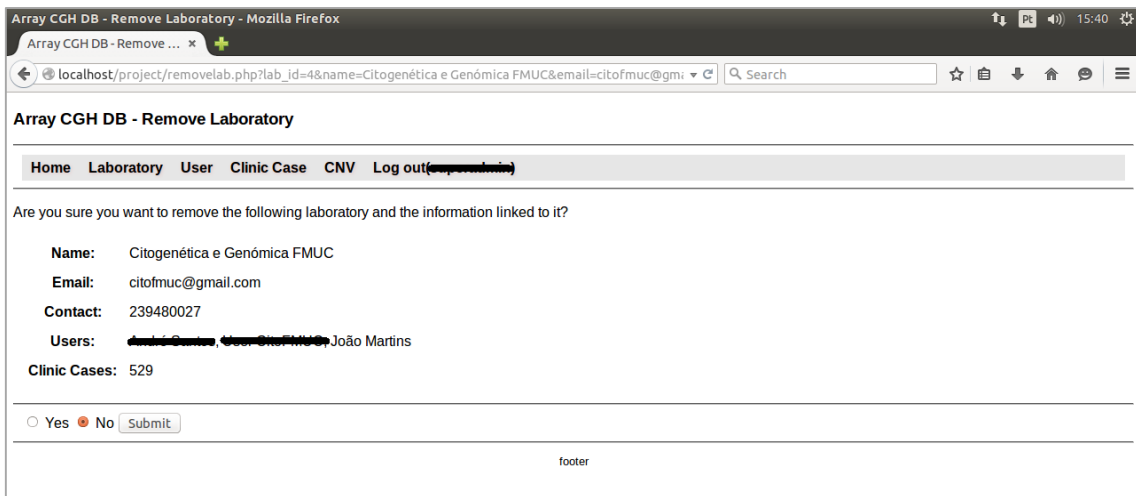


Figure 42 – Super-Administrator: Remove Laboratory page. Is shown all information of the laboratory, and all users and the number of Clinic Cases linked to the laboratory about to be removed.

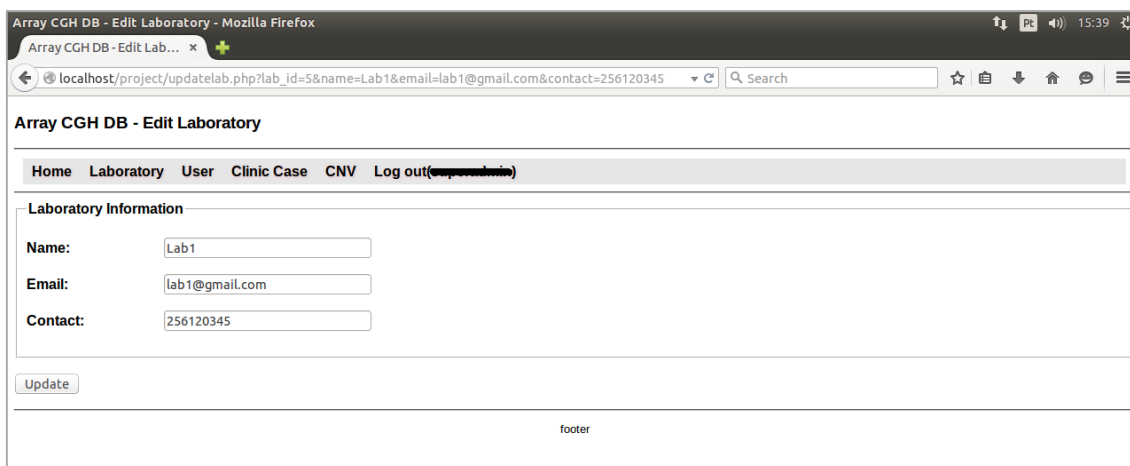


Figure 43 – Super-Administrator: Edit Laboratory page. Here, Name, Email and Contact of a laboratory can be changed.

5.5. Validation

In order to perform a validation for the arrayCGH DB, 529 Clinic Cases with several CNV records from *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra* were uploaded using the “Upload of Clinic Cases”. After that, some queries were performed to report about the information stored in the database. At this moment, arrayCGH DB has 3254 CNV records. Below is given information about size, alteration type, classification and chromosomes of the 3240 CNV records stored. CNV records with wrong information, bugs, without class or alteration type will be discussed on section 6. Bugs Detected and Fixed.

5.5.1 Distribution of CNVs by size

Figure 44 report about the distribution of CNV sizes in the arrayCGH DB.

