

Universidade do Minho
Escola de Engenharia

Eduardo Sabina dos Santos Valente

Development of computational tools for the integrated analysis of DNA microarray data with applications in cancer research

The MAP-i Doctoral Program of the Universities of Minho, Aveiro and Porto





Universidade do Minho



April 2015



Universidade do Minho
Escola de Engenharia

Eduardo Sabina dos Santos Valente

Development of computational tools for the integrated analysis of DNA microarray data with applications in cancer research

The MAP-i Doctoral Program of the Universities of Minho, Aveiro and Porto



Universidade do Minho



supervisors:
Professor Doutor Miguel Francisco de Almeida Pereira da Rocha
Professor Doutor Rui Manuel Vieira Reis

April, 2015

DECLARAÇÃO

Nome

Eduardo Sabina dos Santos Valente

Endereço eletrónico: essvalente@gmail.com

Título da tese

Development of computational tools for the integrated analysis of DNA microarray data with applications in cancer research

Orientadores:

Professor Doutor Miguel Francisco de Almeida Pereira da Rocha

Professor Doutor Rui Manuel Vieira Reis

Ano de conclusão: 2015

Designação do Doutoramento:

Informática

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

Universidade do Minho, 29/04/2015

Assinatura: _____

STATEMENT OF INTEGRITY

I hereby declare having conducted my thesis with integrity. I confirm that I have not used plagiarism or any form of falsification of results in the process of the thesis elaboration. I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, _____

Full name:

_____

Signature: _____

Acknowledgements

I want to thank all the people who directly or indirectly contributed to this long doctoral program came to a good conclusion.

I want to thank especially my supervisor, Miguel Rocha, for once again accepting supervise my work, for his good guidance and readiness to help, without which would not have been possible to achieve this goal.

I also want to thank my joint supervisor, Rui Reis, the clarifications provided on the approach to the case study.

Thanks to my family, mother, brothers and sisters by encouraging spirit that ever undertaken.

I thank my cabinet colleague, Natércia Santos, for the companionship and friendship that has shown all over the years that this work take place.

To my wife, Sonia Marques, a big thank for the affection, love and understanding, extremely necessary during the implementation of such an extensive and arduous task.

Thank you all!

Development of computational tools for the integrated analysis of DNA microarray data with applications in cancer research

Abstract

DNA microarrays are one of the most used technologies for gene expression measurement. However, there are several distinct microarray platforms, from different manufacturers, each with its own measurement protocol, resulting in data that can hardly be compared or directly integrated. Data integration from multiple sources aims to improve the assertiveness of statistical tests and predictive modeling, reducing the data dimensionality problem.

This work intends to establish a basis for the integration of gene expression measurements from several manufacturers, a problem that can be addressed at different levels. The reannotation process, will be addressed, representing a cornerstone of multi-platform integration. The proposed approach is based on a reannotation from probesets to transcripts, preserving valuable information for further analysis. This work will provide both the methods for this integration process, as well as computational tools to achieve these goals.

World Health Organization ranks astrocytomas, brain tumors, in four grades, being the fourth grade the most aggressive. Glioblastoma, the fourth grade tumor, is one of the most severe human diseases that almost inevitability leads to death. Physicians address the classification in grades through direct inspection. In this context, there is a need for good automatic predictors of tumor grade, which are not affected by human misclassification errors and that can be made with less invasive diagnostic tools.

Gene expression data from glial tumors will be used as case studies, considering data from Agilent, Affymetrix and Illumina platforms. This data will be used to perform the

stages involved in the process of selecting a good tumor grade predictor. In this work, the information integration from heterogeneous platforms is highlighted, evidencing the particularities of choosing approaches working at gene, transcript or probeset levels. Distinct machine learning algorithms and integration methods are tested, analyzing their ability to produce a good set of predictors for tumor grade.

Keywords

Microarray probe reannotation; gene differential expression; mRNA sequence alignment; gene expression; glioblastoma; multiplatform microarrays data integration; filter methods; wrapper methods.

Desenvolvimento de ferramentas computacionais para a análise integrada de dados de microarrays de ADN com aplicações no estudo do cancro

Resumo

Os microarrays de ADN são uma das tecnologias mais utilizadas para a medição de expressão de genes. No entanto, existem várias plataformas distintas, de diferentes fabricantes, cada um com seu próprio protocolo de medição, resultando em dados que não podem ser diretamente comparados ou integrados. A integração de dados de múltiplas fontes visa melhorar a assertividade de testes estatísticos e modelação preditiva, reduzindo o problema da dimensionalidade dos dados.

Este trabalho visa estabelecer uma base para a integração de medições de expressão de genes de vários fabricantes, um problema que tem diferentes níveis de abordagem. Será abordado o processo de re-anotação, uma pedra angular da integração multiplataforma. A abordagem proposta baseia-se na re-anotação de sondas para transcritos, preservando informação valiosa para análise posterior.

A Organização Mundial da Saúde classifica os astrocitomas, tumores cerebrais, em quatro graus, sendo o grau quatro o mais agressivo. O glioblastoma, um astrocitoma de grau quatro, é uma das mais graves doenças humanas que quase inevitabilidade conduz à morte. Os médicos efetuam a classificação do tumor através de inspeção direta com métodos invasivos ou quando o paciente já está morto. Existe uma necessidade de bons indicadores de grau do tumor, que não sejam afetados por erro humano de classificação e que possam ser feitos com ferramentas de diagnóstico menos invasivas.

Serão utilizados dados de expressão de genes de células da glia como caso de estudo, considerando-se dados de plataformas Agilent, Affymetrix e Illumina. Estes dados serão utilizados para executar as etapas envolvidas no processo de seleção de um bom modelo para previsão do grau do tumor. Neste trabalho, é destacada a integração de informações

a partir de plataformas heterogéneas, evidenciando as particularidades de escolher abordagens a nível de genes, transcritos ou sondas. Serão testados diferentes algoritmos de aprendizagem de máquina e diferentes métodos de integração, analisando a sua capacidade de produzir um bom conjunto de modelos para prever o grau do tumor.

Palavras-chave

Re-anotação de sondas de Microarrays; expressão diferencial de genes; alinhamento de sequências de mRNA; expressão de genes; glioblastoma; integração multiplataforma de dados de microarrays; métodos de filtragem; métodos *wrapper*.

Contents

1 INTRODUCTION 1

1.1 CONTEXT AND MOTIVATION 2

1.2 OBJECTIVES 3

1.3 THESIS STRUCTURE 4

2 GENE EXPRESSION: BIOLOGICAL PROCESSES, TECHNIQUES AND DATA ANALYSIS 7

2.1 BIOLOGICAL PROCESSES OF GENE EXPRESSION 7

2.1.1 The genome and the genes 7

2.1.2 Transcription 8

2.1.3 Splicing and alternative splicing 9

2.1.4 Translation 10

2.2 DNA MICROARRAYS 10

2.2.1 Fundaments of Microarrays 12

2.2.2 Normalization 13

2.2.3 Probe design 18

2.2.4 Microarray Data Levels and Representation 20

2.3 STATISTICAL ANALYSIS OF GENE EXPRESSION DATA 21

2.3.1 Hypothesis tests and p-value 22

2.3.2 Parametric Tests 22

2.3.3 Non-parametric statistics 24

2.3.4 ANOVA and general linear models 24

2.4 UNSUPERVISED LEARNING 26

2.4.1 Clustering 26

2.4.2 Principal component analysis 28

2.5 SUPERVISED LEARNING 28

2.5.1 Classification in gene expression context *29*
2.5.2 Classification algorithms *30*
2.5.3 Feature selection *33*
2.5.4 Classifiers combinations *40*
2.5.5 Validation *42*
2.6 GENE EXPRESSION DATA INTEGRATION 43
2.6.1 Previous works *44*
3 METHODOLOGY 49
3.1 SUMMARIZATION OF THE PROBLEM 49
3.2 REANNOTATION 50
3.2.1 Data collection *51*
3.2.2 Mapping *51*
3.2.3 Clone treatment *53*
3.2.4 Steps of the reannotation process *54*
3.3 DATA INTEGRATION 55
3.4 VALIDATION WITH SUPERVISED LEARNING 57
3.4.1 Filtering *58*
3.4.2 Wrapper *58*
4 APPLICATION TO GLIOMAS GENE EXPRESSION DATA 61
4.1 CASE STUDY 61
4.1.1 Gene expression on open-access databases *63*
4.2 DATA COLLECTION AND PRE-PROCESSING 66
4.2.1 Gene Expression Omnibus *67*
4.2.2 Rembrandt *74*
4.2.3 TCGA *75*
4.2.4 Probes sequences *77*
4.3 REANNOTATION 78
4.3.1 Sequence alignment for all probes *81*
4.3.2 Genome annotation *83*
4.3.3 Mapping *84*
4.3.4 Reannotation analysis *85*

4.4 CLASSIFICATION AND FEATURE SELECTION 88

5 SOFTWARE DEVELOPMENT 93

5.1 REQUIREMENTS 93

5.2 ARCHITECTURE 94

5.3 TOOLS AND RESOURCES 95

5.3.1 R system and packages 97

5.4 DATA COLLECTION AND REPRESENTATION 99

5.4.1 GEO simple datasets 100

5.4.2 GEO complementary datasets 103

5.4.3 GEO series 103

5.4.4 TCGA 104

5.4.5 Rembrandt 107

5.5 REANNOTATION 108

5.5.1 Probe sequences collection 108

5.5.2 Probes matches 109

5.5.3 Genes annotation 110

5.5.4 Mapping probes to genes 111

5.5.5 Mapping probesets to genes 111

5.5.6 Exons Annotation 112

5.5.7 Mapping probes to exons 113

5.5.8 Transcripts annotation 113

5.5.9 Mapping exons to transcripts 114

5.5.10 Mapping probesets to transcripts 114

5.5.11 Rebuild expression datasets 115

5.5.12 The reannotation Enhanced Entity-Relationship diagram 116

5.6 DATA INTEGRATION 116

5.7 FEATURE SELECTION AND CLASSIFICATION 117

6 CONCLUSIONS 121

List of figures

Figure 1. Illustration of the transcription process. 9
Figure 2. Illustration of alternative splicing. 9
Figure 3. Illustration of the translation process. 10
Figure 4. Microarray experiment with a) two-channels; b) one channel..... 12
Figure 5. Affymetrix probeset structure. 15
Figure 6. Agilent and Affymetrix probesets target sequence zones for gene ZFYVE20. The black line represents the extension of the gene and the blue lines represent the exons. In red, are the Agilent probesets and in purple are the Affymetrix probesets. The figure also includes the list of transcripts and respective numbered exons that comprise each transcript. 19
Figure 7. Generic structure of a decision tree. 30
Figure 8. Macroscopic pathology of Glioblastoma multiform (Source http://it.wikipedia.org/wiki/ Glioblastoma; April 2015)..... 62
Figure 9. Example of a TCGA aliquot barcode where it is indicated the meaning of each section..... 65
Figure 10. Affymetrix probesets relation between platforms. 70
Figure 11. Expression/grade distribution for Affymetrix probeset 201292_at. 89
Figure 12. IGEIS general architecture 94
Figure 13. Example of a section of an expression data frame. 100
Figure 14. Example of a section of a phenotype data frame. 101
Figure 15. Structure of TCGA expression folders. 104
Figure 16. TCGA phenotype folders. 105
Figure 17. TCGA Clinical file example, with two entries..... 106
Figure 18. Database table to store vendor probes sequences. 109
Figure 19. UCSC Human Genome, version hg19. 109
Figure 20. Database table to store probes matches. 110

List of figures

Figure 21. Database table to store genes annotation. 111
Figure 22. Database table with probes to genes mapping. 111
Figure 23. Database table with probesets to genes mapping. 112
Figure 24. Database table to store exons annotation. 113
Figure 25. Database table with probes to exons mapping. 113
Figure 26. Database table to store transcripts annotation. 114
Figure 27. Database table with exons to transcripts mapping. 114
Figure 28. Database table with probesets to transcripts mapping. 115
Figure 29. The reannotation Enhanced Entity-Relationship diagram. 116
Figure 30. Example of the structure of a machine learning dataset. 118

List of tables

Table 1. Phenotype data transformation for dataset GDS2853. 67
Table 2. Phenotype data transformation for samples in GSE3185 series that are not on GDS2853. 68
Table 3. Phenotype data transformation for dataset GDS3069. 69
Table 4. Phenotype data transformation for dataset GDS1962. 69
Table 5. Phenotype data transformation for datasets GDS1815 and 1816. 71
Table 6. Phenotype data transformation for datasets GDS1975 and GDS1976. 73
Table 7. Phenotype data transformation for series GSE19114. 73
Table 8. Phenotype data transformation for Rembrandt. 74
Table 9. Phenotype data transformation for lower grade gliomas on TCGA. 75
Table 10. Glioblastoma phenotypic data transformation on TCGA for Affymetrix HG-U133A dataset. 76
Table 11. Glioblastoma phenotypic data transformation on TCGA for Agilent G4502A_07_1 dataset. 76
Table 12. Glioblastoma phenotypic data transformation on TCGA for Agilent G4502A_07_2 dataset. 77
Table 13. ZFYVE20 transcripts identified in Ensembl. 79
Table 14. Number of genome matches per probe. 82
Table 15. Some Probesets Annotation by GeneAnnot. 83
Table 16. Differential expression for gene HOXD1 probesets. 85
Table 17. Transcript-oriented differential expression results for ZFYVE20. 86
Table 18. Gene-oriented differential expression results for ZFYVE20. 87
Table 19. Probeset-oriented differential expression results for ZFYVE20. 87
Table 20. Differential expression, by probeset, considering all contrasts in Affymetrix GDS1962, for gene KITLG. 87

List of tables

Table 21. Differential expression, by probeset, considering all contrasts in Agilent G4502A_07_2 and _3, for gene KITLG. 87
Table 22. Differential expression, by transcript, considering all contrasts in Affymetrix GDS1962, for gene KITLG. 88
Table 23. Differential expression, by transcript, considering all contrasts in Agilent G4502A_07_2 and _3, for gene KITLG. 88
Table 24. Results for classifiers (columns), filtering methods (rows) at the probeset level. #P - number of probesets; ER - Error Rate. 90
Table 25. Results for classifiers (columns), filtering methods (rows) at the gene level. #G - number of genes; ER - Error Rate. 90
Table 26. Results for classifiers (columns), filtering methods (rows) at the transcript level. #T - number of transcripts; ER - Error Rate. 90
Table 27. LOOCV error rate for Affymetrix and Agilent data integration. 91
Table 28. Confusion matrices for prediction of the new dataset. 92
Table 29. Attributes table for GDS2853. 102
Table 30. Data table for GDS2853. 102
Table 31. All possible values for sample field on a aliquot barcode and the corresponding phenotype meaning. 106

Acronyms and abbreviations

Affy	Affymetrix
Agil	Agilent
AIC	Akaike information criterion
ANN	Artificial Neural Network
ANOVA	Analysis of variance
BC	Before Christ
BLAST	Basic Local Alignment Search Tool
BLAT	BLAST - Like Alignment Tool
BRB	Biometric Research Branch
BTIC	Brain Tumor Initiating Cells
cDNA	Complementary DNA
ComBat	Combining Batches
DNA	Deoxyribonucleic Acid
eBayes	Empirical Bayes
GA	Genetic Algorithm
GBM	Glioblastoma
GDS	GEO Dataset
GEO	Gene Expression Omnibus
GPL	GEO Platform
GSE	GEO Series
GSM	GEO Sample
HGG	High Grade Glioma
IDF	Improved Decision Forest
IGEIS	Integrated Gene Expression Information System
kNN	k-Nearest-Neighbours
LDA	Linear Discriminant Analysis
LGG	Lower Grade Glioma
LOESS	Locally weighted scatter plot smooth
LOOCV	Leave-One-Out Cross Validation
MIAME	Minimum information about a microarray experiment
MAS	MicroArray Suite

Acronyms and abbreviations

mRNA	Messenger RNA
NB	Naive Bayes
NCBI	National Center for Biotechnology Information
NCI	National Cancer Institute
NHGRI	National Human Genome Research Institute
PCA	Principal Component Analysis
PCA-CG	Principal Component Analysis - Conjugate Gradient
Rembrandt	REpository for Molecular BRAin Neoplasia DaTa
rfSet	Reduced feature set
RMA	Robust Multiarray Analysis
RNA	Ribonucleic Acid
SCF	Stem cell factor
SVM	Support Vector Machine
TCGA	The Cancer Genome Atlas
tRNA	Transfer RNA
UTR	Untranslated Region
UUID	Universally Unique Identifiers
WHO	World Health Organization

1 Introduction

Diagnosis can be defined as the classification of a disease by its symptoms or as a description of all these symptoms. Anyone can make a diagnosis, through their senses, by analyzing if the physiological state of an individual falls outside of what is considered to be the normal pattern.

The oldest registers of diagnosis known date back to the twenty seventh century BC, were written in Egypt, in papyrus, where it is possible to find the description of various diseases, including cancer [1]. The critical, logical and rational analysis using empirical methods in the diagnosis of diseases appears in Babylonian records of the eleventh century BC [2]. It is important to note that, even nowadays, empirical observation methods are still used in Medicine for diagnosis.

From Ancient Greece, the earliest references to the association of certain diseases with heredity are known, through works such as that of Hippocrates (460-370 BC), considered to be the father of modern medicine [3]. However, the basic heredity laws were established several centuries later by Gregor Mendel (1822-1884), through his study of the pea *Pisum sativum* [4].

Other branches of science have brought significant technological contributions to Medicine, increasing the accuracy and efficiency of traditional diagnosis methods and suggesting new ones. Among these contributions, a highlight must be done to the invention of the microscope, by Galileo Galilei (1564-1642), who came to open up a whole new perspective of the existence of 'universes within the universe' [5]. This, together with other technological advances, have contributed to the discovery of the structure of DNA in 1953 by Watson and Crick [6]. In the 1970s, Maxam and Gilbert proposed a method for sequencing DNA molecules [7]. The method that prevailed was, however, the one proposed by Frederick Sanger in 1975, which is the basis for the majority of methods still in use today. It involves a reaction where chain-terminator nucleotides are labeled with fluorescent dyes, combined with fragmented DNA, DNA

sequencing primers and DNA polymerase. Each nucleotide in the DNA sequence is labeled with a different dye color and a chromatogram is produced, with each color representing a different letter in the DNA code – A, T, C, or G [8].

DNA sequencing has brought a whole new range of possibilities in the diagnosis of diseases, especially those related to hereditary factors. In this regard, new techniques were developed to allow reading DNA sequences, that were increasingly accessible to the scientific community and companies on this knowledge area.

In a related area, Edwin Southern developed the method of southern blot for the analysis of DNA, which has lead to the northern blot for RNA analysis and western blot for proteins. This method searches a specific DNA sequence in a sample, using a specific signature probe, which hybridizes with its complementary DNA section, indicating that the genetic information exists in the sample. It can, therefore, be used to count the number of copies of a gene in the genome [9]. This was the basis for the appearance of the microarrays technology in 1995, by the Oxford Gene Technology company, which largely increased the number of genes that can be analyzed at the same time, providing a genome cover in almost its entire length [10].

1.1 Context and motivation

The association of phenotypes to gene expression profiles suffered a great boost with the advent of microarray technology. With fewer investment, it becomes possible to make a comprehensive analysis of the expression behavior of all known genes in a single experiment. Since the technology was created, several studies have been conducted that use microarrays. Over these years, several patients were giving permission to use their clinical data for investigation. These data ware being collected in public databases, reaching a considerable number of experiments recorded. For example, Gene Expression Omnibus currently contains 56755 series, which include 1382178 samples, from several species [11].