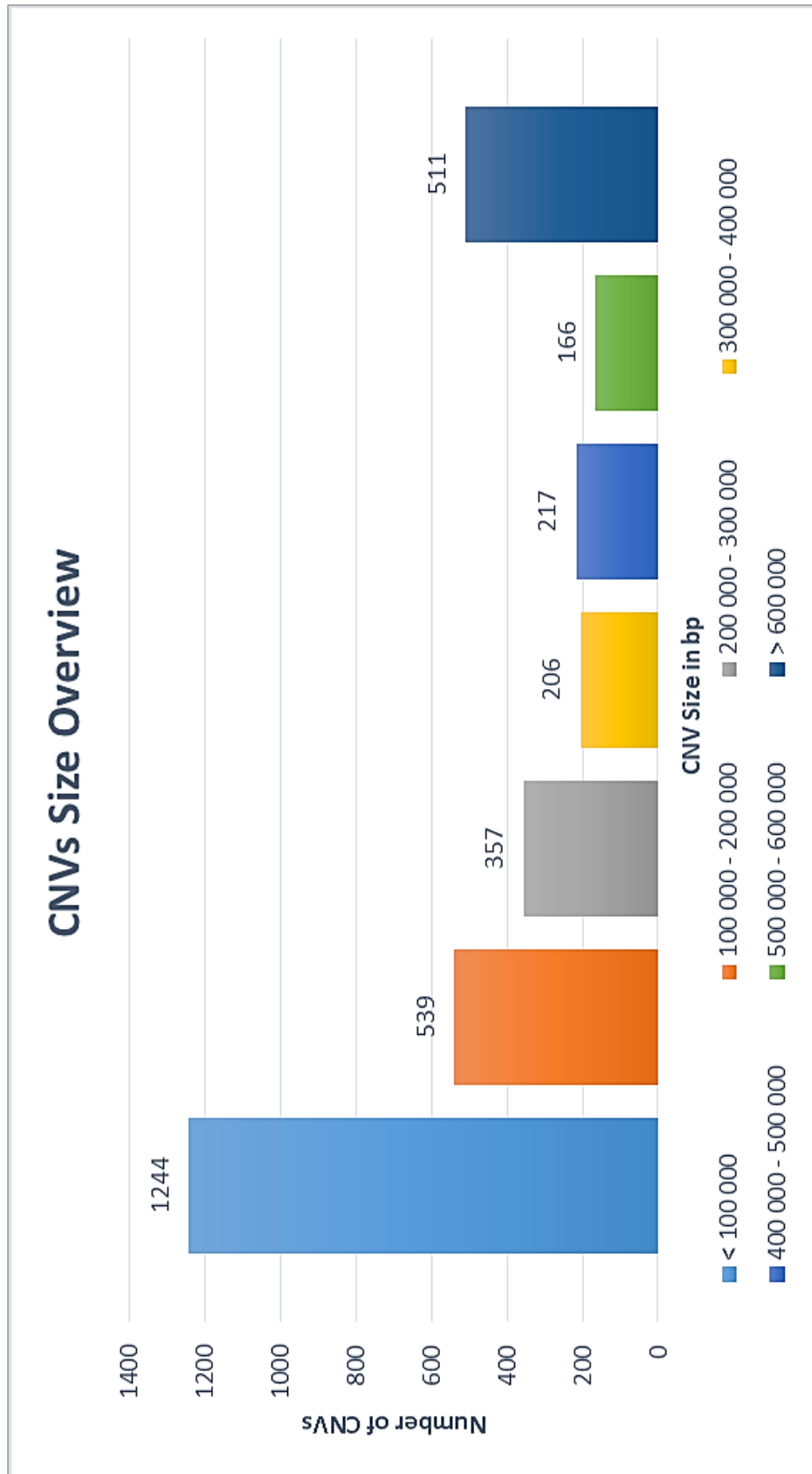


Figure 44 - CNVs Size Overview. *Distribution of CNV size in the arrayCGH DB.*

5.5.2. Distribution of CNVs by alteration type type

Figure 45 shows the distribution by alteration type of CNVs in the arrayCGH DB.

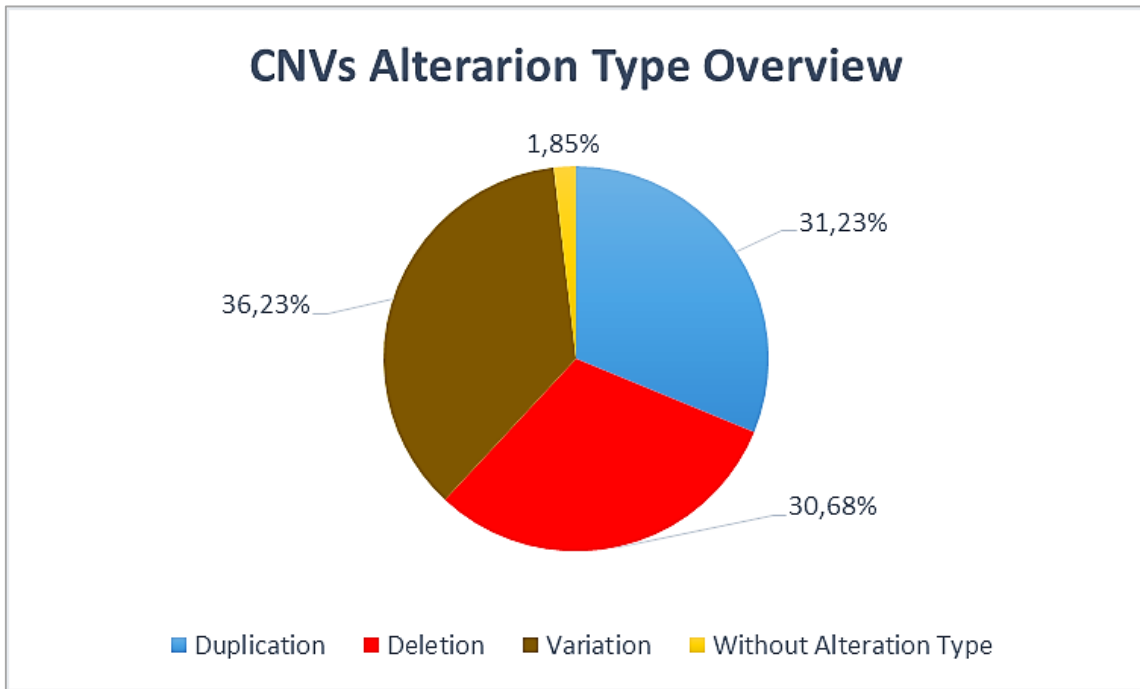


Figure 45 - CNVs stored in the arrayCGH DB alteration type overview.

5.5.3. Distribution of CNVs alteration type by size

Figure 46 shows the distribution of CNVs alteration type by size.

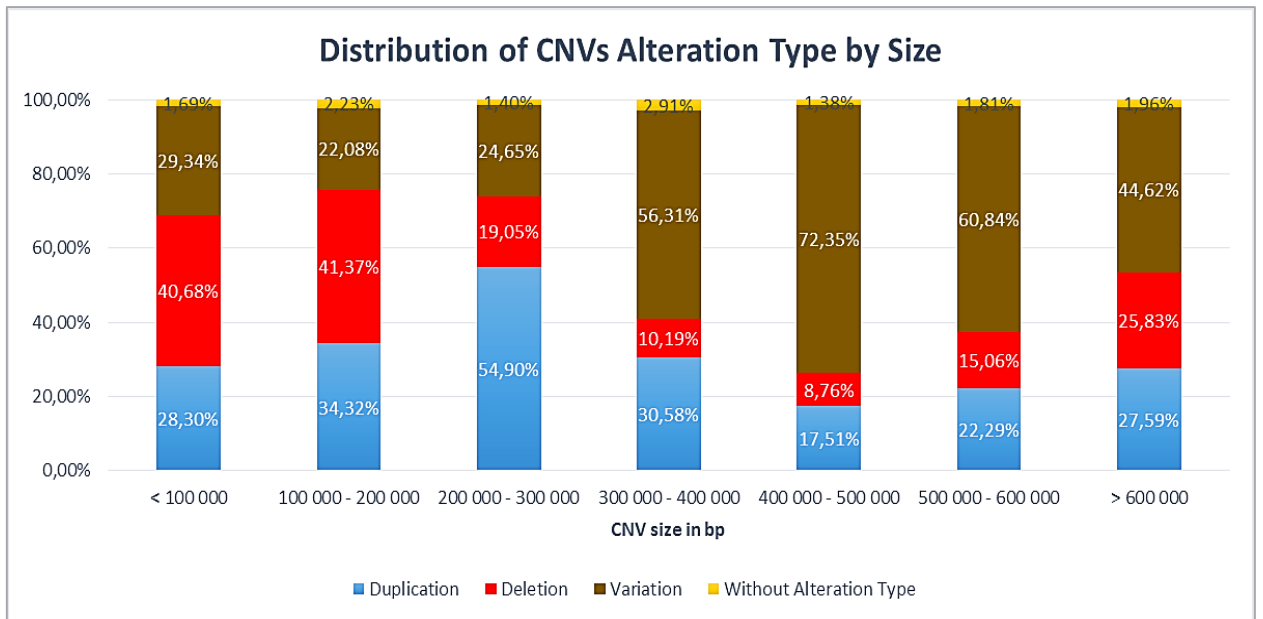


Figure 46 –Distribution of CNVs alteration Type by CNV size. Percentages are related to the number of CNVs shown in Figure 44 for each size interval.

5.5.4. Distribution of CNVs per chromosome and Distribution of CNVs alteration type per chromosome

Figure 47 shows the number of CNVs per chromosome and Figure 48 the distribution of CNVs alteration type per chromosome in arrayCGH DB.

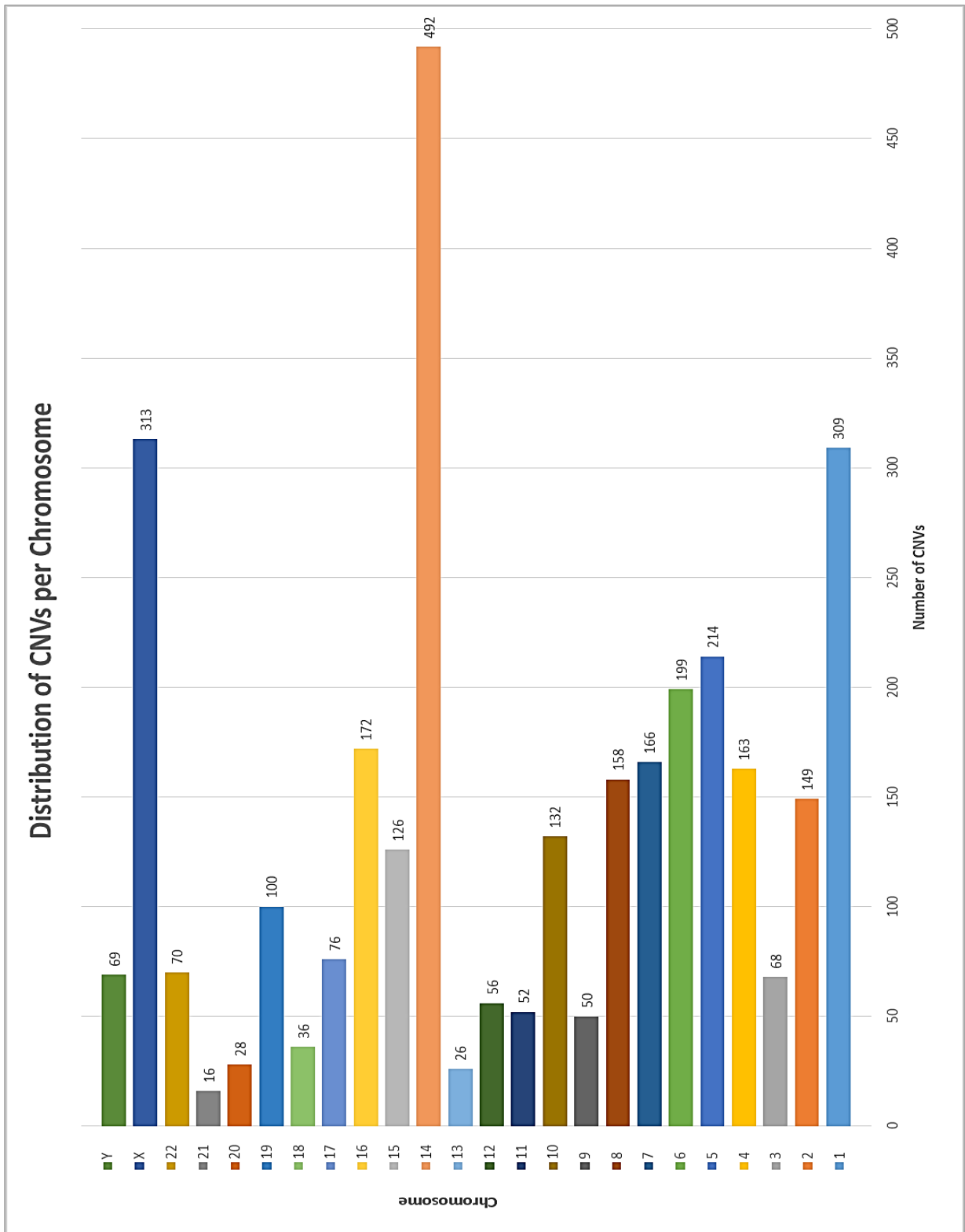


Figure 47 - Distribution of CNVs per Chromosome.