Some projects, related or not with gene expression databases, provide built in tools to analyze gene expression, but these are limited to one dataset at a time. The problem is that the majority of datasets are poor in the number of samples, opening a big gap on the representativeness of any study intended to achieve a universal conclusion.

Therefore, it is needed to gather datasets from different sources, integrate them on richer datasets, to improve the strength of the statistical studies results over a specific subject. The degree of complexity of this task extends to various levels, since data can be in different databases, originate from different manufacturers or simply different platforms from the same manufacturer. Each platform has its own way of disposing the information and its own standards. To be able to do such a comprehensive integration, the requirements are immense, and require a thorough and detailed analysis of the problem.

1.2 Objectives

In this work, the main aim is to develop computational tools for gene expression data integration, able to process and integrate datasets from multiple sources, but also to run machine learning algorithms and construct analyses tools to respond to specific research questions, related to disease diagnosis. The system will be validated by a case study, related to the integration and analysis of data from glioblastoma studies.

In this context, the proposed work will have the following scientific/ technological objectives:

- To develop methodologies and computational tools that allow the integration of the distinct types of gene expression data, that are relevant for a given problem, including the ones coming from the several experimental platforms and available Bioinformatics databases. It should be possible to create an integrated data repository, able to answer the questions relevant in data analysis/ data mining;
- To develop methods and tools for the analysis and mining of gene expression data able to deal with the different types of relevant information and to create models that integrate the various types of available information. The final goal will be to create classification models from the data that can be applied to new cases. Previous efforts on the development of similar methods will be taken into account;

- To develop optimization methods able to efficiently identify good sets of biomarkers, coming from the various sources of available information that can be used in the previous classification models;
- To apply the developed methods and informatics tools in a case study related to glioblastoma.

1.3 Thesis structure

This thesis is organized in six chapters, being this the first, where a general overview of the work is done. The other chapters are the following:

- Chapter 2 - **Gene expression: biological processes, techniques and data analysis**:

  In this chapter, the relevant cellular processes are presented, including the structure and function of genomes and the fundamentals of gene expression and experimental techniques for their measurement. A review on the most common statistical methods used on gene expression data analysis and data mining is provided, addressing subjects like data normalization, feature selection and classification.

- Chapter 3 - **Methodology**:

  In this chapter, the principal guidelines to be adopted on the construction of the required system are defined. Some algorithms and the mathematical considerations for some tasks are put forward.

  The methodology is presented, being defined the different stages of the process, to provide a better understanding of the context.

- Chapter 4 - **Application to gliomas gene expression data**:

  In this chapter, the application of the proposed methodology to a specific case study is presented, related to astrocytoma WHO grade classification. A brief description of the case study is made.

An introduction to the gene expression databases used is provided, pointing their methods to access data and the structure of that data.

Each section of the integration system is highlighted, providing the results obtained at each stage.

- Chapter 5 - **Software development**:

  This chapter provides a description of the software architecture, at the database level and at the 'business' level and their interactions. The tools and programming languages used in this work are described, while some software packages utilized also are mentioned.

  A reference to the principal functions created at each stage of software development is made.

- Chapter 6 - **Conclusions**:

  In this chapter, a critical analysis of the developed work is made, stressing the achieved objectives, possible alternatives to the options adopted and future work possible to be done based on what was implemented.

2 Gene expression: biological processes, techniques and data analysis

In this chapter, the relevant cellular processes will be presented, including the structure and function of genomes and the fundamentals of gene expression and experimental techniques for their measurement (e.g. DNA microarrays). A review will be provided on the most common statistical methods used on gene expression data analysis and data mining, addressing subjects like data normalization, feature selection and classification.

2.1 Biological processes of gene expression

In this section, the biological aspects with relevance for this work will be described. Some specific details are purposefully omitted, to facilitate the understanding of the big picture.

2.1.1 The genome and the genes

On eukaryotic cells, most of the genetic material (genome) is contained on the cell nucleus. The genome is the structural and functional model of living beings. It consists of a set of double-stranded helical chains of deoxyribonucleic acid (DNA), that carry information in a sequence of nitrogenous base pairs (A-adenine, G-guanine, C-cytosine, T-thymine). Each base on one side of the helix binds to its complementary on the other chain, forming links that look like stairs.

Genes are sections of the genome carrying all the necessary information to encode a protein (or several). Genes have been classically associated to particular phenotype

attributes of the individual (eye color, members size, etc.) or molecular functions. However, the gene function influence is in the majority of the cases not visible, being associated with non-visible functions of the cells. The number of nitrogenous bases pairs that form a gene is variable, but the lowest possible sequence capable of carrying information is composed by three bases, called a codon [12], which encode an amino acid.

Not all genetic materials are genes. In fact, less than 2% of the genome codes proteins. The rest of the DNA sequence, sometimes mistakenly called junk DNA, intervenes in other functions like signaling the start and end of genes and controlling gene expression levels. Even inside an eukaryotic gene, there are coding segments, called exons, and non-coding segments, called introns [12].

The fact that an individual possesses a certain gene does not imply that the feature associated to it will manifest. The function of a gene depends on its expression, which partially explains why cells with the same DNA give place to different tissues. Gene expression is the process by which a protein is produced from its encoding gene, and can by divided into the phases of transcription, splicing and translation, which are addressed in the next sections.

2.1.2 Transcription

The transcription process, which occurs in the eukaryote cell's nucleus, starts with the identification of the section of the genome corresponding to the gene that will be expressed. By action of specific enzymes, the double chain is split into two strands. One of the strands of the DNA is then transcribed, producing a section called precursor RNA[1] (pre-mRNA), which is complementary to the template DNA. The transcription is made on the 3' → 5' direction, on the antisense strand of the gene. The sense strand is also able to originate expression and both strands must be considered as potential expression generators [13][14]. After transcription, the pre-mRNA is released and the DNA helix connects back again, reconstructing the original structure (Figure 1).

[1] A ribonucleic acid (RNA) sequence is composed by A, G, C and U-uracil bases.

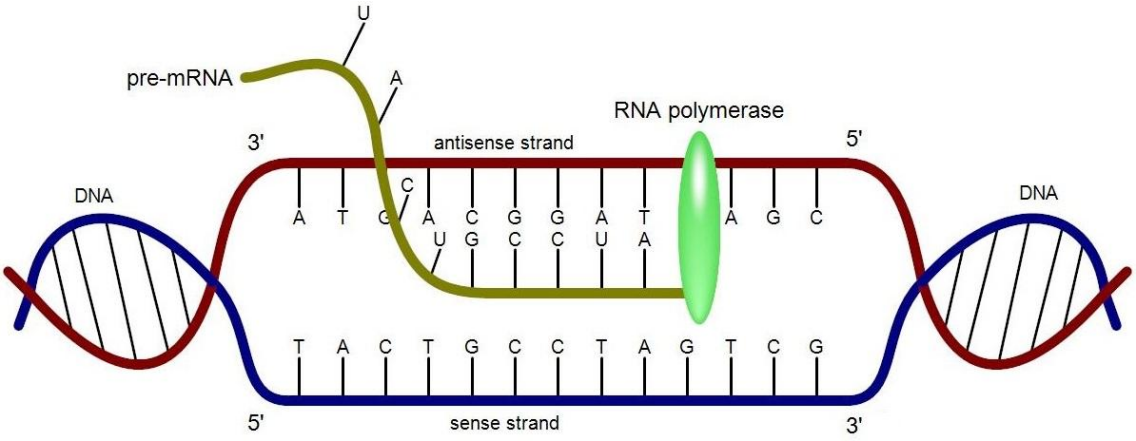


Figure 1. Illustration of the transcription process.

2.1.3 Splicing and alternative splicing

The pre-mRNA sequence is composed of regions called exons and introns in an interleaved manner. Only exons are used for protein coding so, on the splicing stage, introns are removed from the sequence, and the exons are merged into a new chain, the mature mRNA. A single gene can contain numerous exons and the splicing process can combine them in various ways. This is known as alternative splicing and occurs in about 60% of human genes [15]. Each combination of exons necessary for the production of a particular protein is a transcript variant of the respective gene. Thus, the same gene can code multiple isoforms of protein molecules with different functions [16].

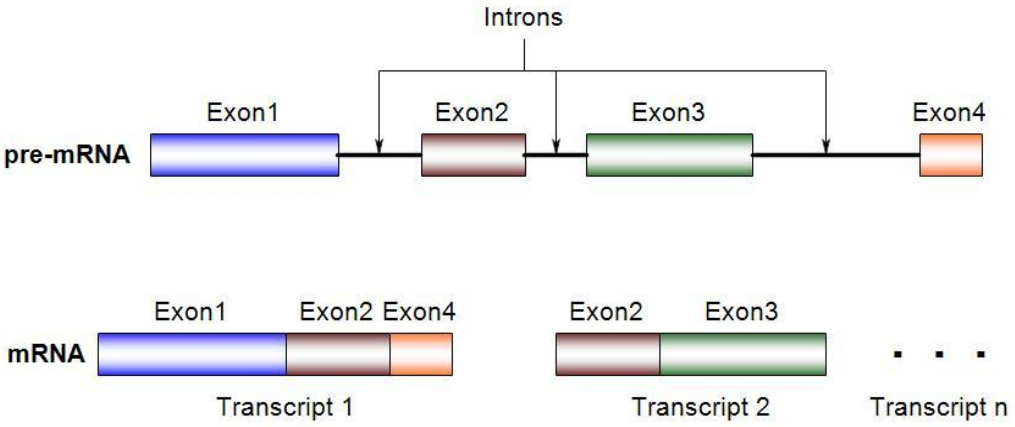


Figure 2. Illustration of alternative splicing.

2.1.4 Translation

After the process of splicing, mRNA leaves the cell nucleus, and attaches to a ribosome in the cytoplasm, where translation is performed. When a section of mRNA comes to a ribosome, it is translated into a chain of amino acids that will form, upon completion of the translation, a full protein (Figure 3). Each codon on mRNA couples with the transfer RNA (tRNA) that carries its anticodon, which specifies the amino acid to be produced.

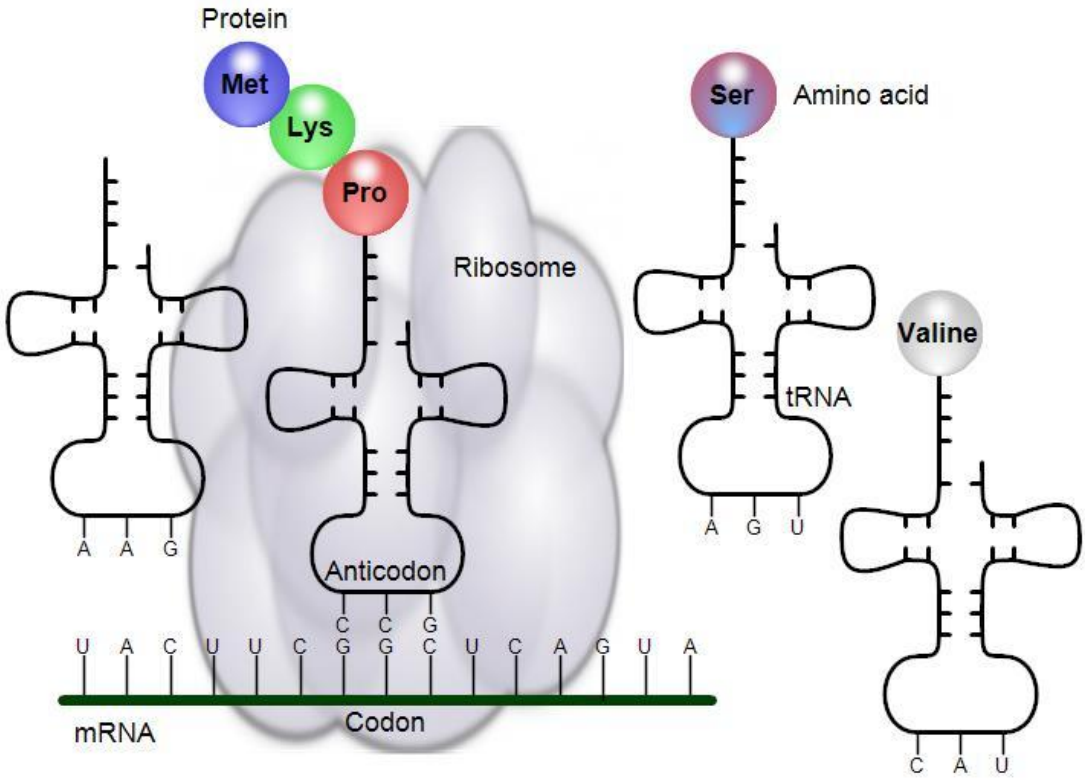


Figure 3. Illustration of the translation process.

2.2 DNA Microarrays

Gene expression levels can be quantified by measuring the amount of mRNA in a sample. The first experimental techniques for this purpose were targeted to a single gene, that needed to be chosen or suggested *a priori* for given studies. That limitation was surpassed with the emergence of DNA microarrays in the 1990s, which are currently widely used in the study of phenotypes from several biological organisms,

with broad applications on areas like the biological sciences, health care, biotechnology, etc. One of the major applications has been in the study of diseases, such as cancer.

As each slide can carry a large amount of probes, microarrays provide a comprehensive representation of gene expression levels for a given cell in just one experiment, for a given condition. This is a great advantage over the traditional analysis of a single gene [17]. The DNA microarray technology allows monitoring the expression levels of thousands of genes simultaneously [18][19][20]. Typically, a microarray experiment has as result an extensive list of genes and their measurements of expression intensity at each spot, or intensity rates generated by comparing a test sample and a normal control sample [21].

Data from microarrays can be used to compare the gene expression levels of healthy and diseased samples, which can be used in diagnosis, through classification, for example, of different types or subtypes of tumors according to their expression profile. Another possible use is the identification of genes that are more related to certain diseases, a specific knowledge that can be used in future gene therapies [22].

Microarray technology also has its limitations. For instance, erroneous or noisy readings are common and can bring serious consequences. Even the laboratory where the experiment is made has an influence on the results of microarray experiments, given the differences on quality of materials and the procedures and techniques of each laboratory. It is, therefore, argued that there is the need for standard protocols for microarrays sample preparation to improve its reliability [23].

Another aspect that has to be taken into account is that data from microarrays is not necessarily linearly correlated to the real levels of gene expression, given some biological aspects that can change the central dogma (e.g. post-translational regulation). Indeed, this relationship is complex and depends on many factors. So, some care must be taken when using this information for understanding biological phenomena. It is an area that requires extensive biological validation and application of bioinformatics [24].

In spite of these limitations, DNA microarrays are still a tool with enormous potential that has allowed to obtain numerous important results in biomedical discovery.

2.2.1 Fundaments of Microarrays

Microarrays are physical structures consisting on slides, which have coupled thousands of specific DNA molecules with a diameter less than 250 microns [25]. A slide is composed by several probes grouped in spots, each being a set of clones of the same DNA sequence. It is intended that these sequences hybridize with a biological sample under study, revealing their gene expression characteristics. The sample consists on mRNA or cDNA (complementary DNA), which is obtained by reverse transcription of the mRNA. For one channel arrays, when the sample contains sequences complementary to the probes, hybridization occurs and the spot emanates light, whose intensity is directly proportionally to the corresponding gene expression intensity (Figure 4 b)). A gene whose mRNA is produced in higher amounts will hybridize more and release more light [25].

Two channel microarrays are used for experiments where it is necessary to compare two samples (Figure 4 a)).

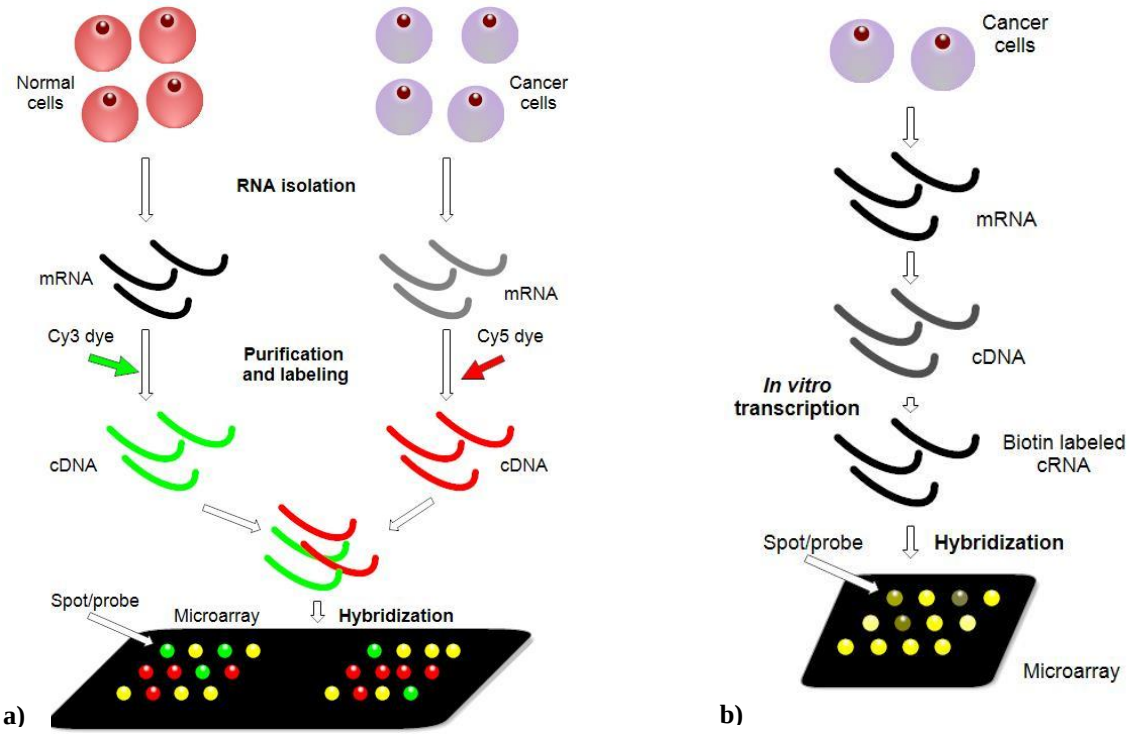


Figure 4. Microarray experiment with a) two-channels; b) one channel.

In these cases, two fluorescent dyes are used to distinguish them. One sample, the one intended to test, is labeled with a red dye (Cy5) and the other is labeled with a green dye

(Cy3) and is used as reference[2]. When two-colored arrays are used, an up-regulated gene will correspond to a predominant red (more mRNA is produced in the sample compared to the reference), a down regulated gene will have a predominant green spot; when there is no difference in the expression levels, the spot will turn predominantly yellow [17].

The spot brightness or color predominance is then scanned and transformed into a representative raw probe value, which can be treated computationally.

The decision on the characteristics of each DNA sequence to use as a probe is made by the microarray manufacturer. Also, they use proprietary methods to measure the spots light intensity, using different methods for data pre-processing. The most basic information that is possible to access is at spot or probe level (CEL file for Affymetrix). It is then possible to apply some normalization procedures, ones coming from the manufacturers, others from research community contributions like Bioconductor [16]. The dataset acquisition and pre-processing greatly affect the outcome of posterior analysis. There is a lot of variability in the results, even using a single platform [26].

2.2.2 Normalization

Normalization is the process of adjusting raw microarray data to remove systematic variation of non-biological origin. The noise and bias present on microarrays have multiple possible sources. Each manufacturer suggests a microarray preparation handling protocol to follow [27] [28], but this does not grant that is exactly executing the same way by the all research staffs that perform gene expression experiments. The material used can also have different characteristics and calibrations between experiments. Many other factors may intervene on data non-biological variation, like the quantities of RNA used on the samples, the physical and environmental conditions of the experiment, background lighting, etc. [16].