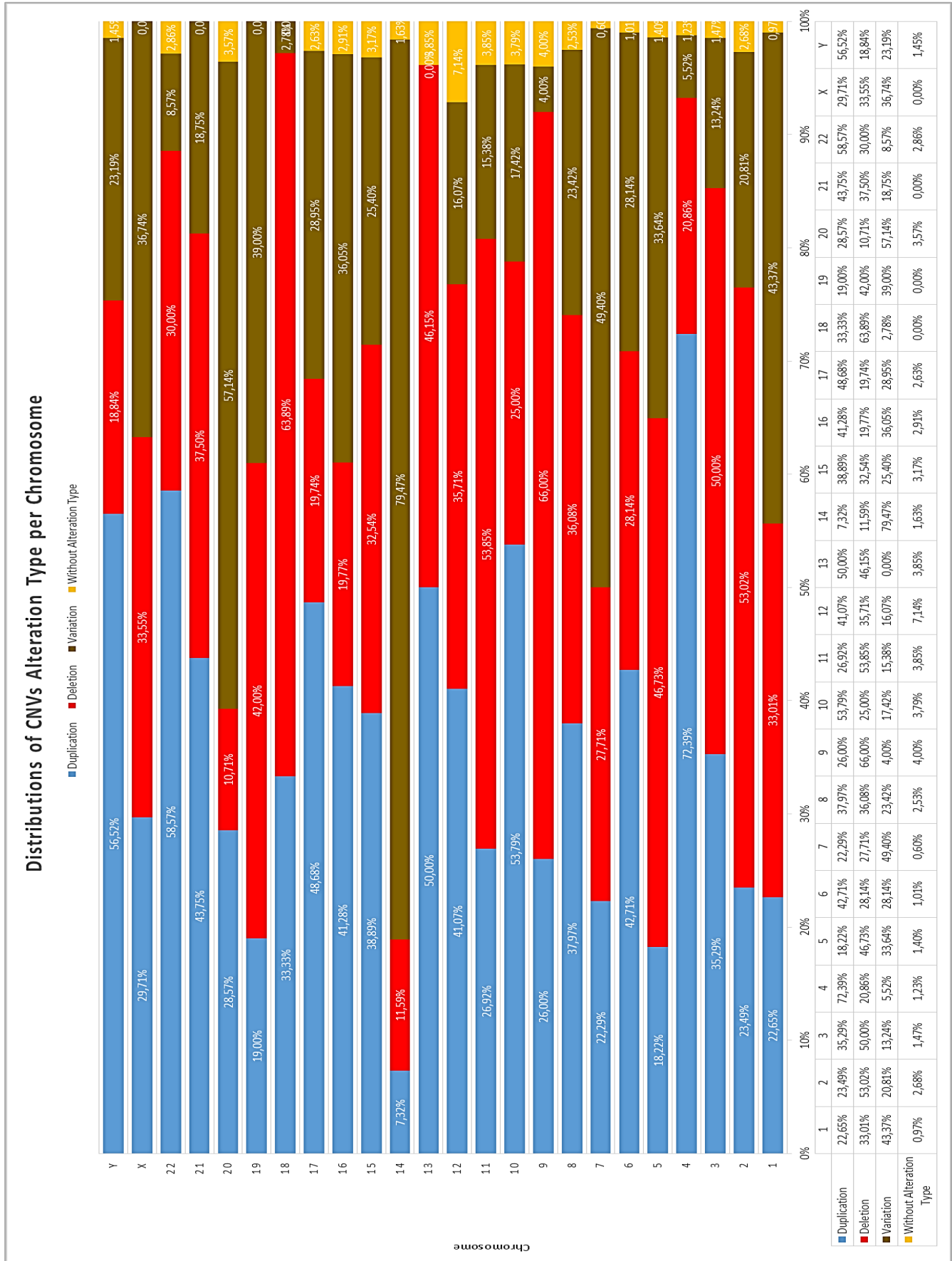


Figure 48 – Distribution of CNVs alteration type per chromosome in arrayCGH DB. Percentages are related to the number of CNVs shown in Figure 47 for each chromosome.

5.5.5. Distribution of CNVs class

Figure 49 shows the distribution of CNVs class.

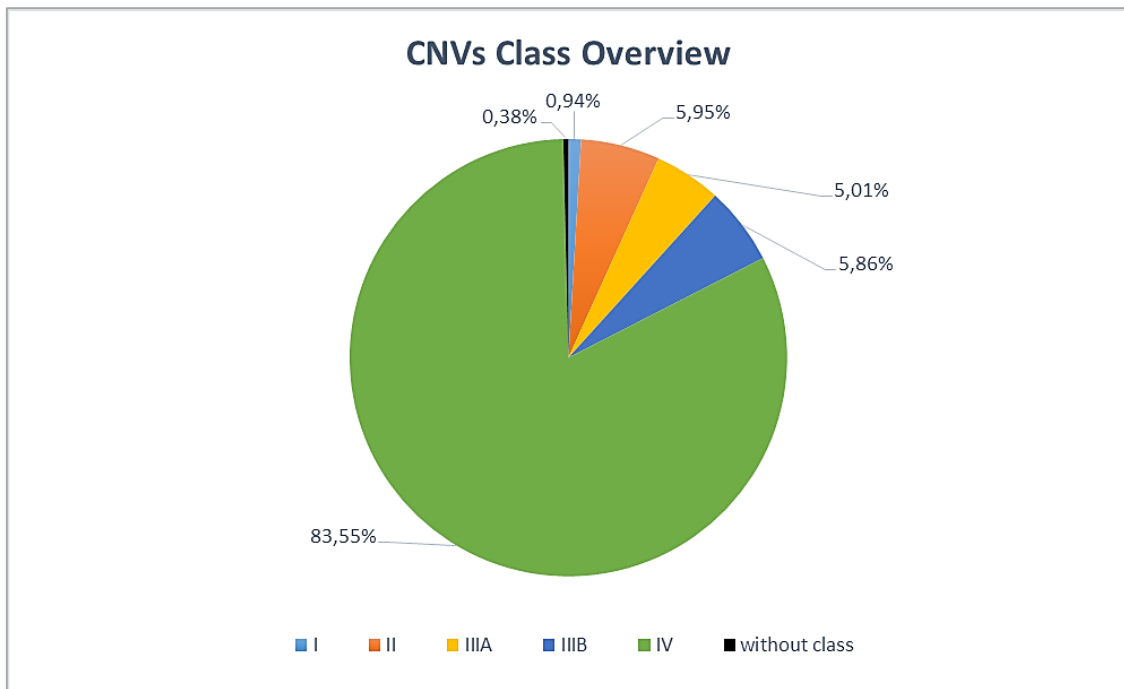


Figure 49 – Distribution of CNVs by class.

5.5.6. Distribution of CNVs class by size

Figure 50 shows the distribution of CNVs class by size.

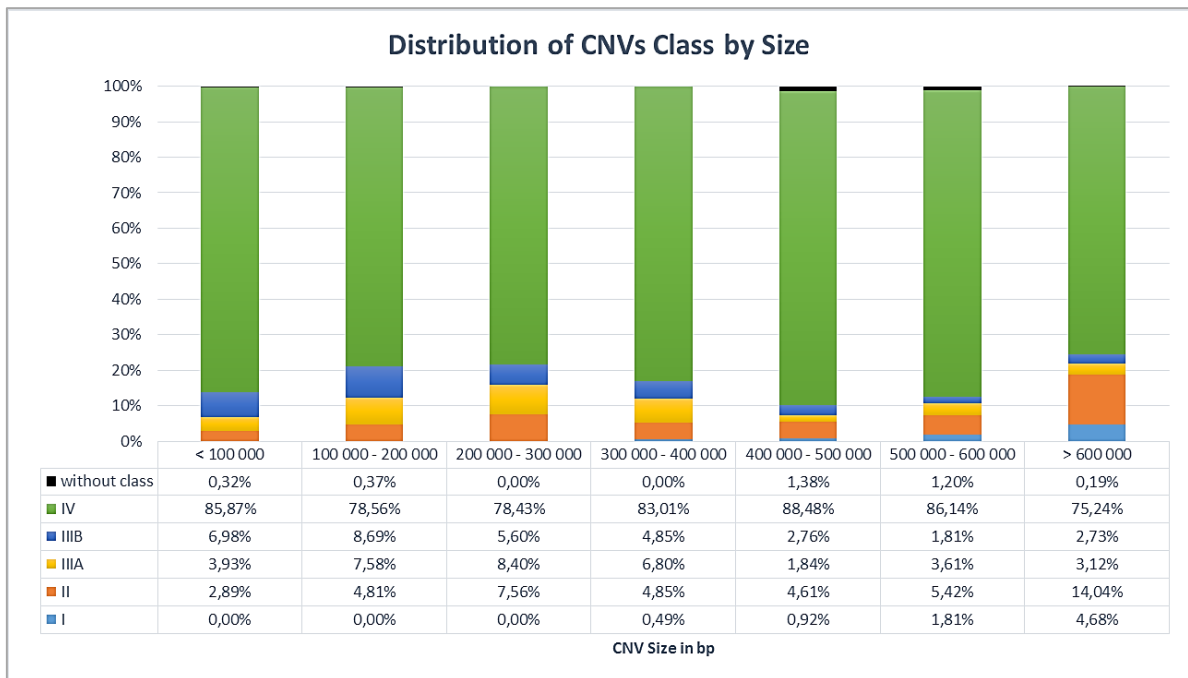


Figure 50 – Distribution of CNVs Class by size.

6. ArrayCGH DB Fixes

During validation and through some queries on the database, it was observed that some CNVs had wrong data, such as negative size, wrong class or alteration type. To fix them, clinic cases of CNVs with wrong data were identified in order to be able analyse the spreadsheets where they came from. After discovering what kind of issues were causing the wrong data, minor changes on arrayCGH DB were performed, wrong data was edited or deleted and re-uploaded.

6.1. Negative Size Fix

Nine CNVs with negative size were identified. This problem was caused when Clinic Cases with “Tabela Classes” format were uploaded with CNV Start greater than Stop. Now, when the upload of Clinic Cases is performed, it is checked if they have CNVs with Start greater than Stop, and if they have they are not stored on database, and Researchers are informed that the clinic case has wrong Start or Stop, causing negative size (Figure 51). The nine CNVs with negative size were fixed using the option edit CNV of arrayCGH DB after consulting the respective “IntervalBasedReport” spreadsheets, where the values were correct (Figures 52, 53, and 54).

6. ArrayCGH DB Fixes

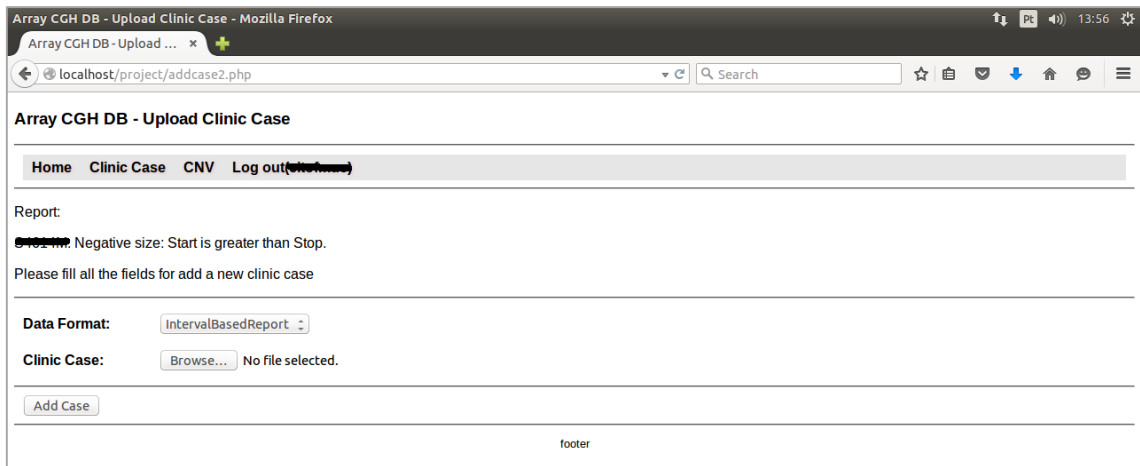


Figure 51 – Upload report of a Clinic Case with one CNV where Start is greater than Stop.