There are many normalization methods and algorithms. Some of them have specific goals, based on what is intended to be done with the data, others are broader, and became standard first steps to do on microarray data normalization. Agilent, Affymetrix and Illumina are three of the main microarray manufacturers, so a brief analysis will be

[2] This is the order in the microarrays used in this work, but other works may use Cy3 for test sample and Cy5 for reference.

made of standard normalization methods that are usually used, and the type of value that is obtained as a result for each probeset.

Agilent

An Agilent microarray spot has attached several DNA clone probes, with a target sequence of 60-mers oligonucleotides. Agilent microarrays are two-channel, being the color of each spot the result of a competitive hybridization. For the data used in this work, the reference sample used is the Universal Human Reference RNA, composed from a total RNA from 10 human cell lines [29]. The reference sample is marked with a green dye (Cy3) and the expression of the sample to be tested is marked with a red dye (Cy5). The measurement of the spot color is handled by an image processing software that makes the average intensity of pixels of that spot. Agilent's feature extraction algorithms quantify signals and their background, perform background subtraction and dye normalization [30]. The signal associated to an Agilent probeset is a log-ratio (Eq. 1).

$$\boldsymbol{Agilent\ Probeset = log2\left(\frac{Cy5}{Cy3}\right)} \qquad \textbf{Eq. 1}$$

that represents the greater or lesser intensity of the test sample in relation to the reference sample. In this way, it is theoretically possible to eliminate background variance of the equation. A common problem on the log-ratio values is the intensity-dependent dye bias. The interaction of the dyes, depending on each concentration, results on a deviation from zero of the mean values. To correct this, an ad-hoc statistic is often used that consists on fitting a smooth locally weighted linear regression (lowess) curve through the log-ratio expression values. This curve is like a collection of local means, which are then subtracted from the original values, stabilizing the overall log-ratio values mean near zero [31].

Affymetrix

An Affymetrix probeset is composed of 11 or 16 probes, consisting each probe on a short 25-mer oligonucleotide. The target sequence of a probe is usually different from the rest of the probes in the same probeset. Each probe of Affymetrix is actually a pair

of probes. One of these probes is the exact complementary sequence of the target intended to measure, called Perfect Match (PM), and the other has only the alteration of one base in the middle of the string, in 13th nucleotide, called Mismatch (MM) [32]. Because there is no contrast between two RNA samples, Affymetrix technology is considered single channel.

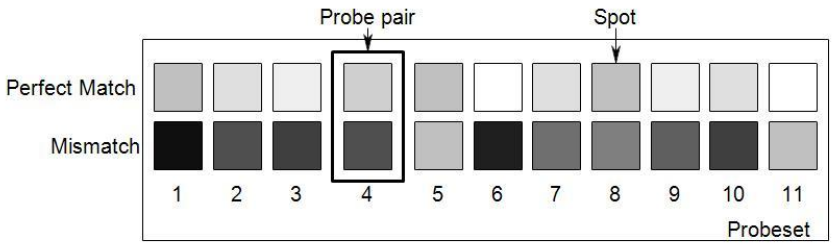


Figure 5. Affymetrix probeset structure.

The single base change in the probe MM is considered sufficient for this probe to not hybridize with the target, and its intensity is considered background noise. This probe is then used to perform background correction over PM, thus obtaining the specific intensity for this probe (Eq. 2).

$$\boldsymbol{Probe\ expression\ value = PM - MM} \qquad \textbf{Eq. 2}$$

To calculate the value of expression for a probeset, statistical methods are applied by Affymetrix, which aim, among other things, to exclude from the calculation the pairs of probes that reside outside normal ranges [33]. Roughly, a probeset is considered to be the mean of the probe pairs (Eq. 3).

$$\boldsymbol{Affymetrix\ Probeset = \frac{\sum(PM_i - MM_i)}{\#PM}} \qquad \textbf{Eq. 3}$$

where:

PM_i - Expression value of the i_{th} match probe of the probeset

MM_i - Expression value of the i_{th} mismatch probe of the probeset

#PM - number of probes for this probeset

Other normalization methods used by proprietary Affymetrix tools include smoothing adjustment, that divides the slide in areas and uses a smooth factor for background calculation by area, and noise correction, to prevent the probe signal being less than background noise, giving place to negative intensities [34].

Two popular normalization procedures for Affymetrix arrays are Microarray Suite 5.0 (MAS5.0) and Robust Multichip Analysis (RMA).

MAS5.0 focuses on within array removal of overall background noise, and consists on a local background correction, using MM probes to adjust PM intensities, and single chip robust summarization [35]. Used alone, the MAS5.0 algorithm can lead to high false positive rates, resulting from exaggerated variance at low intensities [36].

RMA includes background correction, non-linear probe-level normalization and robust multiarray summarization based on quantile normalization. This procedure excludes MM probes from the calculations, and focuses only on reducing batch effects between multiple arrays. The total gene signal is estimated by fitting a linear model to the probe data that takes into account the probe affinity effect [37]. The resulting values are in $\log_2$ scale, thus not the same scale of MAS5.0 output [35].

Illumina

An Illumina probeset is composed of several probes of the same 50-mer oligonucleotide, which are linked to beads individually optically coded. Each bead can be optically interrogated by laser excitation via a fiber-optic bundle attached to it, using a proprietary technique [38].

Illumina offers three normalization algorithms: average normalization, rank invariant normalization, and cubic spline normalization. Average Normalization is used when there are different batches of arrays, with inter batch levels variation. For normalization, an array or the average of a group of arrays is selected as reference (μ^v). The intensity (I) for a batch is normalized dividing the reference by the average intensity for all arrays on that batch (μ_i).

$$\boldsymbol{I_{pi(norm)} = I_{pi} * \frac{\mu^v}{\mu_i}} \qquad \textbf{Eq. 4}$$

p - probe index

i - array index

Rank Invariant Normalization is used when the majority of the genes on the experiment are not differentially expressed. A group of probes is identified whose rank does not change on the experiment. A gene is considered rank invariant if its rank change (r_{gi}) with respect to the equivalent reference (r_{gv}) is less than 0.05.

$$\frac{|r_{gi} - r_{gv}|}{r_{gv}} < 0.05 \qquad \textbf{Eq. 5}$$

A least squares fitting is used to compute the coefficients a_i and b_i of the linear relation between probes intensity on the reference array (y_v) and rank invariant array (y_i).

$$y_V = a_i y_i + b_i \qquad \textbf{Eq. 6}$$

The normalized probe intensities ($y_{i(norm)}$) are then calculated by:

$$y_{i(norm)} = \frac{y_i - b_i}{a_i} \qquad \textbf{Eq. 7}$$

Cubic Spline Normalization is used when there is a considerable difference of intensity between samples, or there are a large number of genes differentially expressed. The intensities of probes on sample *i* are divided in *Nq* quantiles, each one determined by:

$$q_i = \frac{i - 0,5}{Nq} \qquad \textbf{Eq. 8}$$

$$Nq = Max\left(15, \frac{Np}{100}\right) \qquad \textbf{Eq. 9}$$

Np → Number of probes in one array

The average normalization is then applied to normalize each quantile, using the corresponding quantile on the reference array [39].

2.2.3 Probe design

With the current state of the art, it is not possible to cover all genome with probes. There are several zones of the genes that are not monitored on a microarray experiment. This means that the expression of some exons is not measured and, thus, the "gene expression" concept is actually flawed. As example is shown, on Figure 6, by the zones of gene ZFYVE20 that are covered by Agilent and Affymetrix probesets. It can be noted that the probesets are directed essentially to the 3' end of the gene, matching the exons ENSE00001242854 and ENSE00001882794, belonging respectively to ENST00000253699 and ENST00000476527 transcripts. There are, however, probesets A_32_P185718 (Agilent) and 1569714_at (Affymetrix) that match several exons spread over multiple transcripts. There is no covering for all exons of the gene and, consequently, there is no cover for all the extension of the transcripts. Taking as example the Agilent probeset A_32_P185718 that reads three distinct exons, present in 6 different transcripts, it appears that it is probably impossible to distinguish gene expression between some transcripts. The target of a probeset is, in fact, a transcript and not the whole gene. To meet this proposal, probe designers try to direct targets to exons that belong to a single transcript [40].

Another goal when designing microarray probes is specificity. The genome areas that present high levels of repeated patterns are discarded, because these areas tend to repeat all over the genome and this would cause cross-hybridization. The smaller size of the Affymetrix probe sequence means that there is a greater chance of cross-hybridization in these probesets [40].

It is also important for a probe to be isothermal, since the variation on temperature during the exposure of the microarray to the test sample could lead to decrease the hybridization specificity [41].

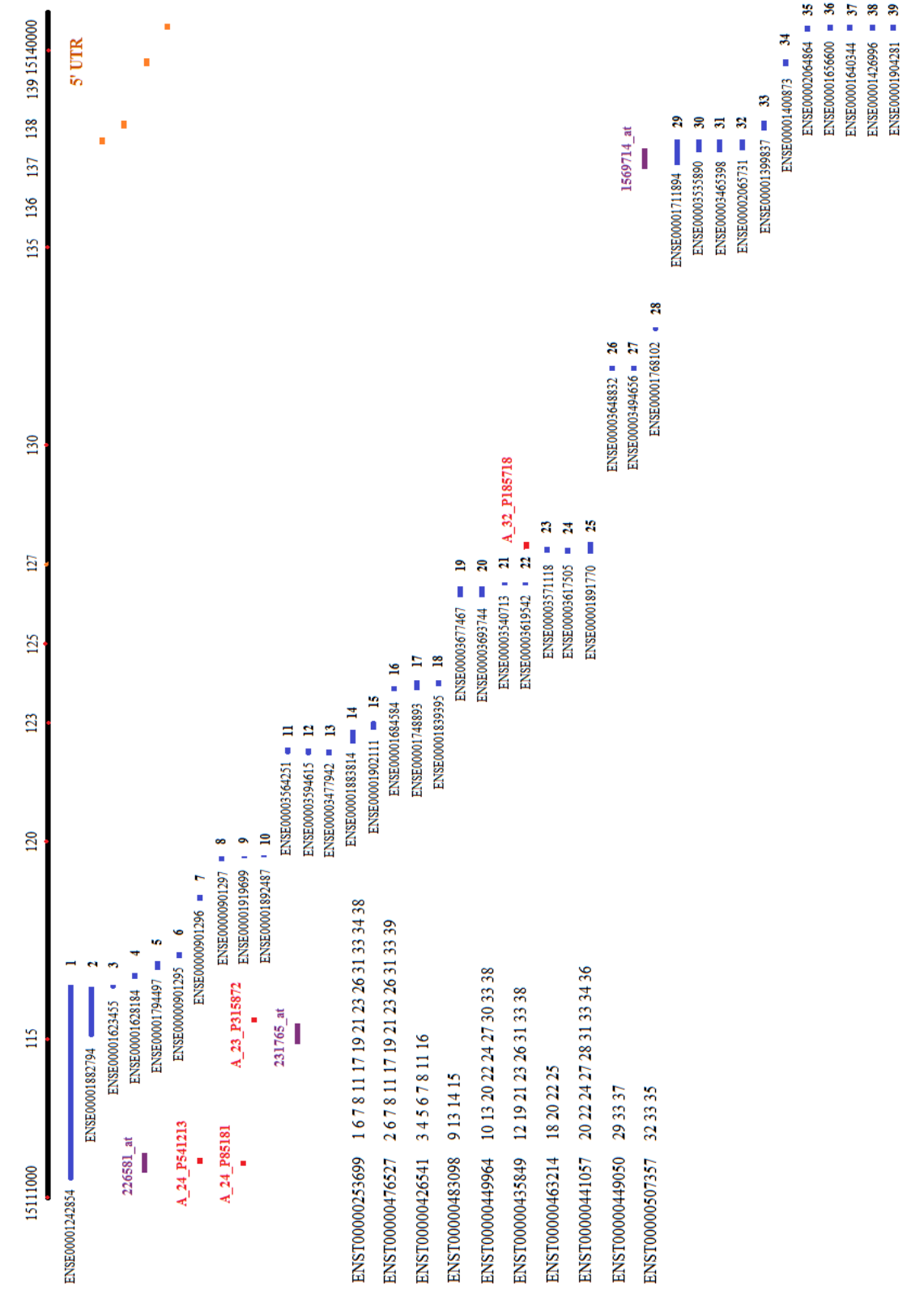


Figure 6. Agilent and Affymetrix probesets target sequence zones for gene ZFYVE20. The black line represents the extension of the gene and the blue lines represent the exons. In red, are the Agilent probesets and in purple are the Affymetrix probesets. The figure also includes the list of transcripts and respective numbered exons that comprise each transcript.

2.2.4 Microarray Data Levels and Representation

Microarray data from several research experiments can be found on open access databases, like Gene Expression Omnibus (GEO) and The Cancer Genome Atlas (TCGA). The data sets can be presented at three levels. At the first level are files with spot information, where it is possible to find intensities per probe and metadata about each spot. On the second level, the expression data is referred by probesets, and thus it is necessary to use some kind of normalization to compute values from probes into values of probesets (lowess normalization is often applied for Agilent, and RMA or MAS5.0 for Affymetrix). On the third level, expression data refers to genes, which have an additional level of applied statistical methods and considerations like to what gene corresponds each probeset. On this level, it is not possible to obtain the normalization methods and parameters used for the calculations of probeset expression, so that information is lost.

This information is very important when trying to integrate datasets from different experiments, because it is not possible to establish a mathematical bridge between expression values if we do not have the minimal information about their calculation process. TCGA stores the three levels of data, but for GEO the data is, generally, at the probeset level. This aspect must be considered when deciding at which level is better to do dataset integration. The normalization already done also has to be taken into account, such as the scale of the data.

A gene expression dataset is represented by a two dimensional array of values (probesets expression matrix), with a size of n samples (columns) by m probesets (rows). Each array element ($W_{i,j}$) represents the value, that is a real number, of the expression of probeset i in sample j. For each dataset, there generally exist corresponding phenotype data, where the phenotypic characteristics for each sample/individual considered on expression dataset will be given.

2.3 Statistical analysis of gene expression data

The development of DNA microarrays, as well as other related technologies (e.g. CGH arrays), has brought major breakthroughs in the possibilities to study how gene expression influences changes on the phenotype.

The analysis of these data focuses mainly on three aspects [40]:

1. Which genes are differentially expressed in two (or more) sets of samples representing distinct conditions (e.g. disease states, treatments, etc)?
2. What are the relationships between the genes/samples that are being measured?
3. Is it possible to classify samples based on measurements of gene expression?

For the first task, we may have, for example, tissue samples collected from the same patient at different instants in time. If there is an outbreak of disease accompanied by a change in gene expression, it is possible to establish a correlation between the two facts.

An example of the second case may occur when trying to find a correlation between genes that share expression patterns in a set of samples.

In the third case, the aim is to identify a set of genes that are directly linked with a particular phenotype and build a classifier that allows distinguishing between sample classes (e.g. can be used to provide an early diagnosis for certain diseases). Therefore, it is possible to act in a more precise way in the treatment of certain diseases or just to make an improved monitoring.

The three previous approaches are possible by analyzing data from microarrays. It is possible to collect information electronically, thus providing an universe of data that can be processed, filtered and worked up to get the needed answers. Thus, physicians are not limited to mere observation of a phenotype, which in critical and specific cases are not effective enough to establish a comprehensive assessment of a patient health situation.

Since many of the analysis methods are based on statistical methods, in this section, some basic statistical concepts needed to analyze gene expression data will be described.

2.3.1 Hypothesis tests and p-value

Statistical inference is the generalization of the conclusions made based on a sample, to an entire population. A test of hypotheses builds a probabilistic model of the observed data, based on a null hypothesis. A common test used in microarray analysis deals with tests regarding the means of two samples. To test if a gene is differentially expressed in two groups, the null hypothesis is that the mean of the expression levels in both groups is the same and then, the means difference is approximately zero.

The null hypothesis is rejected when there is a small chance (given by a significance level) to observe its occurrence. If the null hypothesis is rejected, it means that the variability in the data reflects the biological effect to be observed.

The p-value is the probability of observing a statistic, for example an average difference in gene expression, that is at least as extreme as the observation in the sample data, given that the null hypothesis is true. For example, a p-value of 0.01 means that there is 1% chance of observing at least the same level of sample differentiation in gene expression, through a random search in samples taken from two populations with the same means. Lower p-values stand for higher confidence in rejecting the null hypothesis [40].

2.3.2 Parametric Tests

Parametric tests can typically be used when the statistical variable X being observed fits a Normal distribution. When executing tests to compare the expression means (for example between sample and control) the most traditional method is a *t*-test. This implies calculating the t-statistic using the following expressions [40]:

Paired samples:

$$t = \frac{\bar{X}}{\sigma/\sqrt{n}} \qquad \textbf{Eq. 10}$$

Unpaired samples:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \quad \textbf{Eq. 11}$$

where:

n – size of the sample(s)

σ – standard deviation

The *t*-test calculates the probability that the observed difference in the means occurs when the null hypothesis is true. It is often assumed that there is equal variance between sample and control [42], but this assumption should be validated.

Multiple testing problem

This problem occurs when trying to perform a set of tests at the same time. Taking as example gene expression, the probability that one of the genes is shown, purely by chance, to be related with a specific phenotype increases with the increase in the number of genes on the test. When multiple testing, we can observe a greater percentage of hypothesis tests that incorrectly reject the null hypothesis, increasing the percentage of false positive discoveries, when compared with single comparison tests. Thus, it is expected that when it is wanted to verify if a specific assumption generalizes for a new dataset when there are a large number of attributes in equation, to verify a decrease on the confidence level [43].

Corrections for the problem were proposed, like the Bonferroni correction, which divides the critical p-value by the number of comparisons to be made [44].

Šidák correction produces better results than Bonferroni, when attributes are independent from each other. If we have n attributes, then the significance threshold α should be corrected by [45]:

$$\alpha_{corrected} = 1 - (1 - \alpha)^{1/n} \quad \textbf{Eq. 12}$$

Holm-Bonferroni is other method for multiple testing problem correction, which adopts a stepwise sequential procedure. It consists, on the first stage, into testing each attribute separately to obtain p-values, followed by an ordering from the smallest to the greatest

p-value attribute. The test with the lowest probability is tested first with a Bonferroni correction involving all tests. The second test is made with a Bonferroni correction involving one less test and so on for the remaining tests [46].

2.3.3 Non-parametric statistics

Non parametric tests are used when the distribution of the data is not known and cannot be assumed as Normal.

Classic non-parametric statistics

The Wilcoxon/Mann-Whitney rank sum test replaces the numerical value of the data by a rank. The sum of ranks for positive values is calculated and compared with a pre-established table for p-value determination. This test is less powerful than the parametric or bootstrap equivalent since it presents less probability of seeing a positive result (rejecting the null hypothesis) when the positive result is true [40].

Bootstrap analysis

Bootstrap analysis does not require Normal distributed data and it is robust against noise and experimental artifacts. It is also more powerful than the classical non-parametric tests. The disadvantage of the bootstrap analysis is the computational effort required [40].

Bootstrap is based on constructing a large number of random data sets by resampling of the original data. The class/group labeling of original samples is shuffled by the random sampling process. The t-statistics are calculated for all the genes in each random data set and the standard t-distribution is used to find the p-value distribution for each gene. For this distribution, an adjusted p-value can be calculated for the original data set [47].

2.3.4 ANOVA and general linear models

The analysis of variance (ANOVA) method can be used to calculate p-values related to finding observed differences in means among more than two conditions [42] and linear models can be used when the phenotypic variables involved are numeric.

The one-way ANOVA method

To compare if the gene expression is different in more than two groups, the analysis could be done in two ways:

- Naively, t-tests can be applied to each pair of groups and select significant genes on one or more of the t-tests;
- Use a statistical test that compares the several groups simultaneously and returns a single p-value.

In the first case, the likelihood of detecting significant differences between two groups due to measurement error is increased and multiple testing should be corrected. Another problem is that each comparison is independent of the other, making it difficult to interpret the results.

The one-way ANOVA compares two or more groups and returns a single p-value that is significant if one or more of the groups is different from the others [40]. Post-hoc tests (such as Tukey test) can be used to find pairs of conditions for which the differences are significant.

Multifactor ANOVAs

In the previous cases, it is considered that there is only one variable affecting the outcome (the group to which the individual belongs). Consider the case of gene expression data where samples belong to patients suffering from two types of leukemia, while also it is believed that the sex of the patient can have an influence that needs to be studied. In this case, the gene expression values would be modeled as a variable that would depend on the type of illness and also the gender, or both. ANOVA models can be built more generically to include two or more factors and to return a p-value for each factor separately [40].

General linear models

In the presence of a continuous variable, we may consider that the gene expression responds linearly to this variable and use general linear models to analyze the data. General linear models return a *p*-value for each factor and can also include interactions, for example when a drug administration presents different results based on the patient gender.

Both ANOVA and general linear models require that the data follow a Normal distribution. However, bootstrap analysis can be applied to these more sophisticated tests, which makes them applicable to microarray analysis [40].

2.4 Unsupervised learning

Unsupervised learning is a kind of machine learning method that not depend on previous known outputs neither on a supervisor. The main concept is to find patterns on data, that represent significant interactions (or the lack of interactions) which would not be considered as pure unstructured noise [48].

2.4.1 Clustering

Clustering is a unsupervised learning method, where is tried to find sets of features with highly correlated profiles, gathering them into clusters. The aim is to find clusters containing attributes that have similar patterns (maximizing homogeneity) and that have patterns distinct from other clusters (maximizing separation).
Several experiments are done in different conditions and the clustering is done according to the similarity between the feature response variations across these experiments [17] [49]. It is particularly useful for gene expression studies where a phenotype is not given, since it can suggest structured distribution into classes, not necessarily identifying that classes. When a gene expression set falls into a determined pattern it is evident that it suggests a specific phenotype branch, and it can thus contribute to class discovery.

Hierarchical clustering

In hierarchical clustering, the clusters are organized hierarchically, existing general clusters and more specialized sub clusters associated with them. The approach may be bottom up, if we start from the bottom of the hierarchy with each sample on a separated cluster and then climb the hierarchy, clustering samples on a pairwise mode. On a top

down approach, we start from the top of hierarchy, with all the samples together on the same cluster, and the samples are being splitted as we descend the hierarchy [50].

k-means

In this type of clustering, k represents the number of clusters into which divide the data. For each cluster is defined a specific mean, or center, and a sample is associated with the cluster with the nearest center, by minimization of the mean squared distance between them [51].

While it was mentioned clustering of samples, the same approach may be applied to gene clustering.

Clustering is particularly useful in suggesting subtypes of a specific disease that are not known a priori. Golub et al. [23] proposed a method of distinguishing between two classes without prior knowledge about these classes. The work was divided into the tasks of finding classes and predicting classes. They began by selecting genes whose expression was most correlated with the class distinction to be predicted. They identified a set of genes that had a better correlation compared with results obtained from random selection. To find the classes, the genes were grouped into clusters using self-organizing maps technique. The number of clusters is defined by the researcher that has in mind that a greater number of clusters could identify subtypes that before were grouped in the same cluster. Some classifications may reflect contamination or other characteristics not related to subtypes of the disease, which suggests making some analysis (e.g. microscopic) to confirm the results.

Ramaswamy et al. [52] created a database of gene expression containing 218 tumor samples, with 14 distinct types of human cancer. They used recursive elimination of attributes. An initial unsupervised analysis was performed (clustering), letting the dominant structure in a dataset dictate the separation of samples into clusters, based on similarity between gene expressions, without prior knowledge about the identity of the samples.

2.4.2 Principal component analysis

Principal component analysis (PCA) consists in the projection of data into a new space oriented by the principal components as axis. For a *n* dimensional dataset, principal components are the n eigenvectors-eigenvalues pairs. An eigenvector is a direction that represents the greater variance of the data. The firs eigenvector represents the greatest variance, the second a little less and so on. An eigenvalue is the value of variance in a given direction. Greater variance means more information, thus the last eigenvector is the less informative. As these principal components represent the data, we can discard the less informative component of the data and still stay with the most representative information of the dataset [53].

PCA is largely used to reduce the dimensionality of the problems, seeking to retain the main features of the data and eliminating redundancy. This method has the disadvantage of typically still leaving a substantial number of features to study and, since it is unsupervised, it might also be unable to divide the classes in the space. Also, principal components are based on linear combinations of the original features, leading to their destruction when dimension is reduced [40].

2.5 Supervised learning

On supervised machine learning, there is a vector of inputs and a vector of outputs, that function as supervisor. The supervisor learning algorithm takes these two vectors and tries to establish a predictor able to predict the output vector, given the input vector, with the maximum accuracy possible. The predictor is then tested on other input/output vectors, not used on training, to validate their generalization power. If the results are acceptable, the predictor can be used to predict outputs in new cases, where only the input vector is known [54]. The predictor is a classifier, if the outputs are discrete or a regression function, if the outputs are discrete. This work will address in particular classifiers.

2.5.1 Classification in gene expression context

A simplified design of classification considers that each class occupies a space with well-defined borders, within the universe of gene expression values. This universe has many dimensions as the number of genes considered, and the amplitude in each axis is given by the expression of the respective gene in the sample. In practice, it is difficult to ensure that each class occupies only a well-defined region of space, which means that there are overlapping samples found in areas different from the one where they are most likely to belong.

As for the linearity of the separation, the classes may be considered as linearly separable (when it is possible to separate the classes using lines or planes), or non-linearly separable (when this is not possible).

The choice of whose genes will be used in classification is very important as it directly influences the ability to separate or not classification areas in the space. For example, the combination of two genes may have a weak division of the area of classification, but the introduction of a third gene may considerably improve this capability. This problem was addressed in the previous section.

Classification in the context of gene expression is used when an hypothesis is formulated that some phenotypic attribute has some kind of relation with the expression of a set of genes. For example, samples from a set of patients with a specific disease can be compared with samples from a set of patients without that disease, and verify if there are genes that present a marked difference on expression between the two sets. If that occurs, then the set of genes that presented a consistent differential expression may be used as biomarkers for that disease on new patients diagnosis.

The identification of biomarkers has a special importance on tumor's diagnosis. In many cases, traditional methods are not able to distinguish tumors with very approximate histopathological characteristics, since it is necessary to take into account various empirical observations, such as morphology, clinical and molecular aspects [25].

Classification accuracy increases with the number of correctly classified samples used on the predictor construction. However, this is not the only aspect to take into account and other factors like data quality, feature selection, classification method and the respective parameters adopted have to be regarded [55].

2.5.2 Classification algorithms

Classification using supervised Machine Learning techniques consists of inducing a functional model from a dataset. In the dataset, a number of examples are available (in this case, corresponding to samples) where the correct classification is known. The expression of each gene will be an attribute. The available examples are used for model training, i.e. building the model and setting their free parameters to adequate values. The model should be able to predict the class of a new example after being trained [22]. In the next subsections, a number of alternative classification algorithms are briefly surveyed.

Decision trees

It is one of the most used classification methods within the machine learning community due to its efficiency and ease of interpretation. A decision tree is a structure where a sample can be classified by sorting their feature values on a tree-like structure. Each node represents a feature and each branch represents a interval of values where the value of that node may lay. Samples are classified starting at the root node and sorted based on their feature values (Figure 7).

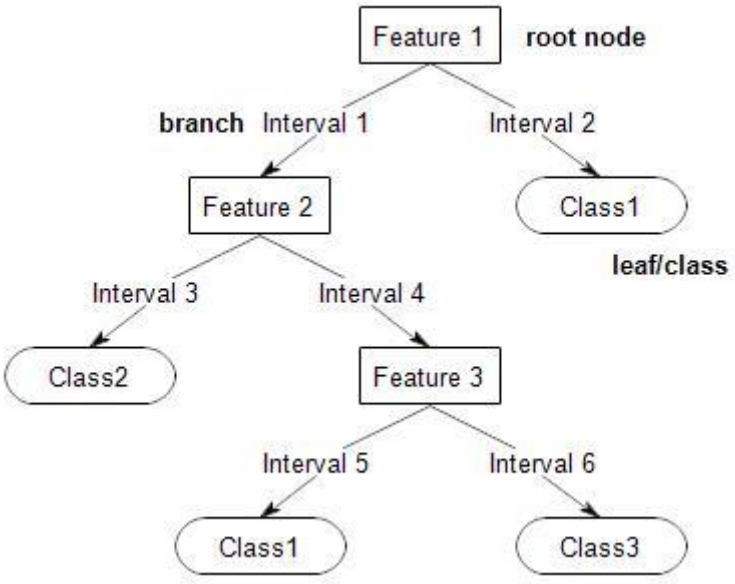


Figure 7. Generic structure of a decision tree.

The main weaknesses of decision trees are their instability and low accuracy in some problems, although this can be remedied with the use of multiple trees together. By pooling several trees we get a decision forest [55].

K-nearest neighbors (kNN)

This algorithm is the simplest representation of Machine Learning techniques based on instances. kNN stores all training data and classifies, usually by majority vote, a new sample according to the most common class in the k nearest neighbors of the new sample [22]. This method requires calculating which samples are similar to a given sample (usually Euclidean distance is considered) [22], [56].

The kNN algorithm is simple [22], intuitive, easily extended to multiple samples, is able to separate classes on the nonlinear space [57]. However, it is not very robust to extreme values, and so these can be misclassified [40].

Centroid classification

The centre of mass of each of the possible classes is calculated, given the known samples. For the new sample, the distance to each centre of mass is calculated, and then the new sample is classified according to the centre of mass that is closer. This classification is easily extensible to more than two samples, uses all available data and is relatively fast. However, it can provide misleading results if data are not linearly separable [40].

Linear discriminant analysis (LDA)

This method is based on the creation of a straight line or hyper plane that best separates the known samples into two classes. Given a new sample, its classification depends on the side of the line or hyperplane where it stands. As the LDA uses the variance of the data to compute the division boundary, it is a better method than centroid classification. However, its application to more than two classes is not as intuitive and, in any case, it only works if the data is linearly separable [40].

Artificial neural networks

An artificial neural network (ANN) can be used to analyze, data when it is not possible to establish a mathematical relationship between inputs and outputs of a system [58].

The method involves training a neural network with samples whose classes are already known, so that it learns to classify correctly even if it be provided with new samples [40].

ANNs have been widely used in the processing of gene expression data due to its ability to successfully identify the complex relationships in large data sets. However, the determination of network topology is a complex problem, depending almost entirely on heuristic methods [59].

ANNs are naturally applicable to more than two classes and have the great advantage of being able to deal with data that are not linearly separable. For those reasons, ANNs are the preferred method for many situations. Its disadvantages are related to the processing burden required to train and validate the network model and to the need of using empirical methods to define the network parameters. In this way, it is not always possible to interpret objectively the results [40]. Another reason for the ANNs not being a first choice for gene selection is the existence of many free parameters to estimate in these models, while only a few training examples are available, which may lead to overfitting [56]. However, there are algorithms which allow training ANNs models on a large scale [58].

Support vector machines (SVMs)

SVMs have some similarity with the LDA method, since they divide the data space using a straight line or hyper plane [60]. The difference lies in how this separation is done, opting for the boundary that maximizes the margin of separation between classes, i.e., the one with lowest classification error [56] [22] [61].

SVM training methods begin by projecting the data in a space with more dimensions to be able to linearly separate the data using a so called kernel function, and then proceed with size reduction. In this way, they are able to deal with nonlinearly separable data. SVM training is faster than ANNs, but have the same need for the application of empirical methods [40].

The popularity of SVMs comes from their ability to generalize and high robustness to deal with large data dimensionality and noise [22]. SVMs have mathematical characteristics that make them suitable for gene expression data analysis, such as flexibility in choosing a similarity function, it is not 'sparse' when dealing with large data sets, and have ability to handle large quantities of attributes and identify outliers [62].

Brown et al. [62] described the use of SVMs to classify genes by functional categories, using gene expression data. They used training vectors concatenated from multiple sources.

Naive Bayes (NB)

It is a simple probabilistic classifier based on Bayes theorem. It is naively assumed the independence between attributes. NB can handle an arbitrary number of independent attributes, whether continuous or categorical [22].

Logistic regression

Logistic regression is used when it is intended to make a prevision where the inputs are continuous variables and the output to be predicted is categorical, consisting of only two classes. Logistic regression is based on the logistic function, which gives the ratio of the odds of an event occurring to it not occurring [63]:

$$\boldsymbol{P = \frac{e^{\propto+\beta x}}{1+e^{\propto+\beta x}}} \qquad \textbf{Eq. 13}$$

Where:

P – Probability of a class versus the probability of the other class

α and β – parameters of the model

2.5.3 Feature selection

Feature selection and classification are the two main inter-related tasks in sample discrimination using gene expression patterns: to find genes that could be good biomarkers for some disease or condition and to create models that are able to predict the outcome of an interesting variable (e.g. diagnostic, prognostic) based on gene expression values for a given sample. Those two tasks are related to basic Data Mining/ Machine Learning problems: attribute/feature selection and building accurate classifiers. In this section, methods for feature selection will be reviewed, while methods for classification werw addressed in the previous section.

The information gathered from microarrays experiments has a typical dimension of thousands of gene expression values for each sample (between 2000 and 50000 [40] [59]). When trying to use this information for classification, we find a huge disparity between the small number of samples (typically a few tens or hundreds at most) and the large number of gene expression measures [25] [22] [19] [59] [56]. This disparity affects the classification accuracy, the estimation error characteristics and the selection of classifiers [64]. These issues, together with variations in gene expression measurements, caused unintentionally during the reading of microarrays, are currently the main difficulties in the classification process using microarray data [57].

If each gene represents a dimension in space classification, working with all of them becomes an almost impossible task. Thus, we need to carefully select the genes that can contribute significantly to the classification of the samples under study. Generally, only the expression of a small set of genes is associated with a particular phenotype, and so it is crucial to make a good analysis and selection of genes [57].

The reduction in size also serves to discard genes with redundant information, reducing the chance of obtaining results from random situations, removing noise [65] and greatly increasing the speed of training of classification models. Thus, the process of feature selection boils down to selecting the set of genes with fewer elements that can classify a disease with a high degree of accuracy [18]. The main approaches for dimension reduction can be divided into filter and wrapper methods [57], which will be addressed in the next sub-sections.

Filter methods

Filtering methods are generally based on assigning weights (or ranks) to attributes based, for example, on the statistical significance of the information (correlation, entropy, etc.), and then carrying out a pre-selection of these before the model inference stage [56]. Filtering is easy to understand and its analysis is relatively fast, but some correlations involving the discarded attributes are irretrievably lost. Examples of filtering methods are selection by correlation, statistical tests (t-tests, one-way ANOVA), information gain and mutual information.

Wrapper methods

Wrapper methods are focused on improving the classification process through adaptive learning, showing in the most of cases better results than filtering. This is due to the preservation of dependencies between attributes and by considering the collective contribution of genes to model construction [61]. However, this approach leads to a greater processing load [57] [56] and to a greater probability of overfitting [61]. In this approach, instead of using an independent test method, a specific machine learning classifier (decision tree, support vector machines, k-nearest neighbor, etc.) is used and the corresponding performance rating is used as a criterion for feature selection [61].

These methods typically rely on an optimization method that tries to optimize an objective function related to the classifier's accuracy. Optimization methods are normally meta-heuristics, such as genetic algorithms, that search for the best subset of the attributes that maximize the objective function [56], which can include both performance and model complexity metrics.

In the simplest cases, greedy approaches are used, that interactively add/remove features from the current set.

The selection can be done recursively until new additions or deletions of genes from one set do not result in increased performance of this group. The recursive process may also stop when a performance level is reached, or when a stopping criterion is reached, as the maximum number of iterations or the minimum number of genes in a set. The performance evaluation is done by an independent testing method [61].

Individual gene selection

Is done by identifying those genes that, on their own, best separate the classes. Then, a set is formed with the best classifiers. This method is quite simple, but not very effective, since genes that are good classifiers individually can fail when associated with other genes. Likewise, genes that are individually weak classifiers can be valuable when associated with other genes [40].

Recursive attribute addition or removal

Recursive methods are made by successive iterations. At each of them, the influence, in terms of information gain, of adding or removing genes to a set is evaluated. Liu et al. [19] presents a scheme for recursive addition of attributes, based on supervised learning and similarity values between selected genes and candidate genes. The scheme is

divided into four phases. In the first, noisy data is filtered, by a selection that results from the application of a statistical test. In the second, a rank is assigned to each gene, according to the accuracy of the classifier. Thirdly, the gene with the highest rank is chosen (if more than one has the same level, the one with the smallest p-value is chosen) that will be the first dimension. Finally, the next dimension is made with two genes with the higher rank and / or lesser correlation, and so on.

Taguchi method

It is a statistical method used mainly in quality management processes, which concerns the analysis of processes or products and their improvement through relatively small changes in design factors. If the parameters that influence a process can take several values, then there must exist values that provide a better realization of the process or product [57]. Applying the concept to the selection of genes, it translates into checking whether the addition of a gene to a subset of genes will cause it to gain information that is sufficient to justify cost of the addition that this will entail.

Pairwise gene selection

Pairwise gene selection differs from the previous selection method in the fact that it identifies pairs of genes that are good classifiers and not individual genes. Since the set of hypotheses of gene pairs is larger than the set of genes themselves, the process becomes slower, but overall it performs better and usually a pair of genes is a better classifier than a single gene [40]. From the previous genes selection methods (individually or in pairs) a set of classifiers emerges, that can be used for the construction of an ensemble classification model. This model gets the results of all classifiers and performs a selection by vote. Each gene or gene pair classifies a sample as being of a particular class. The final class is the one that has obtained the greatest number of votes.

Selection of the best classifier

The same set of genes may show different results when used in different learning algorithms, so it is important to determine not only the best set of genes but also the classification model that best fits this set.

Küçükural et al. [18] proposed a GA with a selection strategy of the type Russian roulette, for selecting a subset of attributes nearly optimal for cancer classification. They

tested several parameter combinations until reaching a combination with optimal results for all data. Compared with LDA and ANNs, the SVM showed the best results in the classification process. The selection of a smaller number of attributes is made from one generation to another, replacing the individuals with the weakest fitness in population by descendants who have better fitness.

Lorena et al. [22] presented an empirical study of complexity in the classification of gene expression data sets related to cancer. The complexity of Machine Learning models was measured and the results indicated that most of these data sets can be effectively discriminated by a linear function. The classes covered in the study are different subtypes of tumors. After a simple and unsupervised selection of genes, three classification techniques were applied: kNN, Naive Bayes and SVMs with LOOCV. To eliminate uninformative genes, the 10% highest and lower values were discarded. Next, the genes whose expression levels differed by less than l-fold at least in c samples from the average expression over the samples were removed.

Liu et al. [24] used genetic algorithms combined with SVMs for classification of cancer in multiple classes in a wrapper approach. The used attributes are determined automatically by interacting Genetic Algorithms (GAs) with SVMs, leading to compact sets of relevant genes with the highest performance rate found to date.