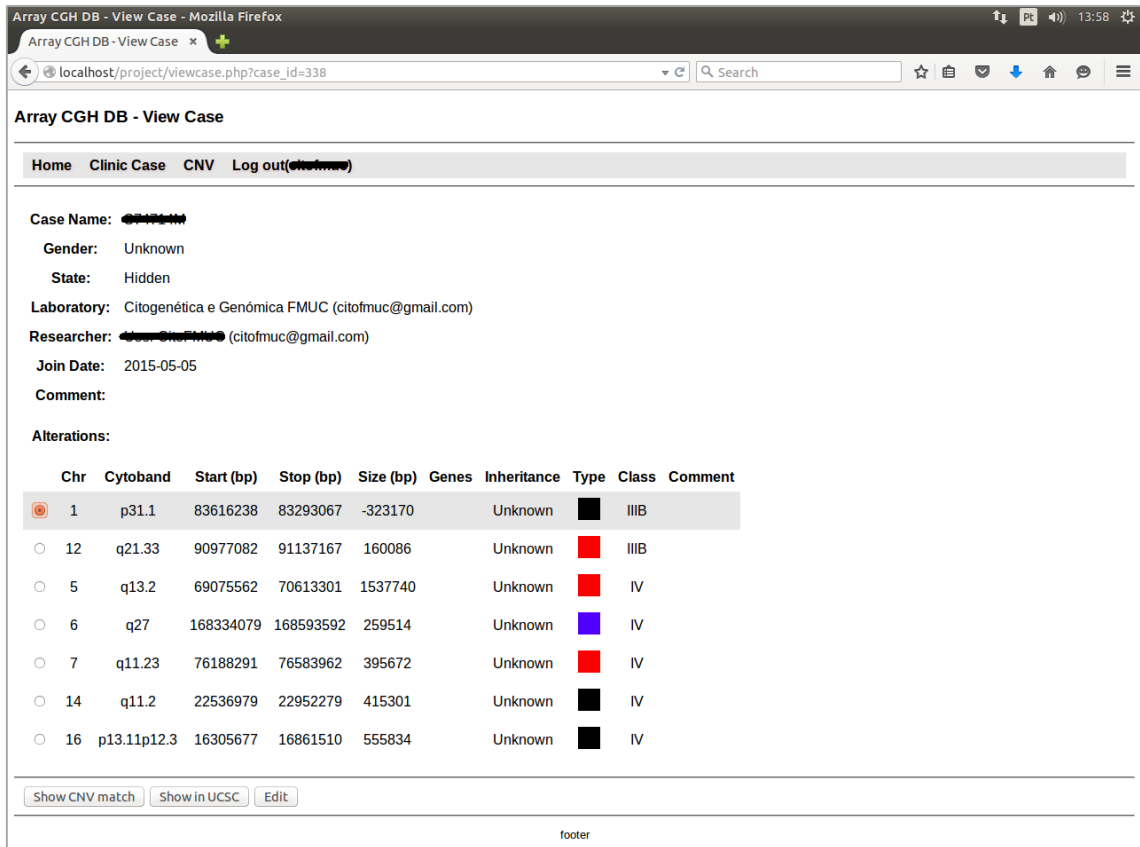


Figure 52 – Clinic Case where one CNV has a Start greater than Stop and consequently has negative size.

Array CGH DB - Edit CNV

Home Clinic Case CNV Log out (username)

Chromosome: 1

Cytoband: p31.1

Start: 83616238

Stop: 83923067

Genes:

Inheritance: Unknown

Type: Variation

Class: IIIB

Comment about this CNV

Comment:

edit

footer

Figure 53 – Editing of Stop from the wrong CNV of Figure 52, after consulting the respective “IntervalBasedReport” where the values were correct.

6. ArrayCGH DB Fixes

Array CGH DB - View Case

Home Clinic Case CNV Log out(~~XXXXXXXXXX~~)

Case Name: S74714M
Gender: Unknown
State: Hidden
Laboratory: Citogenética e Genómica FMUC (citofmuc@gmail.com)
Researcher: ~~XXXXXXXXXX~~ (citofmuc@gmail.com)
Join Date: 2015-05-05
Comment:

Alterations:

Chr	Cytoband	Start (bp)	Stop (bp)	Size (bp)	Genes	Inheritance	Type	Class	Comment
1	p31.1	83616238	83923067	306830		Unknown	■	IIIB	
12	q21.33	90977082	91137167	160086		Unknown	■	IIIB	
5	q13.2	69075562	70613301	1537740		Unknown	■	IV	
6	q27	168334079	168593592	259514		Unknown	■	IV	
7	q11.23	76188291	76583962	395672		Unknown	■	IV	
14	q11.2	22536979	22952279	415301		Unknown	■	IV	
16	p13.11p12.3	16305677	16861510	555834		Unknown	■	IV	

Show CNV match Show in UCSC Edit

footer

Figure 54 – Clinic Case of Figure 52 after being edited.

6.2. Class Fix

Seven CNVs were identified where class was misspelled, caused in Clinic Cases with “Tabela Classes” format upload, and five CNVs where the class is uncertain on the spreadsheets (double classification with question mark). In order to fix these problems during the upload of Clinic Cases, it is now verified if CNVs class matches the known types (“I”, “II”, “IIIA”, “IIIB” and “IV”) and if not, the clinic case with wrong class is not stored in the database and Researchers are informed that the clinic case has wrong class(es). (Figures 55, 56 and 57)

Join Date: 2015-05-05
 Comment:

Alterations:

Chr	Cytoband	Start (bp)	Stop (bp)	Size (bp)	Genes	Inheritance	Type	Class	Comment
<input type="radio"/>	2	q31.1	172751102	172945551	194450	Unknown	■	IIIA	
<input type="radio"/>	3	q25.1	151411145	151983625	572481	Unknown	■	sup	
<input type="radio"/>	5	q35.3	178288278	178367320	79043	Unknown	■	IIIA	
<input type="radio"/>	6	q24.2	144328244	144338632	10389	Unknown	■	sup	
<input type="radio"/>	12	p12.3	19484149	19942388	458240	Unknown	■	sup	
<input type="radio"/>	15	q26.2	96878073	96966138	88066	Unknown	■	Supr	
<input type="radio"/>	X	q26.3	134292347	134347917	55571	Unknown	■	IIIA	
<input type="radio"/>	1	q23.3	161537454	161618904	81451	Unknown	■	IV	
<input type="radio"/>	2	q37.3	242924058	242948040	23983	Unknown	■	IV	
<input type="radio"/>	4	p16.3	36424	68211	31788	Unknown	■	IV	
<input type="radio"/>	5	q13.2	69705562	70388844	683283	Unknown	■	IV	
<input type="radio"/>	6	q26	162636302	162799381	163080	Unknown	■	IV	
<input type="radio"/>	11	p15.4	5690064	5820979	130916	Unknown	■	IV	
<input type="radio"/>	14	q11.2	22272200	22952279	680080	Unknown	■	IV	
<input type="radio"/>	14	q32.33	107148739	107185512	36774	Unknown	■	IV	
<input type="radio"/>	20	q13.32	57464121	57467344	3224	Unknown	■	IV	

Show CNV match Show in UCSC Edit

Figure 55 – Clinic Case with several CNVs with wrong class (sup).

6. ArrayCGH DB Fixes

Array CGH DB - View Case

Home Clinic Case CNV Log out(**citofmuc**)

Case Name: **XXXXXXXXXX**
 Gender: Unknown
 State: Hidden
 Laboratory: Citogenética e Genómica FMUC (citofmuc@gmail.com)
 Researcher: **XXXXXXXXXX** (citofmuc@gmail.com)
 Join Date: 2015-05-05
 Comment:

Alterations:

Chr	Cytoband	Start (bp)	Stop (bp)	Size (bp)	Genes	Inheritance	Type	Class	Comment
2	q24.3	168220323	168357465	137143		Unknown	■	IIB	
3	q26.1	163309236	163824322	515087		Unknown	■	IIB	
9	p21.2	26301109	26473358	172250		Unknown	■	IIB	
13	q13.2	86225246	89316204	3090959		Unknown	■	IIB	
1	q21.2	149041013	149244438	203426		Unknown	■	IV	
5	q13.2	69705562	70404381	698820		Unknown	■	IV	
7	q34	142455543	142487154	31612		Unknown	■	IV	
14	q11.2	22578531	22897089	318559		Unknown	■	IV	
16	p13.11p12.3	16525289	16861510	336222		Unknown	■	IV	

Show CNV match Show in UCSC Edit

footer

Figure 56 – 4 CNVs with wrong class (IIB) caused by a misspelling on the uploaded spreadsheet.