Ando and Iba [56] have focused on discrimination by classes of different gene profiles, testing some combinations of supervised learning and combinatorial search, in a wrapper approach. The authors also used an extended genetic algorithm method for feature selection. The problem of searching for the optimal subset of genes for classification was proven to be a NP-complete problem. If there are n genes, there will be 2^n possible subsets of these genes, which make the analysis of all possible cases an impractical approach. They obtained the best results by combining a genetic algorithm optimized with a Naive Bayes classifier.

Li et al. [49] made a systematic study of sensitivity, reproducibility and stability of gene selection to help in the choice of algorithm parameters. A genetic algorithm is combined with the kNN method to identify genes that, together, can discriminate between different classes. The strategy is based on finding several subsets of attributes and then evaluating the relative importance of genes in the classification of the sample, by examining the frequency with which each gene appears in these subsets. It is considered that the most frequent are the most relevant in distinguishing between classes.

Ooi and Tan [66] used genetic algorithms to do multi-class prediction. They made a scheme based on genetic algorithms which automatically determines the members of the group of genes to use, as well as the optimal size of the group that maximizes the success of classification, using a classification method by maximum similarity

Deb and Reddy [67] tried to minimize the attributes set and minimize the amount of wrongly classified samples. The classification is made considering only two classes. The study highlights the need for a collaborative effort between biologists and computational algorithm to achieve a classification process more consistent and meaningful.

Hybrid Models

Some authors prefer to apply a hybrid method that combines filtering with wrapping, as seen in [61]. A dimensionality reduction method is considered to be hybrid when there is interaction between feature selection and classification. An independent test and a performance evaluation function are used to evaluate the subset of attributes [61]. An example of a hybrid method can be seen in [68], where fuzzy classification rules were used to identify correlations between genes. A gene is rejected if it is not correlated with other genes [68]. There is not always a well-defined line of separation between filter, wrapper and hybrid methods. Some feature selection methods are briefly described below.

Wang et al. [60] intended to find the smallest set of genes that would ensure a highly accurate classification of cancer from microarray data using supervised machine learning algorithms. Thus, the processing burden and the noise from irrelevant genes would be significantly reduced. The authors argue that it is possible to extract simple diagnostic rules that lead to accurate diagnoses without the need for complex classifiers. Fewer genes in the tests lead to lower costs. The work allows developing research to find possible biological relationships between a small number of genes, the cancer evolution and its treatment. The authors used the following approach: (i) choose the most important genes through a ranking; (ii) to test the classification ability of all combinations, via a classifier. The adopted method was to classify first with each gene, verifying if a good classification was obtained and then repeat the process with each possible pair of genes and so on. The classification algorithms used were Fuzzy Neural Networks and SVMs. The validation was done using 5-fold cross-validation.

Chuang et al. [57] proposed a hybrid system that combines *a priori* selection of attributes, based on their correlation, with genetic algorithms for a global search in the space of attributes, trying to escape from sub-optimal solutions. The genetic algorithm is assisted by the Taguchi method and kNN is used to classify eleven profiles. Methods are validated with LOOCV, based on the calculation of Euclidean distances.

Peng et al. [61] proposed a hybrid process that combines filtering with wrapping. They made a pre-selection of attributes to make a more effective search for the subsets with increased performance of classification. They used ROC (Receiver Operating Characteristics) to characterize the classification performance of the attributes individually and combined.

Zervakis et al. [69] studied gene selection algorithms to reveal inefficiencies in the performance evaluation processes and advance with features that can reduce the uncertainty in the algorithmic validation. They did a PCA to the three initial data sets to verify the separability between classes. First, all data was used for training, and subsequently the attributes (wrapper) were iteratively removed. In each iteration, the classifier is trained with the available attributes and the result is validated with an independent gene set. The process is repeated until the minimum set of genes with the highest classification accuracy is obtained. At this point, the Q-statistic is used to check the level of differentiation of the genes within each class and the correlation with the disease gene.

With the notion that the dimensional reduction leads to an inevitable loss of information, such as the correlation between the genes that are discarded, it may be interesting to investigate learning methods that can handle a large number of attributes.

Boger [58] opted for a hybrid approach, using an ANN trained with data from microarrays, using recursive pruning of the inputs and re-training techniques to identify the most relevant genes. The algorithm used to train the model was PCA-CG (Principal Component Analysis - Conjugate Gradient), trying to demonstrate the applicability of algorithms for large-scale training when analyzing data from gene expression microarrays.

2.5.4 Classifiers combinations

The combination of several classifiers (ensembles) tends to bring some robustness to the classification process, whereas the parameters can be changed between training models or different sets of genes can be tested as attributes.

Huang et al. [55] proposed an improved decision-forest (IDF) with a graphical user interface. In their experiments, data sets corresponding to the expression of four genes are used. They used a built-in mechanism for selecting attributes that, together with the fact that they deal with a few genes and only considering three parameters of the classification method, leads to a more efficient training, when compared to other methods like SVMs, or leads to fewer branches than the traditional decision forests. The different trees that make up the forest are obtained by applying, in each construction, different variables in the root nodes. The parameters considered were the minimum number of leaves on a node, the maximum number of trees in the forest and an indicator of how often an attribute can be used in the forest. The authors argued that their method is more immune to variations in sample size than other methods.

Chen et al. [59] argue that a single ANN for classification becomes an unstable process and does not have the required accuracy. The solution is to make a filtering of the genes through t-test and use a combination of several ANNs for classification. The classification method proposed was compared with the use of only one ANN, with the bagging of ANNs and with SVMs, having obtained the best results. The grouping of ANNs was proposed by Hansen in 1990 and improved the ability to generalize with little computational cost. As a selection method, the t-test proved to be an effective method in the problem of binary classification, better than Fisher Information Gain. The ensemble included 10 ANNs, with a number of hidden nodes limited between 3 and 50.

Deutsch [65] proposed grouping predictors, each one using a different combination of genes, to generate an optimal set of predictors. LOOCV validation was used to measure the ability of the models. The best results were obtained when a parameter was added that enhanced the importance of those classifiers that could separate the data by type of cancer. A wrapper method was used, but also with a filter applied at the beginning. For the wrapper a GA was used as the optimization engine, together with a kNN classifier.

The use of multiple classifiers of different types makes it possible to make a further contribution of the combination of multiple sources, boosting the results in different kinds of problems.

Kim and Cho [70] found an optimal combination among several attribute classifiers using a GA with deterministic crowding. A set of classifiers allows to obtain more reliable solutions then a single algorithm. A selection has to be made of the best classifiers, since the use of all possible classifiers together would be impossible to achieve. ANNs, kNN, SVM and Structure Adaptive Self-Organizing Map have been selected. For attribute selection Pearson Correlation, Spearman Correlation Coefficients, Euclidean Distance, Cosine Coefficient, Information Gain, Mutual Information, Signal-to Noise Ratio and PCA were applied.

Cho and Ryu [25] made a classification framework that combines a pair of classifiers trained with mutually exclusive attributes, to classify cancer patients through their gene expression profiles. Using attributes with non-overlapping correlation is desired to form different classifiers for different aspects of the training data, thus increasing the solution space that is possible to achieve. The process is divided into attribute selection and classification. The selection is made by considering multiple attributes with mutually exclusive characteristics. In this way, several candidate solutions are formed that can be combined to get closer to the optimal solution and, thus, discard the use of many attributes that only cause noise, redundancy and overfitting. Cross analysis, such as Pearson and Spearman correlation coefficients, and *a priori* knowledge can be used to measure the degree of attribute information. Euclidean distance can also be used to measure the degree of similarity (distance) between two vectors of attributes. To measure the performance of each vector information gain and mutual information are used. The various classifications are made using the subset of selected attributes and the complementary use of various methods of classification (ANNs, SVMs, kNN). ANN is employed to combine the outputs of several classifiers.

Umpai and Aitken [71] researched over the multi-class classification. The methods that have been used in the gene selection were t-statistics, information gain, ratio of sum of squares of between-groups to within-groups and PCA. In this study, the RankGene software for feature selection was used. They used, as classifiers, the Fisher LDA, the discriminant of Maximum Likelihood rules, Decision Trees, SVMs, kNN and ensembles of classifiers. The performance was tested with bootstrapping. The evolutionary algorithm considered a population of classifiers as predictors, whose effectiveness could be determined using them as attributes in a kNN classifier. The parameters that were explored are population size and number of attributes.

2.5.5 Validation

To evaluate the models ability to correctly classify new samples, with which they were not trained, a number of techniques are available, often using test sets or cross-validation [40]. A test set is taken from the entire original data set, being the remaining data used to train the model. After training, the model is applied to the classification of the test data, measuring their ability to generalize. This validation is quite effective, showing the generalization error of the model, but becomes impractical when the total amount of data is not sufficient to form training and test sets that represent satisfactorily the problem [40].

In cross-validation, the original dataset is divided into k subsets. The validation process is executed k times and, in each iteration, one of the sets is used for testing and the remaining sets (k-1) are used for training. Each of the k partitions is, therefore, used as a test set only once. The final accuracy is measured by the sum of the number of correctly classified samples in each partition. Cross-validation is very useful when there is limited data available. One variant of this method is to consider partitions with only one sample - Leave-One-Out cross validations (LOOCV) - where each sample is predicted through the rest of samples in the original dataset.

The validation process assumes an even more important role when different combinations of various classifiers and various methods of feature selection are considered, with each hypothesis leading to different results. The incorporation of attribute selection within the machine learning method is already used in certain models (hybrids), thereby decreasing the multiplicity of combinations and the need to provide a data set for validation tests [55].

As the method of cross-validation can be computationally heavy, it is sometimes avoided, but this should not be the case for microarray data because there are generally few samples. So, many authors use LOOCV that is often criticized because of high error variance compared with 5 or 10-fold cross-validation [20].

2.6 Gene Expression Data Integration

The existing statistical models work with a reduced number of predictor features and demonstrate greater effectiveness with higher numbers of available classified samples. The characteristics of microarray gene expression are exactly in the opposite direction, i.e., generally in an experiment the number of samples taken is reduced. On the other hand, a microarray generates a huge amount of attributes, which at first are potential biomarkers for a particular phenotype.

For a better evaluation / identification of biomarkers it is necessary to increase the number of available samples, and this could be done integrating datasets from different sources and different platforms, but this process present obstacles at various levels.

The first decision that must be made when planning to make an integration of expression datasets from various sources is at what level this integration is intended. If the objective is to integrate data from microarrays with the same reference (same platform), we have a problem of data standardization and scaling, since there is always a difference in the background noise and amount of mRNA used between experiences [72]. To solve this issue, different methods can be applied, including multiple linear regression or some more complex methods such as quantile normalization, to make a scaling of the data to be a fit between datasets.

If microarray platforms are not the same, we will have a problem of feature mapping, since the probesets used on a platform differ from those used in the other. The correct annotation of probesets is a very important step in the process of integration of expression data. It is possible to combine the values of a probeset expression with another only if both have the same target. The interpretation of what is a probeset varies between manufacturers of microarrays. A platform can contain probesets which are associated with the same gene, but with completely different expression values [73] [15].

The manufacturers, themselves, provide annotation to their probesets, but with the constant evolution of platforms these files become quickly obsolete [32].

In fact, the target of the probesets are exons or sets of exons that comprise transcripts. That is, if two probesets have as target different transcripts from the same gene, it is normal that the measured expression values are different.

2.6.1 Previous works

Integrating multi-platform microarray data raises many difficulties, and therefore the authors who take this challenge focus on specific integration issues, and the methods used widely diverge. The main distinction is at which level the interface between platforms is made. Since the majority of the datasets available are at the probeset level, the first approach could be to directly map probeset values. However, the data files generally do not have sufficient information to achieve this mapping. The traditional methods integrate at the gene level, where it is possible to resort to probeset-gene mapping files available from manufacturers. However, several inconsistencies were pointed out for this approach and recent studies are turning to the transcript level [15].

Kostadinova [74] examines how the combination of several related microarray datasets affects different areas of preprocessing and analysis of gene expression data, such as missing value imputation, gene clustering and biomarkers detection. Meng et al. [75] use a top-level approach, with multiple co-inertia analysis to relate two datasets, by maximizing the covariance between eigenvectors in paired dataset analysis. With this method, they can integrate and compare multi-omics data, independent of data annotation. The majority of the authors also focus on a single vendor or on datasets that are only one-channel or two-channel based. Papiez et al. [76] combined data sets from two types of microarrays: oligonucleotide and cDNA. They extract a set of genes common for both platforms and remove batch effects obtaining combined p-values from the two experiments. Even being different platforms, both datasets considered only intensity values. Although being difficult, data integration has been seen as an important achievement, even leading to proprietary solutions, like the ones described by Willis et al. [77]. In this work, a combination of Thomson Reuters solutions was used, including the Data Annotation and Processing tool, MetaCore Pathway Analysis and the Biomarkers Module of Thomson Reuters Integrity for a simple workflow to process omics data and identify biomarkers of target engagement through pathway analysis and data integration.

Culhane et al. [78] made a co-inertia analysis to relate datasets without the need of identifiers annotation, using a multivariate method that identifies co-relationships in multiple datasets. The method was able to visually cluster genes with similar expression patterns between platforms. The authors claim that it is possible to assist in the selection of the strongest features from each dataset for subsequent analysis. This approach

prevents the bottleneck caused by the annotation based methods for integration, that limit the genes on study to those identified in all the datasets involved.

Woo et al. [79] compared the variability in expression between three different types of microarrays (Affymetrix GeneChip, cDNA and oligonucleotide). The gene identifiers and probeset-gene mapping were obtained from TIGR and GenBank. The expression value for each gene was obtained by averaging probesets linked to the gene, adding experimental effects like dye and array effects. The platforms were compared by concordance of the F-statistics using a 2x2 cross-classification of significant vs. non-significant genes. Affymetrix microarrays shown greater variation in the magnitude of expression between replicates. However, in terms of statistical significance, a greater concordance between Affymetrix GeneChips and oligonucleotide arrays was shown. This analysis was done at the gene level.

Jarvinen et al. [80] compared data between different expression microarrays, using Affymetrix, Agilent cDNA and a custom-made microarray from a cDNA library. The comparison was made at the gene-level, using the UniGene cluster ID as a common handle among platforms. The expression values of probesets with the same ID, called clones, were averaged. A total of 7936 identifiers for Affymetrix, 7117 for Agilent and 7273 in the cDNA microarray were identified, but only 2340 identifiers were in common among the three platforms. After processing information, the authors obtained 1147 identifiers in common, to which they applied the Pearson and Spearman correlation to compare the gene expression between platforms. The results indicate variability across the three platforms, being greater among commercial arrays.

Ballester et al. [32] established an annotation pipeline at the transcript-level. The probe sequences were collected from the Affymetrix site and were aligned with the genome using the Ensembl exonerate tool. An association probeset-transcript was made if at least 50% of their probes match the transcript, considering 1 bp as maximum mismatch. Probes with match in more than 100 different places on the genome were discarded. Aiming to improve annotation, the 3' untranslated regions (3' UTRs) were considered.

Bellis [73] pointed the problem of annotation dependency on the reliability of files provided by manufacturers with the association of probesets-transcripts-genes. The authors refer that the assumption that an exon read lead leads to a specific transcript is ambiguous, which leads to a need for extra validation. To analyze if two probesets annotated to the same gene are measuring the same entity, they used Pearson correlations, complemented by networks of positive and negative correlations. This

structure, compares all the probesets and genes and assigns them to different classes. The result was a full textual description of the association of each probeset to genes, exons and transcripts.

Choi et al. [81] advocated the integration of platforms by studying the size effect (difference of expression) resulting from various experiments at the expense of the application of conventional statistical methods. An index was established to measure the magnitude of the result of treatment or effect co-variable that was possible to compare between experiments. The model has been complemented with a Bayesian analysis of differential expression in each dataset. Focusing on the outcome of each separate experience, and integrating later, discards the problem of integration of identifiers.

Zhu et al. [82] based their work on gene expression correlation between platforms, thus preventing the decrease of identifiers that exist while crossing common genes between multiple platforms.

Sarmah and Samarasinghe [83] integrated Affymetrix data with cDNA data using regression (linear, polynomial-cubical, LOESS, bootstrap aggregating) and Artificial Neural Networks (self-organizing maps and feedforward networks), coming to the conclusion that the relationship between platforms is not linear, being better represented by feedforward neural networks. To homogenize transformed values Affymetrix $\log_2$ ratio. Bivariate linear regression is still used to test the strength and ability of linear model prediction. Many of the experiments are made at the level of gene, which does not allow to identify when each gene transcript is expressed and can lead to incorrect interpretations. For example, if a probeset measures the expression of an active gene exon, another probeset can measure the expression of an exon of the same gene that has a different behavior, leading to a general expression of the gene, with little meaning.

Yu et al. [15], aligned sequences of probes, transcripts and proteins through the International Protein Index, for the Mouse Genome 430A 2.0 Array and Human Genome U133A Array. The application of the new method has demonstrated an increased consistency of expression between synonymous probesets and strengthened the correlation between protein expression interactively. The probesets that hybridize with more than one transcript on the same gene are referred to as synonymous, with the level of the gene that hybridize with just one are referenced as synonymous to transcript level. Probesets were studied only associated with transcripts that encode proteins.

In addition to the above mentioned, there is still the problem of signal difference between platforms. To resolve this problem, Bossers et al. [72], proposed an approach

based on signal strength, instead of the traditional ratio-based, for dual-channel arrays, defending that the analysis of intensity is more reproducible and more sensitive in the detection of differential expression of genes. In support of this approach, the authors indicate that the current dual-channel arrays platforms have a quality so high that the variance inter-array became much less pronounced.

It is rare to find a work that integrates data from platforms with a different measurement basis, like the log intensities and log ratios.

There are several studies that were dedicated to the annotation of probesets. However, their results are limited to some probesets or just some arrays. The methods used range from the application of rules from simple statistics (average, maximum, ...), to associate various probesets of the same gene, to the interpretation of the name assigned to each probeset as an indicator of the likelihood of cross-hybridization that probeset [15]. It is also frequent to find analyses by results of experiments, trying to find correlations and indexes that can compare entities that were not known at the outset.

3 Methodology

In this chapter, the principal guidelines to be adopted on the construction of the required system will be presented. Some algorithms of key features and the mathematical considerations for some tasks will be described.
The methodology will be presented oriented to the different stages of development, to a better understanding of the context.

3.1 Summarization of the problem

The major issue for statistical analysis of microarray data is dimensionality. In a typical experiment, we obtain a table with thousands of genes for a small number of samples [21]. This leads to situations where it is difficult, or even impossible, to employ classical prediction algorithms leading to poor prediction accuracy in discriminant models [84]. While a great number of studies have focused on data dimensionality reduction [85], applying statistical methods like Principal Component Analysis, Linear Discriminant Analysis or k-Nearest Neighbors, other works turned their attention to the increase in the number of samples [81]. One way to increase sample size is the integration of microarray data from different studies over the same phenotypes. Data integration from multiple sources aims to improve the assertiveness of statistical tests, reducing the data dimensionality problem. When the platforms have the same identifiers, this implies a data adjustment to normalize inter-platform data levels. When the platforms have different variable identifiers (i.e. different probesets), a reannotation method is needed to make a bridge between them.
The first focus of this work is on the reannotation process, a cornerstone of multi-platform integration. The proposed approach is based on a reannotation from probesets to transcripts, preserving valuable information for further analysis.

The next step addresses the stages involved in the process of selecting a good prediction tool, based on microarray gene expression data. The information integration from heterogeneous platforms is highlighted, evidencing the particularities of choosing approaches working at gene, transcript or probeset levels. The strategy involves testing distinct machine learning algorithms and integration methods, analyzing their ability to produce a good set of predictors for the selected phenotype.

3.2 Reannotation

The reannotation process addressed here, aims to achieve a universal procedure that allows the translation of probesets from distinct microarray platforms, into a common backbone. Data reannotation from multiple sources of gene expression records can be done in four main steps:

1. **Data collection**. Consists in gathering data from different sources, normalizing the types and values into tables with useful unambiguous information.
2. **Mapping**. Consists in making a bridge between identifiers from different data tables. Originally, these identifiers could be gene names, probeset names/IDs, etc. On probeset-oriented data sets, each microarray manufacturer has its own identifiers. It is then necessary to perform a reannotation to achieve a common backbone of identifiers for all data sets. The consistent homogenous annotation of probesets is a very important step to compare expression data from different platforms. It will only be possible to combine the values of expression of a probeset with another one if both have the same target. The manufacturers provide annotation to their probesets, but with the constant evolution of platforms, this is quickly outdated [32].
3. **'Clone' treatment.** Consists in deciding what to do with probesets that read the same gene. The interpretation of what a probeset measures diverges between platforms. Typically, there are distinct probesets associated with the same gene, but these often result in very different expression values for the same phenotype [15].
4. **Model validation**. Consists on comparing data from different platforms with the new annotation to conclude about the possibility of integration.

3.2.1 Data collection

The first decision to be made is at what level to perform the data integration: at microarray spot level, probeset level, transcript level or at the gene level. An important factor is at which level the data are available. If data are on a higher level, then it is not possible to descend to a lower level. This stage also encompasses scale correction of expression values, as some are on a logarithmic scale and others are on an absolute scale. As most differential expression tools use the logarithmic scale, this will be chosen. The data should be available online, on open access databases, being the methods used to extract information intrinsic to the means and tools available from the respective database.

A more intricate problem is the standardization of phenotypic metadata. Clinical data are often provided in plain text files, which lack normalization, where it is common to have distinct information together on the same field. This leads to a need of manual intervention to achieve homogeneous and unambiguous information that associates gene expression levels to specific phenotype attributes.

3.2.2 Mapping

Using available maps

Some manufacturers provide files with the probeset-gene correspondence. However, these files do not consider the probesets specificity and lack information for some probesets. This is also a flawed method, because these files are not always available, and they can be out of date. With the release of new platforms, with new names for probesets, these files become obsolete. Other maps have the disadvantage of being static, so it is not possible to do much with them if they have not the desired information.

Sequence alignment

To grant full power on the available information about probesets, it is necessary to descend to the more basic information available, that is, the probe sequence. Having this information, it is possible to perform and have control on our own mapping methods.

Since Affymetrix has different probe sequences on the same probeset, it is necessary to collect sequence data for each probe. For Agilent and Illumina, the sequence is the same for all probes of the same probeset.

Using a sequence alignment tool, it is possible to obtain the locations of the genome where the respective probe finds correspondence, i.e., the respective chromosome, position (start an end) inside that chromosome and strand. The simplified alignment algorithm is the following:

```
for each chromosome
        for each strand
                probesMatches <- getProbesMatches
                starts <- getMatchesStartingPoints
                ends <- getMatchesEndingPoints
                saveOnDatabase(probe_id, chromosome, start, end, strand)
        end for
end for
```

This information is then crossed with gene, transcript and exon information to determine exactly to which of these structures the probe matches.

The probe-gene mapping algorithm is the following:

```
for each chromosome
        chrMatches ← probeMatches(chromosome)
        for each chrMatch
                geneMatches ← probeMatches(genes on this chromosome)
                if exists geneMatches
                        save probe-gene map info in DB
                else
                        save probe match as "extra_gene" in DB
                end if
        end for
end for
noMatches ← get probes with 0 matches
save noMatches as match in gene "---" in DB
```

In this way, it is possible to determine if there is a good probability of cross-hybridization for each probe, which allows to assign them a value of specificity.

3.2.3 Clone treatment

With data at the probeset level, it is possible to do reannotation at gene or transcript level. The most simple approach is to use a mapping file to gather which probesets are associated with which genes and perform a gene wise mean of these probeset values. If the sequence alignment of probe to genome association is made before, the wrong associations are ideally discarded. Adding the cross hybridization counting, it is possible to decide when a probeset is used to calculate the overall expression value for that gene or, when used, to define a significance factor.

For the probeset-transcript association, the same principle also applies. For the contribution ratio of a probeset to a transcript expression value, the number of matches that the probeset makes with the transcript will be used, divided by the total number of matches on the whole genome Eq. 14.

$$\boldsymbol{psetR_{(ij)} = \frac{npm_{(ij)}}{npm_{(ij)} + npn_{(i)}}} \qquad \textbf{Eq. 14}$$

where*:*

psetR - probeset ratio for this transcript;

npm - number of probe matches on the transcript;

npn - number of probe matches outside the transcript*;*

i - probeset index*;*

j - transcript index.

For the determination of *npn,* it is necessary to check if each hit is outside the bounds of the target transcript, because a probe may have a hit in another transcript that has overlapping sections and, therefore, be identified erroneously as an outside match. When it is not possible to find the probe sequences, or when a probe does not have matches, it will be discarded.

A basic algorithm for probesets-transcripts mapping could be:

```
for each transcript
        for each probeset
                trHits ← get total hits on transcript
                totalHits ← get total hits
                if trHits > 0
                        ratio ← trHits / totalHits
                        insert mapping info on DB
                end if
        end for
end for
```

A transcript expression value will be the weighted average of corresponding probesets Eq. 15:

$$\boldsymbol{te_{(j)} = \frac{\sum_{i=1}^{n}\left(\boldsymbol{psetV_{(ij)} * psetR_{(ij)}}\right)}{\sum_{i=1}^{n}\boldsymbol{psetR_{(ij)}}}} \qquad \textbf{Eq. 15}$$

where:

te - transcript expression;

psetV - probeset expression value;

n - number of probesets that match transcript *j*.

3.2.4 Steps of the reannotation process

The reannotation process consists on the following specific steps:

1) Use an open database with genome assembly to collect the information about the genome (e.g. location of genes, transcripts and exons);

2) Collect, from the available datasets, the probeset identifiers and each probe's target sequence;

3) Match the probe sequences with the genome, using a sequence alignment tool;

4) Cross information from 1 and 3 to determine which exons are read by each probe;

5) Associate matched exons to respective transcripts or genes;
6) Calculate the probeset ratio;
7) Calculate the expression value by transcript or gene;
8) Restructure the datasets with the new annotation and with adjusted expression values;
9) Validate the model.

3.3 Data integration

Data integration is the main objective of this work. With that in mind, it is intended to obtain richer datasets, with more explanatory samples and thus, with greater generalization capabilities. Having solved the inter-platform features common annotation, there are still other significant challenges.

Even if two datasets are obtained using the same platform, they inevitably suffer batch effects. Batch refers to microarrays processed at one site over a short period of time, using the same platform [86]. The non-biological variation between batches could be a simple constant noise signal and, in this case, linear models are sufficient to correct that variation. Taking a group of samples with the same outcome from one of the datasets (the reference), another group of samples with similar outcome from another dataset, it is possible to obtain a relation point. It is necessary to take other two groups of samples from the same datasets to obtain the second relation point and then obtain the *a* and *b* coefficients of a linear model. The model is then applied to all data of the second dataset, aiming to adjust it to reference dataset Eq. 16.

$$\boldsymbol{A_{(ij)} = \frac{O_{(ij)} - b}{a}} \qquad \textbf{Eq. 16}$$

where:

O - Original expression value;

A - New expression value;

i - Sample index;

j - Feature index.

When the variance factor is not constant, or even not linear, the problem is more puzzling, and more sophisticated methods are required, like SVMs and ANNs. Johnson et al. [87] proposed parametric and nonparametric empirical Bayes frameworks for adjusting data for batch effects that is robust to outliers in small sample size batches. They called it ComBat, and its available in 'sva' Biocondutor package.

Choosing the group of samples to use on the model construction is not trivial either. It is not rare to find datasets with only one phenotype characteristic discriminated, even if discarding the different phenotype interpretations that could be done by each researcher. Having this into account, some studies put forward the idea of using housekeeping genes to compare divergences between batches [88] [89]. Housekeeping genes are those that maintain their expression values relatively unchanged in all cells of an organism. Thus, theoretically, they are the perfect features to detect non-biological changes. However, this is not so linear, since these genes may present significant variations under certain conditions. A variation from this approach can be done identifying the genes that do not present significant variation for a specific feature. This can be done applying ANOVA or Pearson's correlation between expression fold change and variation on the phenotype outcome.

Another level of complexity is achieved when trying to integrate data from heterogeneous platforms, like single channel and dual channel. Taking as examples Affymetrix and Agilent, we may consider, with a certain level of error, that we have:

$$\begin{aligned} \boldsymbol{Expression}_{\boldsymbol{Affy}} &= \boldsymbol{log_2}\left(\boldsymbol{I_{Affy}}\right) \\ \boldsymbol{Expression_{Agil}} &= \boldsymbol{log_2}\left(\frac{\boldsymbol{I_{Agil}}}{\boldsymbol{I_{CellLine}}}\right) \end{aligned} \qquad \textbf{Eq. 17}$$

where:

I_{Affy} - Affymetrix expression intensity;

I_{Agil} - Agilent expression intensity;

$I_{CellLine}$ - Cell line expression intensity.

Since we have here different scales, it is necessary to implement a transformation to obtain equivalent values. So, the objective is to isolate I_{Affy} and I_{Agil} to obtain intensity comparable values. The big problem here is that Agilent datasets at probeset level have only the final ratio value, being impossible to separate I_{Agil} from $I_{CellLine}$.

When the nature of the problem we wish to solve is classification, a reference class of samples could be used to get an adjusted ratio (*Ar*) for both platforms.
For Affymetrix, we would have:

$$\begin{cases} Exp = log_2(I) \\ Exp_{\ ref} = log_2(I_{ref}) \end{cases} <=> \begin{cases} I = 2^{Exp} \\ I_{ref} = 2^{Exp_{\ ref}} \end{cases}$$
$$Ar = \frac{I}{I_{ref}} = \frac{2^{Exp}}{2^{Exp_{\ ref}}} \qquad \textbf{Eq. 18}$$

While, for Agilent, the expressions are given by:

$$\begin{cases} Exp = log_2\left(\frac{I}{I_{CellLine}}\right) \\ Exp_{\ ref} = log_2\left(\frac{I_{ref}}{I_{CellLine}}\right) \end{cases} <=> \begin{cases} \frac{I}{I_{CellLine}} = 2^{Exp} \\ \frac{I_{ref}}{I_{CellLine}} = 2^{Exp_{\ ref}} \end{cases}$$
$$Ar = \frac{\frac{I}{I_{CellLine}}}{\frac{I_{ref}}{I_{CellLine}}} = \frac{I}{I_{ref}} = \frac{2^{Exp}}{2^{Exp_{\ ref}}} \qquad \textbf{Eq. 19}$$

The Cell Line factor is eliminated on Agilent data, obtaining a comparable expression ratio. The various transformations and possible sources of noise that an expression signal may suffer, makes it an entity that not always behaves like expected. So, it would be unwise to defend a unique method for all cases of data integration. In this line of thought, the best strategy to adopt is to try a set of processes to verify empirically their adequacy for the case under study.

3.4 Validation with supervised learning

To validate and test the integration methods described on the previous section, it is interesting to use the integrated data to perform prediction tasks and compare the results with the ones obtained with the individual original datasets. For each model, a feature selection process is done, in tandem with the training of some machine learning algorithms and their use in predictive tasks, to obtain a set of comparable results.

3.4.1 Filtering

The overall aim of the process is to select the best pair of training algorithm - feature set, which could be achieved testing all possible algorithms against all possible combinations of feature sets. Obviously, this would result on a huge time consuming processing burden given the typical number of genes involved in these datasets. Then, the first step is typically to do a filtering stage, to reduce the initial set to a more reasonable subset. Two filtering methods will be used in this work: the first is based on the Pearson's correlation of each feature with the phenotype (dependent variable or class); the second will be based on differential gene expression tests using the empirical Bayes (eBayes) method, which will provide a rank with p-values of the features that best differentiate the phenotype classes. Since some algorithms do not deal well with data collinearity, features that have a high Pearson's correlation among them will be eliminated. From this stage, only the 200 features with best results are preserved, forming a reduced feature set (rfSet).

3.4.2 Wrapper

With rfSet, a greedy algorithm will be applied to choose, for each classifier, the best feature subset:

```
bestFeatureCombinations ← NULL
setSize ← 1
while(setSize <= maxSetSize)
newCombinations ← allCombinations(bestFeatureCombinations, rfSet)
        model ← trainAlgorithm(newCombinations)
        errorRate ← LOOCV(model)
bestFeatureCombinations ← bestNewCombinations(errorRate)
        setSize ← setSize + 1
```

The first iteration is made with sets with only a single feature, testing each algorithm with each of the rfSet features. The features which present the best results will be kept and then used as the base for the next set of tests, where an additional feature, from the rest of filtered features, is included to organize sets of 2-features combinations. This

method is repeated until reaching the *maxSetSize* threshold. The test criterion will be the lower classification error rate (number of misclassifications over total number of samples) by Leave One Out Cross Validation (LOOCV).

The classifiers to use on the algorithm are that ones with broader application on classification tasks, like kNN, SVM and LDA.

Firstly, the method will be applied separately to Affymetrix and Agilent datasets, to verify the strength of learning with relative homogeneous datasets. Next, the two types of platforms will be integrated and the same method will be applied to verify the differences. The integration will be made in six different ways:

- *Raw*, where the datasets are joined without any transformation;
- *Norm*, where the data is mean subtracted and divided by standard deviation;
- *Linear*. A set of reference features are selected, that present the lowest dataset and inter-grade expression variation in all datasets. When joining two datasets, one is used as base and a linear model is built that represents the linear variation between expression values of references from the base and the references of the new dataset. That model is then used to adjust the entire values of the new dataset. This is applied to linear data, without log2 transformation;
- *LinearLog2*. The same as the previous, but applied to log2 transformed data;
- *Grade*. Since the grade is an ordered value, dividing all data by the mean of values of common grade brings the heterogeneous datasets to levels that are comparable. This is applied to linear data, without log2 transformation;
- *GradeLog2*. The same as the previous but applied to log2 transformed data.

4 Application to gliomas gene expression data

In this chapter, the application of the proposed methodology on a specific case study, will be presented, related to the determination of astrocytoma WHO grade, based on expression data. A brief description of the case study will be made here. An introduction to the gene expression databases used, their methods to access data and the structure of that data will be provided.
Each section of the integration system will be highlighted, providing the results obtained at each stage.

4.1 Case study

Among brain cells, glial cells have functions that support the proper functioning of neurons. They intervene on the regulation of the chemical and physical conditions of brain environment, formation of myelin, and provide protection for neurons in the brain and peripheral nervous system. Astrocytes are the most frequent glial cells, making the connection between neurons and the blood stream. Among their functions are the regulation of the chemical environment of neurons and recycling neurotransmitters released during synaptic transmission. Astrocytoma, a tumor with its genesis on astrocyte cells, is the most frequent glioma, a family of brain tumors that occurs on glial cells.
The World Health Organization (WHO) classifies astrocytomas in four levels, being the increase in level directly associated with the increase in tumor malignancy [90] [91]:

- **WHO grade I**. The pilocytic astrocytoma is considered low grade and is the only level of the tumor passive to be benign, since it does not evolve to higher levels. Generally, they are circumscribed and often cystic with solid mural nodules. Their fiber-like appearance is responsible for the tumor to be named

as pilocytic, but some parts of the tumor may not fit in this description. This grade of tumor is characteristic of children and young adults;

- **WHO grade II**. The diffuse astrocytoma is also considered low grade, but since it evolves to higher levels, it is a malignant type of cancer. The ages of focus are similar to that of grade I. The principal characteristic of this tumor is to be difficult to determine, by imaging, direct observation during surgery, or by gross pathological examination where the tumor ends and normal tissue begins. Survival expectation for patients with this grade of astrocytoma is 3 to 10 years;
- **WHO grade III**. The anaplastic astrocytoma is high grade, and evolves to grade IV. Survival expectation is 2 to 5 years;
- **WHO grade IV**. Glioblastoma multiform (GBM) is a high-grade astrocytoma that almost inevitably causes death. The therapies being applied just extend the patient life expectation a bit more, but even with surgical intervention and other therapies, the median survival expectation for a patient with glioblastoma is about 12 months [92]. It is generally detected on middle age adults and is visually identified by cellular and nuclear anaplasia, micro vascular proliferation, and necrosis (Figure 8). If the tumor arises *de novo* as GBM, it is called primary glioblastoma, if it is an evolution from a low grade glioma, it is called secondary glioblastoma.



Figure 8. Macroscopic pathology of Glioblastoma multiform (Source http://it.wikipedia.org/wiki/ Glioblastoma; April 2015).

Besides the WHO grade classification, other molecular classifications of GBM are pointed by some authors, like: proneural, proliferative, and mesenchymal; or classical, proneural, mesenchymal and neural [93]. A clear direct relation between classifications has not been established.

Oligodendrogliomas represent 5 to 6% of gliomas [91] and appear in some GBM studies. They are generally more circumscribed than astrocytomas, but may in some cases contain neoplastic astrocytes and, in those cases, are called oligoastrocytomas. Oligodendrogliomas may be classified as grade II or grade III.

The correct classification of the pathology of a patient, in particular in cancer, is essential to decide which therapies may be applied [25]. The diagnosis of cancer is usually done through microscopic and immunological tests over tissues biopsies, an extremely invasive process. Often, tumor samples with an atypical morphology complicate the analysis. In addition, certain types or subtypes of tumors may have very little differentiation between them [22]. The analysis based on gene expression is extremely important, given the gaps that traditional diagnosis methods still present. Since tumor grade is the most established and robust predictor of disease outcome [94], biomarkers with expression levels correlated with tumor grade would be valuable. Several studies and experiments have been made on the epigenetic approach to understand the cancer mechanisms and forms of prognostic and diagnostic [90] [92] [94].

4.1.1 Gene expression on open-access databases

There are large amounts of gene expression data available on public databases. For example, in Gene Expression Omnibus (GEO) and The Cancer Genome Atlas (TCGA) is possible to find several studies about glioblastoma. However, each dataset comes from an independent experiment and has few available informative samples. The result of a statistical analysis on one of these datasets suffers from a lack of a good representative sample set. To improve the strength of these tests, it is necessary to integrate data from several sources, but this approach leads to complex challenges.

These databases work with contributions from several particular sources, as seen in the work of Costa et al. [92]. This work was based on HOXA genes that encode critical transcriptional regulators of embryonic development that have been implicated in

cancer, namely the relevance of their activation mechanisms in glioblastoma. The Kaplan-Meier method was used to estimate overall survival. Through this method, it is possible to compare the expected lifetime of patients with different gene expression profiles. It was shown that there is a relevant correlation between some HOXA genes expression and the evolution of glioblastoma. The gene expression data was stored in the GEO database, with the reference GSE4271.

Phillips et al. [94] analyzed 76 samples from newly diagnosed cases of WHO grade II and IV astrocytoma, profiled via DNA microarrays. Groups of genes with expression profiles strongly related with survival were identified and, from those, three distinguished groups of samples were identified by clustering. The 35 most robust markers for the three groups were finally selected. New samples were classified by centroid classification. The data used was also from GSE4271.

Gene Expression Omnibus

GEO is a public functional genomics database supporting MIAME-compliant data submissions (http://www.ncbi.nlm.nih.gov/geo/), maintained by the National Center for Biotechnology Information (NCBI). Data in this repository are organized on four main structures: samples, series, datasets and platforms.