Array CGH DB - Upload Clinic Case

Home Clinic Case CNV Log out(**citofmuc**)

Report:
XXXXXXXXXX One classification value is wrong.
XXXXXXXXXX One classification value is wrong.

Please fill all the fields for add a new clinic case

Data Format: IntervalBasedReport -
 Clinic Case: Browse... No file selected.

Add Case

footer

Figure 57 – Two Clinic Cases with wrong class which are not stored on ArrayCGH DB.

6.3. Alteration Type Fix

Fifty five CNVs were identified where the alteration type was “unknown”, which happened when Clinic Cases with “Tabela Classes” format were upload with misspelling on the alteration type, and three CNVs had an alteration type of “Triplificação”, which is a region not described on Table V. In order to fix these problems, now, when uploading Clinic Cases performed, it is verified if CNVs alteration type matches the known types (“Deleção”, “Duplicação”, “Amplificação”, “Triplificação” and “Variação”) and if not, the clinic case with wrong type is not stored in the database and Researchers are informed that the clinic case has wrong alteration type(s). The new alteration type, Triplification, was added and it is now recognized in all pages of arrayCGH DB (Figures 58 and 59).

Array CGH DB - View Case

Home Clinic Case CNV Log out(Admin)

Case Name: ██████████
 Gender: Unknown
 State: Hidden
 Laboratory: Citogenética e Genômica FMUC (citofmuc@gmail.com)
 Researcher: ██████████ (citofmuc@gmail.com)
 Join Date: 2015-04-29
 Comment:

Alterations:

Chr	Cytoband	Start (bp)	Stop (bp)	Size (bp)	Genes	Inheritance	Type	Class	Comment
7	p14.3	33135353	33171145	35793		Unknown	Unknown	II	
13	q21.31	65531359	65651922	120564		Unknown	Triplification	IV	
1	q31.3	196837584	196891668	54085		Unknown	Unknown	IV	
5	q13.2	69426248	70613301	1187054		Unknown	Unknown	IV	
7	q11.23	76136320	76583962	447643		Unknown	Unknown	IV	
15	q11.2	24384453	24470140	85688		Unknown	Unknown	IV	
16	p13.11p12.3	16544699	16878082	333384		Unknown	Unknown	IV	
16	p11.2p11.1	34482042	34727349	245308		Unknown	Unknown	IV	
18	p11.32	1932997	1980846	47850		Unknown	Unknown	IV	
19	p12	20591370	20701620	110251		Unknown	Unknown	IV	

Show CNV match Show in UCSC Edit

Figure 58 – Clinic Case with one CNV where the alteration type is Triplification.

6. ArrayCGH DB Fixes

Array CGH DB - CNV match - Mozilla Firefox

Array CGH DB - CNV ma... x

localhost/project/cnv.php?chr=20&start=29833386&stop=29920086&search=Search+by+position&ge Search

Array CGH DB - CNV match

Home Clinic Case CNV Log out(~~username~~)

Chr: 20

Start: 29833386

Stop: 29920086 Search by position

Gene: Search by gene

Results:

User track

○ *******

○ ***

○ ***

○ *******

○ *******

Start: 29833386 Stop: 29920086 Region type: Triplication

View Clinic Case Show in UCSC

footer

Figure 59 – CNV match where one CNV with alteration type of Triplication appears.

7. Overview, Conclusions and Future Perspectives

7.1. Overview

Microarray Comparative Genomic Hybridization (arrayCGH) allowed a significant advance on the diagnose of unexplained development diseases by detecting genomic copy number variations (CNVs) that were previously undetectable by other types of cytogenetic technologies. In order to determine if a CNV is pathogenic or not, it was necessary compare a CNV with other CNVs already classified as benign or pathogenic, which is performed, in most of the cases, by querying a database with CNVs records already analysed and classified. (Section 2. Background)

At *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra* are performed per year about 1000 arrayCGH tests and results are stored in spreadsheets. This kind of management of arrayCGH results do not provide an easy way to find the results they need to help in the decision about the pathogenicity of recent CNVs. This situation prompted them to look into a more efficient way to save and query their results, so they decided to request the development of an in-house relational Database (DB) to store their results and an application which allows them to feed and query the DB. (Section 2. Background)

Although researchers were aware that they need a database, they did not have more defined than the original request: “Develop and implement a relational database which should contain arrayCGH results from patients with cognitive deficit, and also to allow a set of queries defined by the laboratory clinicians in a way that can answer to questions or hypotheses related with new arrayCGH results”. So it was necessary to start a software development process. First, a set of software requirements was defined, using different techniques such as brainstorming, role playing and observation, which allowed to identify the real requirements and to add functionalities to the software that were not in the original first request. (Section 3.1. Software Requirements)

Afterwards, with the more complete requirements information, Use Case Diagrams (UCD) were defined in order to model the dynamic behaviour of the system and were developed an Entity-Relationship Model (ERM) and a Relational Model (RM), which helped to set tables, columns, columns data types and to map the relations between the tables. This allowed to construct a relational database to store all the information the software needed to fulfil the requirements. (Section 3.2. Software Design)

When the software design was finished, a LAMP (Linux, Apache, MySQL and PHP) server was set up which allows to analyse requests, store information and deliver responses according to the requests. This solution was chosen because it is a well-known and robust solution, and in addition, has the advantage of being freely available (open source). MySQL was used to create the tables and the relations, defined on RM, to store the information the software need to work, and PHP to create the scripts which make it possible to perform the functionalities defined through the requirements. Apache was responsible for receiving and delivering user requests. (Section 3.3. Server: LAMP technology)

Security was a requirement considered in all stages of Software Development. The access to data stored in the database is regulated by user-level permissions, and all information requests through arrayCGH DB are analysed to determine if the user has permission to access the information he is asking for. Deletes are on cascade, which is crucial to safeguard the data referential integrity and will prevent errors related to this subject. Some precautions were taken in order to avoid SQL injection; data validation processes are used through PHP functions and PHP MySQL functions in order to inhibit this kind of database attacks. By backing up the database daily it is possible to restore all the information stored in case of any system crash or corruption. (Section 4. Security)