A series is identified by the prefix 'GSE' and comprises all the information about a respective study. It may involve several patients with distinct stages of a pathology and multiple microarray structures. The information in a series is submitted, in an open manner, by the researchers responsible by the study, which have only to respect the general MIAME-compliant format of data. This gives place to a loose structure that needs manual treatment to be normalized. Sometimes, it is even harder to find the desired information due to particular medical-oriented manner to expose data.

To surpass this lack of organization, GEO supplies the datasets, identified by the prefix 'GDS'. This format of data is curated by GEO staff and aims to establish some data structure norms like, for example, each dataset only have data measured with the same type of microarray in the same experimental conditions.

A sample, with prefix 'GSM', corresponds to one array measurement for some subject under study on a certain pathology condition. Series and datasets are composed by a group of samples.

A platform stores information for a specific microarray, about their probesets. Here, it is possible to find probeset specific information, like their target and foreign keys to get more information on other databases, like GenBank.

More detailed information about the GEO project can be found in [95] [96] [97].

The Cancer Genome Atlas

TCGA is maintained by a partnership between the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) and gives support to studies concerning a limited set of tumors. It intends to improve the understanding of the molecular basis of cancer through the application of genome analysis technologies, including large-scale genome sequencing (http://cancergenome.nih.gov).

TCGA samples are collected from voluntaries and treated by Biospecimen Core Resources to meet the desired criteria for publication. Genome Data Analysis Centers provide information processing, analysis and visualization tools and the final data is organized and made available by the Data Coordinating Center. TCGA users that want to get data, may access the Data Portal and retrieve a disease-related set of data batches. Here, we found data in the three levels (probe, probeset and gene) and it is possible to obtain, plus gene expression, clinical files for each biospecimen (sample), among other types of information.

Biospecimens are identified by barcodes and Universally Unique Identifiers (UUID). Even though TCGA are transiting to the second form, the barcodes could be useful by itself, since their component sections have meaning, unlike UUIDs. The most complete form of the barcode is the aliquot, that can be seen in Figure 9.

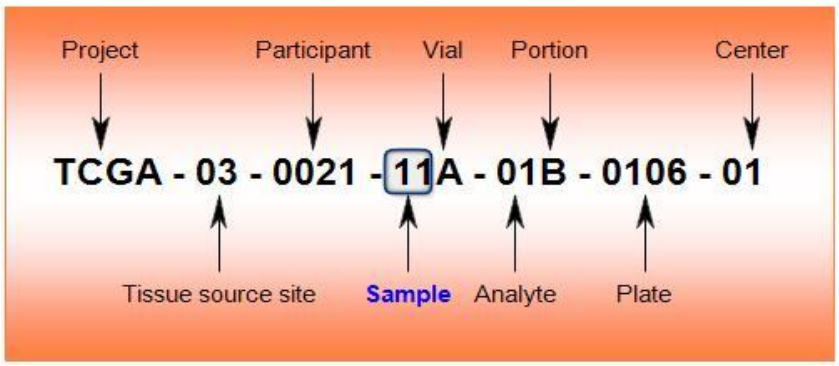


Figure 9. Example of a TCGA aliquot barcode where it is indicated the meaning of each section.

This information could be useful when some basic phenotype information is not clear on clinical data. More specifically, the sample section indicates the generic state of the patient pathology. For example, "01" in sample section means "Primary Solid Tumor" and "11" means "Solid Tissue Normal".
More detailed information about TCGA project can be found in [98].

Rembrandt

The REpository for Molecular BRAin Neoplasia DaTa (Rembrandt) is an open access knowledgebase framework, created by Ann Wiley and supported by NCI (https://caintegrator.nci.nih.gov/rembrandt). This data aggregation can be used to query, analyze, and visualize features to uncover critical correlations among diverse types of data. Beside the online tools, it is possible to download gene expression and phenotypic information for several gliomas, based on the Affymetrix platform U133 Plus 2.0.

Ensembl

Ensembl is a project held by European Bioinformatics Institute and Wellcome Trust Sanger Institute (http://www.ensembl.org), driven by an independent scientific advisory board. This database is not a repository for gene expression datasets, but comprises genome information for vertebrates and other eukaryotes species, which is essential to integrate gene expression data. The main goal of the Ensembl project is to automatically annotate the genome, integrate this annotation with other available biological data and make this publicly available via web. Ensembl site provides, among other functionalities, BLAST/BLAT online tool for DNA sequencing and BioMart for genome annotation retrieval.
More detailed information about Ensembl project can be found in [99].

4.2 Data collection and pre-processing

To apply the data integration methods, and test the results, it was necessary to collect data from several sources, with different microarray structures. The premises and keywords for data searching were:

- Public available data and metadata;

- Must be focused on astrocytoma / glioblastoma;
- Must have the WHO grade classification available;
- Must be possible to access information about the microarray platform;
- Other phenotypic information could be useful in future, like survival, gender, etc.

4.2.1 Gene Expression Omnibus

From GEO, the search was focused on GDS, since these structures are more organized and it is easier to retrieve their information. After applying a filter for glioblastoma, these where the datasets found: GDS2853 [100], GDS3069 [101], GDS1962 [102], GDS1815 [94], GDS1816 [94], GDS1975 [103] and GDS1976 [103]. All these datasets were associated to Affymetrix platforms. The series GSE19114 [104], associated with an Illumina platform was also collected.

GDS2853

Title: Low and high-grade astrocytomas

Summary: Comparison of low and high-grade astrocytoma brain tumors. Results provide insight into the molecular differences between the two types of tumors [11].

Platform: GPL8300 - [HG_U95Av2] Affymetrix Human Genome U95 Version 2 Array

Sample count: 16

Expression values processing: The expression values were on direct expression count format and there were missing values. An algorithm was applied, to fill missing values and then a $\log_2$ transformation was performed.

Phenotype: Information collected from GDS2853 and the transformation performed is provided in Table 1.

Table 1. Phenotype data transformation for dataset GDS2853.

Original		New		
disease state	**other**	**tumor_type**	**grade**	**source**
8 low grade astrocytoma	tumor	astrocytoma	low_grade	tumor
6 high grade astrocytoma	tumor	astrocytoma	high_grade	tumor
2 control	cell line	astrocytoma	high_grade	cell_line

An analysis of the work associated with this dataset was needed, to be sure that low grade refers to WHO 1 and 2 grade, whereas high grade refers to WHO 3 and 4. The cell lines were obtained from high-grade gliomas, therefore received this rating. As the cell lines may differ expression in relation to the original tissue [105] [106], it was decided to consider a phenotype attribute "source" that would identify them.

GDS2853 is associated with the GSE3185 series, and its analysis showed that there were more samples in the same study, but obtained with a different microarray. Thus, through the series platform matrix, it was possible to obtain more 3 samples:

Platform: GPL96 - [HG-U133A] Affymetrix Human Genome U133A Array

Sample count: 3

Phenotype: The phenotype attributes for these samples also have to be collected from GSE3185 platform matrix. These data did not show the same structure as the previous and therefore had to be manually inserted Table 2.

Table 2. Phenotype data transformation for samples in GSE3185 series that are not on GDS2853.

Original		New		
disease state	**other**	**tumor_type**	**grade**	**source**
3 low grade astrocytoma	tumor	astrocytoma	low_grade	tumor

GDS3069

Title: Various brain tumors

Summary: Analysis of 12 primary brain tumor biopsies with some variation in their histological diagnoses. These results, together with those obtained from miRNA profiling by real-time PCR, provide insight into the relationship between endogenous fluctuations in miRNA and mRNA expression levels [11].

Platform: GPL96 - [HG-U133A] Affymetrix Human Genome U133A Array

Sample count: 12

Expression values processing: The expression values were on direct expression count, so, a $\log_2$ transformation was performed

Phenotype: The phenotypic information was collected from GDS3069 Table 3.

Table 3. Phenotype data transformation for dataset GDS3069.

Original	New	
disease state	**tumor_type**	**grade**
2 low grade anaplastic mixed glioma	anaplastic_mixed_glioma	low_grade
6 high grade glioblastoma	glioblastoma	high_grade
1 high grade glioblastoma/gliosarcoma	glioblastoma_gliosarcoma	high_grade
2 high grade gliosarcoma	gliosarcoma	high_grade
1 low grade oligodendroglioma	oligodendroglioma	low_grade

GDS1962

Title: Glioma-derived stem cell factor effect on angiogenesis in the brain

Summary: Analysis of gliomas of different grades. Stem cell factor (SCF) is overexpressed by glioma cells, but its role in gliomagenesis has been unclear. Results identify a positive correlation of SCF expression with increasing glioma grade, which supports a role for SCF in tumor angiogenesis in the brain [11].

Platform: GPL570 - [HG-U133_Plus_2] Affymetrix Human Genome U133 Plus 2.0 Array

Sample count: 180

Expression values processing: Null values treatment and $\log_2$ transformation

Phenotype: The phenotypic information was collected from GDS1962 Table 4..

Table 4. Phenotype data transformation for dataset GDS1962.

Original		New	
disease state	**tissue**	**tumor_type**	**grade**
23 non-tumor	not applicable	normal	0
26 astrocytomas	7 tumor grade II	astrocytoma	2
	19 tumor grade III		3
81 glioblastomas	81 tumor grade IV	glioblastoma	4
50 oligodendrogliomas	38 tumor grade II	oligodendroglioma	2
	12 tumor grade III		3

The 23 non-tumor samples were obtained from epilepsy patients.

GDS1815 and GDS1816

These two datasets have the same series associated (GSE4271). In this study, the same samples were measured by two Affymetrix platforms that are complementary, the Human Genome U133A Array and the Human Genome U133B Array.

Affymetrix documentation shows that U133A considers the traditional genes that were already measured on older platforms and U133B considers a new set of probesets referring to newer gene studies, presenting the relation shown on Figure 10 [107].

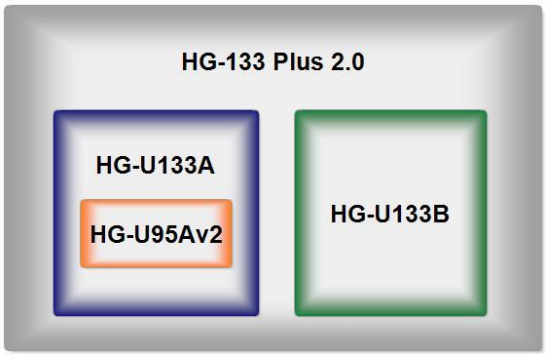


Figure 10. Affymetrix probesets relation between platforms.

Even though HG-U95Av2 probesets nomenclature is found on HG-U133A, these probesets are not identical, since on the former platforms a probeset has 16-20 probe pairs, and in the HG-133 set a probeset has 11 probe pairs [108].

Since the experiments were made on the same conditions, it was decided to join them in one dataset, called GDS1815_6. But even the "same conditions" are not a guarantee of batch effect inexistence. When crossing probeset names from HG-U133A and HG-U133B, it was detected that some of them are equal on both platforms. These probesets were used to make a connection with the two platforms, smoothing at the same time the possible non-biological differences between them. A linear model was applied over each column of the intersect section on one dataset, to predict the columns of the intersect section on the other dataset. The F-statistics computed for each model show an evident relation between them, being the maximum p-value of 2.2e-59. Each model was then applied on the columns of the complete dataset to obtain GDS1816 adjusted to GDS1815. In the case in which the probesets are not repeated from one platform to the other, a simple junction was made while retaining the original and the adjusted expression values. In the case of common probesets, a pair-wise average of expression values between the two platforms was done. As the nomenclature of the samples varies from one dataset to another, the samples of GDS1815 were chosen as base for GDS1815_6.

Title: Glioma-derived stem cell factor effect on angiogenesis in the brain

Summary: Analysis of high-grade glioma (HGGs) samples from cases of WHO grade III and IV astrocytomas. Results classify HGGs into molecular subclasses with prognostic value that predict survival and disease progression. The molecular subclasses resemble key stages of neurogenesis [11].

Platforms:

GDS1815 → GPL96 - [HG-U133A] Affymetrix Human Genome U133A Array

GDS1816 → GPL97 - [HG-U133B] Affymetrix Human Genome U133B Array

Sample count: 100

Expression values processing: Log_2 transformation

Phenotype: The phenotypic information was collected from GDS1815 (that is equivalent to the information on GDS1816) Table 5.

Table 5. Phenotype data transformation for datasets GDS1815 and 1816.

Original				New	
disease state	**specimen**	**cell type**	**other**	**grade**	**survival**
24 WHO grade III	21 primary	20 proneural	3 "less than 52 wk survival"	3	182
			2 "52 to 103 wk survival"		542
			3 "104 to 154 wk survival"		903
			2 "155 to 209 wk survival"		1274
			10 "more than 209 wk survival"		inf
		1 mesenchymal	"less than 52 wk survival"		182
	3 recurrent	2 proneural	"unknown"		unknown
		1 mesenchymal			
20 WHO grade IV	recurrent	5 proneural	unknown	4	unknown
		4 proliferative			
		11 mesenchymal			
6 WHO grade IV without necrosis	primary	4 proneural	1 "52 to 103 wk survival"	4	542
			1 "104 to 154 wk survival"		903
			2 "more than 209 wk survival"		inf
		2 mesenchymal	2 "52 to 103 wk survival"		542
50 WHO grade IV with necrosis	primary	6 proneural	1 "less than 52 wk survival"	4	182
			1 "52 to 103 wk survival"		542
			2 "104 to 154 wk survival"		903

			1 "155 to 209 wk survival"		1274
			1 "more than 209 wk survival"		inf
		24 proliferative	8 "less than 52 wk survival"		182
			11 "52 to 103 wk survival"		542
			2 "104 to 154 wk survival"		903
			3 "more than 209 wk survival"		inf
		20 mesenchymal	3 "less than 52 wk survival"		182
			8 "52 to 103 wk survival"		542
			5 "104 to 154 wk survival"		903
			1 "155 to 209 wk survival"		1274
			3 "more than 209 wk survival"		inf

For the survival attribute, the weeks were converted to days, and the middle point of the interval was calculated.

Even if not in the table, the necrosis state, specimen and cell type were also saved, that basically follow the corresponding original attributes values.

GDS1975 and GDS1976

These two datasets were also produced with HG-U133A and HG-U133B, in the same conditions, so the same considerations of the previous section apply.

Title: Gliomas of grades III and IV

Summary: Analysis of grades III and IV gliomas of various histologic types. Results used to develop a gene-expression based, histology independent-classification scheme, and provide insight into the biology of gliomas [11].

Platforms:

GDS1975 → GPL96 - [HG-U133A] Affymetrix Human Genome U133A Array

GDS1976 → GPL97 - [HG-U133B] Affymetrix Human Genome U133B Array

Sample count: 85

Expression values processing: Log_2 transformation

Phenotype: The phenotypic information was collected from GDS1975 (that is equivalent to the information on GDS1976) (Table 6).

Table 6. Phenotype data transformation for datasets GDS1975 and GDS1976.

Original		New	
disease state	**cell type**	**tumor_type**	**grade**
26 grade III glioma	8 anaplastic astrocytoma	astrocytoma	3
	7 anaplastic mixed oligo-astrocytoma	oligodendroglioma_astrocytoma	
	11 anaplastic oligodendroglioma	oligodendroglioma	
59 grade IV glioma	glioblastoma	glioblastoma	4

For all datasets, a "description" field was also added, to store general information present on the original datasets (e.g. " brain, Right Anterior Temporal: Extract52_le1 ")

GSE19114

Title: A transcriptional module initiates and maintains mesenchymal transformation in brain tumors [human data]

Summary: SNB19 human glioma cells and GBM-derived brain tumor initiating cells were transduced with STAT3 and CEBPB shRNA lentiviral vectors [11].

Platform: GPL6947 - Illumina HumanHT-12 V3.0 expression beadchip

Sample count: 74

Expression values processing: According to the authors of the study, data were obtained with an iScan BeadArray scanner (Illumina) and standardization focused on stabilization of variance and robust spline normalization with R lumi package [109]. This package uses classes based on ExpressionSet class, which handles expression values on $\log_2$ scale, thus, no additional transformations were done on the values.

Phenotype: This is the source type from where it is more difficult to extract information, as being the authors who introduced the information, it can take quite distinct and non-standard formats. The phenotype data of GSE19114 have several fields without relevant information for this work. The only field used was "characteristics_ch1" (Table 7).

Table 7. Phenotype data transformation for series GSE19114.

Original	New		
characteristics_ch1	**tumor_type**	**grade**	**source**
44 cell type/line: BTIC	glioma	low_grade	brain_tissue
30 cell type/line: SNB19	glioblastoma	high_grade	cell_line

In the absence of more information, all samples in this series were considered as tumors, being BTIC (brain tumor initiating cells) samples taken directly from the human

brain tissue and SNB19 grade IV glioma cell line derived from the resection of a glioblastoma multiform.

4.2.2 Rembrandt

Title: Rembrandt gliomas data repository

Platform: [HG-U133_Plus_2] Affymetrix Human Genome U133 Plus 2.0 Array

Sample count: 538

Expression values processing: Rembrandt allows to download CEL files, after selecting the desired type of glioma. These data files are spot (probe) oriented, thus the RMA algorithm was applied for normalization, having as result values on $\log_2$ scale.

Phenotype: The phenotypic data was taken using Biometric Research Branch (BRB) Array Tools, on an Excel datasheet format. BRB is an integrated package for the visualization and statistical analysis of DNA microarray gene expression data [110]. Some problems arise when trying to make the association between CEL files and samples on the phenotype spreadsheet, as the IDs of the samples in the excel file were not 100% identical to the names of the files and not all samples had suitable phenotypic data. To stabilize the samples names basing on CEL files names, several changes were made, such as the removal of the reference to the array from which they originate. Due to the restrictions of attribute names in the database, a non-numeric prefix was also added to each sample name and unrecognized characters were replaced (Table 8).

Table 8. Phenotype data transformation for Rembrandt.

Original		New	
disease_type	**who_grade**	**tumor_type**	**grade**
148 astrocytoma	65 II	astrocytoma	2
	58 III		3
	25 unknown		unknown
227 gbm	2 I	glioblastoma	1
	134 IV		4
	91 unknown		unknown
11 mixed	4 II	oligodendroglioma_astrocitoma	2
	3 III		3
	4 unknown		unknown
28 non_tumor	-	normal	0
57 oligodendroglioma	30 II	oligodendroglioma	2
	23 III		3
	14 unknown		unknown

67 unknown	1 III	unknown	3
	66 unknown	unknown	unknown

Even being a repository with a large number of samples, a considerable portion of them have incomplete or useless phenotypic information, so they were discarded.

Later, it was discovered that GDS1962 samples are also present on Rembrandt. However, the latter has more samples and more phenotypic information, such as survival length and gender that is not on the former, so this data dataset can be useful for further studies.

4.2.3 TCGA

TCGA data was collected on Data Portal, by Data Matrix. The search criterion was to retrieve all batches, at level 2, related to gliomas (Low Grade Gliomas - LGG and GBM). TCGA considered as LGG grades II and III, being GBM grade IV. Both gene expression and clinical data were retrieved.

LGG

Platform: AgilentG4502A_07_3

Sample count: 27

Expression values processing: None. Original values are on a $\log_2$ ratio scale.

Phenotype: Data was obtained from clinical files, at level 3. The transformations were minor, only on grade attribute (G2 → 2; G3 → 3) (Table 9).

Table 9. Phenotype data transformation for lower grade gliomas on TCGA.

New		
grade	**tumor_type**	**gender**
7 "2"	1 astrocytoma	m
	2 mixed	f
		m
	4 oligodendroglioma	2 f
		2 m
20 "3"	9 astrocytoma	1 f
		8 m
	7 mixed	3 f
		4 m
	4 oligodendroglioma	2 f
		2 m

The data also includes survival, not shown on table.