After the software development, the database was populated with 3254 CNVs records from several Clinic Cases, uploaded through the Upload of Clinic Cases feature, which was developed with the propose of analysing spreadsheets with CNV information, and thus storing them in the ArrayCGH DB. Next, in order to know about the CNV data stored, several queries were made on the CNVs stored in database. This process allows to analyse the distribution of CNVs for several CNV attributes, such as alteration type,

size, class and chromosome, and in addition, it helps to identify some problems with the uploaded information. The identified problems were analysed and to fix them, some adjustments in the ArrayCGH DB were made. (Section 5. Results and Discussion and Section 6. ArrayCGH DB Fixes)

7.2. Conclusions and Future Perspectives

The arrayCGH DB provides an efficient way for uploading spreadsheets of arrayCGH results performed at *Laboratório de Citogenética e Genómica da Faculdade de Medicina de Coimbra*, and also performing queries of previously analysed CNVs, through a CNV matching executed with CNVs from arrayCGH DB and/or UCSC Genome Browser. Furthermore, the arrayCGH DB allows Researchers to keep track and edit their arrayCGH results and Administrators to share arrayCGH results with other laboratories and consequently with other Researchers. This makes arrayCGH DB a valuable tool in daily clinical diagnose for laboratories and clinicians using this technique. The data stored about CNVs could also be a starting point for new studies related with human unexplained development diseases that could bring new discoveries in this field.

In case a new laboratory requests the use of arrayCGH DB, it is necessary to analyse the arrayCGH reports of that laboratory in order to know if they fit in the upload formats available (“IntervalBasedReport” or “TabelaClasses”) and in the database model developed. If not, in the developed solution, it is possible to add a new format, reusing all the work done, and to upload the arrayCGH results from the new laboratory into the arrayCGH DB.

The greatest difficulty identified when developing this work was to gather the software requirements because laboratory researchers are not familiar with informatics terminology and this made difficult the discussion about some functionalities/behaviours of the software. In order to solve this problem were developed in a first step and shown some software functionalities to the researchers, which allowed them to visualize and understand some concepts difficult to explain, and thus get their feedback.

Although all the goals of the project have been accomplished with the ArrayCGH DB solution, it is possible to do more and better in future by code refactoring, improving or adding new functionalities and by giving to the ArrayCGH DB a better visual presentation. Shortly, code refactoring will allow to have a more robust code without any changes on the present functionalities, but will improve the code design making even easier to add new functionalities and manage the arrayCGH DB. There are a many new functionalities possible to add, for example, put a spreadsheet on the Clinic Case View page allowing Researchers to edit multiple CNVs of the same Clinic Case at the same time, create an export or print option for Clinic Cases providing a tool for Researchers to deliver arrayCGH reports through the arrayCGH DB, develop a more attractive and informative home page with some information about the project and with some CNV overview graphs, as the graphs used at section 5.5. Validation. However, the main task at this moment should be to populate the database with the Clinic Cases that are not yet uploaded, about 2500 Clinic Cases, obtain and analyse the users' feedback, and then discuss the future of the arrayCGH DB.

References

1. Miller, D., *et al.* (2010). Consensus Statement: Chromosomal Microarray is a First-Tier Clinical Diagnostic Test for Individuals with Development Disabilities or Congenital Anomalies. *The American Journal of Human Genetics*. ISSN: 86, 749-764.
2. Cheung, S., and Pursley A. (2011). Comparative Genomic Hybridization in the Study of Human Disease. eLS. JohnWiley&Sons,Ltd. DOI: 10.1002/9780470015902.a0005955.pub2
3. Hillman, S, *et al.* (2010) Additional information from array comparative genomic hybridization technology over conventional karyotyping in prenatal diagnosis: a systematic review and meta-analysis. *Wiley Online Library*. DOI: 10.1002/uog.7754.
4. Bejjani, B., and Lisa, S. (2006). Application of Array-Based Comparative Genomic Hybridization to Clinical Diagnostics. *Journal of Molecular Diagnostics*, Vol. 8, Nº. 5. DOI: 10.2353/jmoldx.2006.060029
5. Lapierre, J., and Tachdjian, G. Comparative Genomic Hybridization. John Wiley & Sons, Ltd.
6. Redon, R, Fitzgerald, T, and Carter, N. (2010) Comparative Genomic Hybridization: DNA labelling, hybridization and detection. DOI: 10.1007/978-1-59745-538-1_17.
7. Adapted from Agilent Oligonucleotide Array-Based CGH for Genomic DNA Analysis Protocol. Version 7.3. Agilent Technologies, Inc. 5301 Stevens Creek Blvd, Santa Clara, CA 95051 USA. March 2014. Manual Part Number G4410-90010.
8. Allemeersch, J., *et al.* (2009). An experimental loop design for the detection of constitutional chromosomal aberrations by array CGH. *BMC Bioinformatics*. DOI: 0.1186/1471-2105-10-380
9. Adapted from Pilonis, Dan and Milles, Russ – *Head First Software Development*. 1st edition. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc, 2008. ISBN 0-596-52735-8. Chapters 1 and 2.
- 10 Adapted from Hamilton, Kim and Milles, Russ – *Learning UML 2.0*. 1st edition. O'Reilly Media, Inc, 2006. ISBN 0-596-00982-8. Chapter 2.
11. Adapted from Teorey, Toby; Lightstone, Sam and Nadeu, Tom –

Database Modeling & Design: Logical Design. 4th edition. 500 Sansome Street, Suite 400, San Francisco, CA 94111: Multiscience Press, Inc, 2006. ISBN 0-12-685352-5. Chapter 2 and 4.

12. Adapted from Beighley, Lynn – Head First SQL. 1st editon. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O’Reilly Media, Inc, 2007. ISBN 0-596-52684-9. Chapter 7.

13. Adapted from Beighley, Lynn – Head First SQL. 1st edition. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O’Reilly Media, Inc, 2007. ISBN 0-596-52684-9. Chapter 4 and 7.

14. From Rosebrock, Eric and Filson Eric – Setting Up LAMP: Getting Linux, Apache, MySQL and PHP Working Together. SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. 2004. ISBN 0-7821-4337-7. Chapter 1.

15. Adapted from Beighley, Lynn and Morrison, Michael. – Head First PHP & MySQL. 1st edition. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O’Reilly Media, Inc, 2009. ISBN 978-0-596-00630-3. Chapter 6.

16. From Rosebrock, Eric and Filson Eric – Setting Up LAMP: Getting Linux, Apache, MySQL and PHP Working Together. SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. 2004. ISBN 0-7821-4337-7. Chapter 6.