GBM

Platform: Affymetrix HG-U133A

Sample count: 557

Expression values processing: None. Original values are on a $\log_2$ scale.

Phenotype: Data was obtained from the aliquote barcode (Table 10).

Table 10. Glioblastoma phenotypic data transformation on TCGA for Affymetrix HG-U133A dataset.

Original	New		
sample on aliquot	tumor_type	grade	gender
10 "11"	normal	0	unknown
547 "01"	glioblastoma	4	3 unknown
			215 f
			329 m

The data also includes survival, not shown on table.

Platform: Agilent G4502A_07_1

Sample count: 122

Expression values processing: None. Original values are on a $\log_2$ ratio scale.

Phenotype: Data was obtained from aliquote barcode (Table 11).

Table 11. Glioblastoma phenotypic data transformation on TCGA for Agilent G4502A_07_1 dataset.

Original	New		
sample on aliquot	tumor_type	grade	gender
122 "01"	glioblastoma	4	53 f
			69 m

The data also includes survival, not shown on table.

Platform: Agilent G4502A_07_2

Sample count: 482

Expression values processing: None. Original values are on a $\log_2$ ratio scale.

Phenotype: Data was obtained from aliquote barcode (Table 12).

Table 12. Glioblastoma phenotypic data transformation on TCGA for Agilent G4502A_07_2 dataset.

Original	New		
sample on aliquot	**tumor_type**	**grade**	**gender**
10 "11"	normal	0	unknown
472 "01"	glioblastoma	4	5 unknown
			179 f
			288 m

4.2.4 Probes sequences

The Affymetrix and Agilent probeset identifiers were collected from the original datasets. Assuming a specific identifier of a manufacturer measures the same sequence, even if the platform is different, a global set of unique probeset identifiers were collected, for each manufacturer.

The Affymetrix sequences were taken from the Web site through the option *Products → Microarray Solutions → 3' IVT Expression Analysis → ... select desired microarray ... → Technical Documentation → Sequence Files (Tabular)*. It was checked that all probesets of U133A and U133B are also in U133 Plus 2.0 and that some U95Av2 probesets are also in U133 Plus 2.0. Thus, the starting set contained 54613 probesets from U133 Plus 2.0, to which were added 12286 U95Av2 specific probesets, making a total of 66899 unique probeset identifiers. On U133 Plus 2.0 array, 9711 probes were identified with duplicate sequences and in the U95Av2 array 1722 duplicate sequences were found. After joining the two sets, repeated sequences were noted in 22351 probes.

The Affymetrix platforms were:

- Human Genome U133 Plus 2.0 Array (54613 probesets)
- Human Genome U133A (22215 probesets)
- Human Genome U133B (22577 probesets)
- Human Genome U95Av2 (12558 probesets)

The Agilent probe sequences were collected with the eArray tool (https://earray.chem.agilent.com/earray/), using the option *Microarray → Browse*

Microarray Designs, selecting "*H. sapiens*". Agilent probe sequences were collected from the following platforms:

- *Human 1A Microarray*
    - Design ID: 011521;
    - Design Format: 1 X 22K;
    - Created: Jan 2005.
- *Human 1B Microarray*
    - Design ID: 011871;
    - Design Format: 1 X 22K;
    - Created : Jan 2005.
- *Human 1A Microarray (V2)*
    - Design ID: 012097;
    - Design Format: 1 X 22K;
    - Created: Jan 2005.
- *Whole Human Genome Microarray*
    - Design ID: 026652;
    - Design Format: 4x44K v2;
    - Created: Dec 2009.
- *Whole Human Genome Oligo Microarray*
    - Design ID: 012391;
    - Design Format:1x44K;
    - Created: Aug 2007.

All platforms reference the human genome build version hg19:GRCh37 (Feb2009).

4.3 Reannotation

As was mentioned earlier in section 2.2.3, each probeset measures only part of a gene, established by the respective platform manufacturer. If alternative splicing is taken into account, rarely two probesets read the same values, unless their sequences are located on the same exon. To show how this fact may affect gene expression, an experiment was made considering two datasets associated with GBM studies, GDS1962 and

AgilentG4502A_07_1. An analysis of variance was conducted between probeset expression values from the same manufacturer, for the same gene. The gene ZFYVE20 (zinc finger, FYVE domain containing 20) was one of the genes that presented the greatest variance between values of probesets from Affymetrix, so it was chosen for further analysis. The gene ZFYVE20 (ENSG00000131381 in Ensembl), is located on chromosome 3, in positions 15,111,580-15,140,670. There are 10 transcripts associated with this gene (Figure 6, Table 13).

Table 13. ZFYVE20 transcripts identified in Ensembl.

Transcript ID	Protein ID	Transcript Biotype
ENST00000253699	ENSP00000253699	protein_coding
ENST00000476527	ENSP00000422551	protein_coding
ENST00000426541	ENSP00000403368	nonsense_mediated_decay
ENST00000483098	—	processed_transcript
ENST00000449964	—	processed_transcript
ENST00000463214	—	retained_intron
ENST00000441057	ENSP00000414020	nonsense_mediated_decay
ENST00000449050	ENSP00000387724	protein_coding
ENST00000507357	ENSP00000427038	protein_coding
ENST00000435849	ENSP00000391039	protein_coding

All transcripts associated to this gene are on the reverse strand (-). In the GDS1962 file, we can find the following probesets for this gene: 1553569_at (+ strand); 1553570_x_at (+ strand); 1569714_at (- strand); 226581_at (- strand); 231765_at (- strand). If the transcripts identified on Ensembl are all on the reverse strand, it could be a mistake to join expression of probesets on the sense strand (present in GDS1962) to compute a gene expression value. This reinforces the idea of the usefulness of a reannotation system for the identifiers of expression values. There is the need to identify which parts of the genome each probeset is exactly measuring, to try to achieve a biological evidence of the relation between expression measurements of probesets from a gene.

Aligning ZFYVE20 probe sequences with the genome

A test was made to check which probesets have probes reading ZFYVE20.

Affymetrix

For Affymetrix, three distinct groups of results were obtained. In the first group, 35 probesets match this gene with just one of its probes, 3 probesets match only with two probes, while a single probeset made a double match of a single probe. Since a probeset has at least 11 probes, these matches can be considered as undesirable hybridizations and, therefore, discarded as gene representative probesets. The probesets that presented a broader match are included in the second group, where the 11 probes of these probesets match with the gene, without extra matches:

- ***1569714_at***

  exons matched: ENSE00001711894, ENSE00003535890, ENSE00003465398, ENSE00002065731.

- ***226581_at***

  exon matched: ENSE00001242854.

- ***231765_at***

  exons matched: ENSE00001242854, ENSE00001882794.

In the third group, a probeset matched with diffuse results:

- ***1008_f_at***

  From the 16 probes of this probeset, only 6 match with the gene. The probes of this probeset also matched 996812 times in regions out of this gene, strongly indicating that this probeset should be discarded.

Considering probesets associated to ZFYVE20 on GDS1962, only probesets 1569714_at, 226581_at and 231765_at were identified, by sequence alignment, as specific for this gene. No matches were registered for 1553569_at and 1553570_x_at. The probeset 1008_f_at is associated to an old platform, the U95Av2, and has not been kept on the HG-U133 Plus 2.0.

Agilent

The TCGA Agilent files do not contain the mapping probeset-gene, which have to be done with an alignment tool:

- ***A_32_P185718***
    - Strand +;
    - Exon matched: ENSE00001891770.
- ***A_24_P541213***
    - Strand +;
    - Exon matched: ENSE00001242854.
- ***A_23_P315872***
    - Strand -;
    - Exons matched: ENSE00001242854, ENSE00001882794.
- ***A_24_P85181***
    - Strand -;
    - Exons matched: ENSE00001242854.

None of these probesets have matches outside the ZFYVE20 gene.
It should be noted that these probesets were all matched with this gene using an alignment tool, unlike the GDS1962 probesets. However, the exons identified in Ensembl for the gene ZFYVE20 are all on the reverse strand, which discards the probesets A_32_P185718 and A_24_P541213 who read the + strand. With the targets of each probeset ranging all gene, it is difficult to make a direct relation between probesets.

Thus, a reannotation process will be conducted where probesets are mapped to transcripts, and another where probesets are mapped to genes, to verify if there are significant differences on posterior data integration experiments.

4.3.1 Sequence alignment for all probes

A match of each probe sequence with the human genome (version BSgenome.Hsapiens.UCSC.hg19) was done. Each probe could have zero or more matches along the genome. A maximum of one base pair (bp) mismatch was allowed, to

not discard probes that fail the hybridization in just a base or considering the occurrence of a SNP in the genome [32].

Affymetrix

From the 799238 Affymetrix probes, less than 5% do not have any matches in the genome. Most probes present only one match, while a maximum of 366059 matches were registered for a single probe. The largest numbers of matches were recorded in 4 probes of the probeset 1008_f_at. There are also cases where the same probe matches in several distinct regions of the same gene. For example, in the probeset 1557055_s_at a probe was detected with 10 distinct matches in the gene RP11-206L10.11 (ENSG00000228794). The distribution of number of matches per probe, considering a maximum of ten matches, is given in Table 14.

Table 14. Number of genome matches per probe.

Matches	1	2	3	4	5
Probes	675561	44974	13528	6229	4097

Matches	6	7	8	9	10
Probes	2776	1761	1502	1106	835

For some probesets (0,6%), no match has been identified for any probe. This could be explained by the upgrade of the genome annotation. Other possibilities could be pointed, like the fact that approximately 200 Mbp of the human genome, mainly from the centromeres and the short arms of the acrocentric chromosomes, are missing from the human reference genome [111]. The Genome Reference Consortium (GRC) builds the genome reference leaving these sequences in structures outside chromosomes. A GRC genome assembly, for *H. Sapiens* species, is composed by [112]:

- 24 "relatively complete" sequences for chromosomes 1 to 22, X and Y;
- A complete mitochondrial sequence;
- Several "unlocalized sequences" (their exact location within the chromosome is not known);
- Several "unplaced sequences" (their chromosomal association is not known);
- Several "alternate loci" (contain alternate representations of specific regions).

Considering these 'extra' genome sections, 487 additional probes were identified. The rest of the probes were analyzed with BLAST tools from Ensembl and UCSC, but continued to return no matches.

Annotation through GeneAnnot [113] presented incomplete results (Table 15) and is gene centric, i.e., it is not possible to obtain the exact chromosome location. The information obtained from GeneAnnot can be found through Affymetrix annotation files. The fact that some probes do not match any region of the genome could be due to outdated Affymetrix probe design (Build 133, April 2001 for U133 Plus 2.0 Array) when compared with the genome assemblies that are frequently updated by GRC (February 2009 for GRCh37/hg19). Probes that showed no match were discarded.

Table 15. Some Probesets Annotation by GeneAnnot.

Probeset	Gene	Chromo.	Strand	Start	End
224344_at	COX6A1	12	+	120875893	120878545
1565535_x_at	M74301	?	?	?	?
1554232_a_at	BC018433	?	?	?	?
208303_s_at	CRLF2	X	-	1314869	1331616

95.2% of the probes with matches registered at least one match in the genes referenced by Ensembl, getting 5% of the probes without apparent connection to any gene. 79.6% of the probes with matches in genes had at least one match in an exon. The remaining cover zones of introns, UTRs, or boundary regions that do not belong in their totality to exons. The annotation process was made by stages, crossing the probe matches obtained from Biostrings with the genome information contained on the Ensembl database.

4.3.2 Genome annotation

For the annotation of genes, transcripts and exons data were used from Ensembl DB.

Exons annotation

From Ensembl, for each exon, information was collected about the corresponding gene, sequence, location (start and end) on the respective chromosome, and strand. A filter was applied to return only the exons present in the 24 chromosomes from 1 to Y. The

information extraction for exons was made using *biomaRt* R package, because it allows to associate the sequence of each exon to the rest of the information.

Transcripts annotation

The transcript data obtained with the *biomaRt* package included non-normalized fields that would require a considerable processing time for their normalization. Thus, the Ensembl data for transcripts and genes was collected using the *BioMart* web tool in CSV format from http://www.ensembl.org/biomart/martview, with the options: *Database: Ensembl Genes 72* → *Dataset: Homo sapiens genes (GRCh37.p11)* → *Attributes*. It was possible to identify, for each transcript, the gene, the strand, start and end positions on the chromosome. In parallel, the transcript-exons map was built.

Genes annotation

The information for the genes was obtained in the same way as for the transcripts, identifying: the designation of each gene, chromosome, the beginning and end of the gene on the chromosome, the band and the amount of gene transcripts. For both genes and the transcripts, information was limited to 24 chromosomes from 1 to Y.

4.3.3 Mapping

To establish which probes match each exon, a cross reference was made between probe matches on the genome, exons annotation and genes annotation. The relation is made by the starting and ending point on the respective chromosome. If the match of a probe lies inside an exon area on the chromosome, then they were mapped. In this way, a probe was automatically associated with the corresponding gene too.

The same strategy was applied to associate exons to transcripts which, in turn, make a bridge between probes and transcripts.

A probeset is associated to a transcript if at least one of its probes reads the transcript. Their relevance on the transcript expression value is determined by their ratio. The expression values were recalculated using Eq. 15, obtaining transcript-oriented datasets, that are common to all, independently of the original source or platform manufacturer.

For the gene-oriented datasets, a simple mean, for each gene, of the probesets associated with it was made.

4.3.4 Reannotation analysis

The transcript-oriented integration was compared with probeset-oriented and gene-oriented integration. For this, a differential expression (DE) study was conducted, using GBM data from Affymetrix and Agilent. The data from Affymetrix was from the GDS1962 dataset, and Agilent data was a combination of datasets G4502A_07_2 and G4502A_07_3, from TCGA. The Affymetrix probeset-gene mapping was obtained directly from the GDS1962 dataset, and the mapping for Agilent comes from GEO GPL1708 platform. The applied model was based on empirical Bayes, with adjusted p-value for multiple comparisons [114]. The model was applied to all possible contrasts of the World Health Organization (WHO) defined grades found in datasets (4-3, 4-2, 4-0, 3-2, 3-0, 2-0), where 0 represents non-tumor samples. To evaluate significance, a threshold of 0.01 was used for p-values.

Probeset-oriented experiment

For the Affymetrix dataset, 27911 significant probesets were identified, with 17914 unique genes. For Agilent datasets, there were 27910 significant probesets, with 14322 unique genes. Crossing the results, a total of 9154 significant genes were found in common between the two vendors. If all probesets of each gene were considered in batch, the result would be different. For example, the Affymetrix probeset 205975_s_at associated with gene HOXD1 has a significant p-value (≈ 0.01), but the gene would have been excluded from the list of significant genes if both probesets associated with the gene were considered (Table 16).

Table 16. Differential expression for gene HOXD1 probesets.

Probeset	Average Expression	P-value
205975_s_at	8.751664	0.00999832
205974_at	5.362827	0.93772757

Gene-oriented experiment

This assay was made using, for each gene, the probeset expression mean. For the Affymetrix dataset, 15975 significant genes were identified. For Agilent datasets, there were 13131. Crossing the results, a total of 7418 significant genes were found in common.

Transcript-oriented experiment

For the Affymetrix dataset, it was possible to identify 40843 significant transcripts, corresponding to 13015 unique genes. For Agilent datasets, there were 51149 significant transcripts, corresponding to 16329 unique genes. Crossing the results, a total of 9950 significant genes were found in common between the two platforms. Considering the transcripts, 27117 significant transcripts were identified in common between the platforms.

Comparing annotations

The DE analysis will be shown for the ZFYVE20 gene with 10 different transcripts (Table 17, Table 18, Table 19). The different results on the probeset-oriented studies reflect the variance of expression patterns between probesets. The gene-oriented option reflects an intermediate result that could be far away from the individual results of the probesets, by not differentiating measures of expressed exons from measures of not expressed exons. On the transcript-oriented experiment, it is possible to compare results between the two vendors. A good biomarker for Agilent, ENST00000476527, has presented worse results with Affymetrix data.

Table 17. Transcript-oriented differential expression results for ZFYVE20.

	Ensembl Transcript id	Average Expression	P-value
Affy	ENST00000253699	9,526849	3,52E-10
	ENST00000476527	8,351509	0,067566
	ENST00000449050	6,908623	0,566146
Agil	ENST00000253699	0,39565	1,97E-30
	ENST00000476527	-0,17256	1,33E-23

Table 18. Gene-oriented differential expression results for ZFYVE20.

	Average Expression	P-value
Affymetrix	11.14515	0.01047128
Agilent	0.6700075	1.642216e-12

Table 19. Probeset-oriented differential expression results for ZFYVE20.

	Probeset	Average Expression	P-value
Affymetrix	226581_at	10,70219	1,79E-16
	231765_at	8,351509	0,069017
	1553570_x_at	14,91368	0,079318
	1553569_at	14,84977	0,121551
	1569714_at	6,908623	0,568003
Agil	A_23_P315872	-0,09812	8,59E-16
	A_24_P85181	1,438134	5,66E-06

Another example will be shown, based on [102] which claim that the SCF/KITLG gene expression has a direct relation with the brain tumor grade. Making an analysis with the same dataset, a result was determined, considering all possible contrasts (Table 20). Applying the same study on the two Agilent datasets mentioned earlier, the results are different (Table 21).

Table 20. Differential expression, by probeset, considering all contrasts in Affymetrix GDS1962, for gene KITLG.

Probeset	Average Expression	P-value
211124_s_at	5,357034	0,00357
207029_at	6,77032	0,00455
226534_at	10,33247	0,55152

Table 21. Differential expression, by probeset, considering all contrasts in Agilent G4502A_07_2 and _3, for gene KITLG.

Probeset	Average Expression	P-value
A_23_P204654	-0,08614897	0,001746
A_24_P64781	0,45391565	0,029701
A_24_P133253	1,52740779	0,156449

The significance obtained does not reach the p-value < 0,001 referred in [102]. Moreover, it is not possible to establish a bridge between the two vendors. The same tests were made with a transcript-oriented basis (Table 22 and Table 23).

Table 22. Differential expression, by transcript, considering all contrasts in Affymetrix GDS1962, for gene KITLG.

Ensembl Transcript id	Average Expression	P-value
ENST00000228280	6,063677	0,001713
ENST00000347404	6,063677	0,001713
ENST00000378535	7,486608	0,00634

Table 23. Differential expression, by transcript, considering all contrasts in Agilent G4502A_07_2 and _3, for gene KITLG.

Ensembl Transcript id	Average Expression	P-value
ENST00000228280	-0,09021	0,001187
ENST00000347404	-0,09021	0,001187
ENST00000378535	0,314779	0,005902

Now, it is possible to see that ENST00000228280 and ENST00000347404 transcripts have identical behavior in both datasets, and a poorer significance is achieved for ENST00000378535 in both datasets. These results may be related to the fact that ENST00000378535 is a non-coding transcript.

Further tests were made, for all possible contrast combinations (results not shown). On these tests, it is noted that the same behavior is extensible to the other contrasts. A correlation test for log fold change, between Affymetrix and Agilent transcript values, when considering the HighGrade-LowGrade contrast, shows a strong 0.83 Pearson's correlation coefficient.

4.4 Classification and feature selection

The problem to solve is a classification task, with 4 distinct classes, where grade 1 was excluded due to the lack of representative samples on datasets and grade 0 refers to non-tumor. To exemplify, the distribution of the expression values across samples of dataset

GDS1962 (Affymetrix) is shown in Figure 11, specifically for the probeset 201292_at (gene TOP2A). This probeset was highlighted from a previous eBayes analysis, presenting a p-value of 8.02E-34 in this analysis.

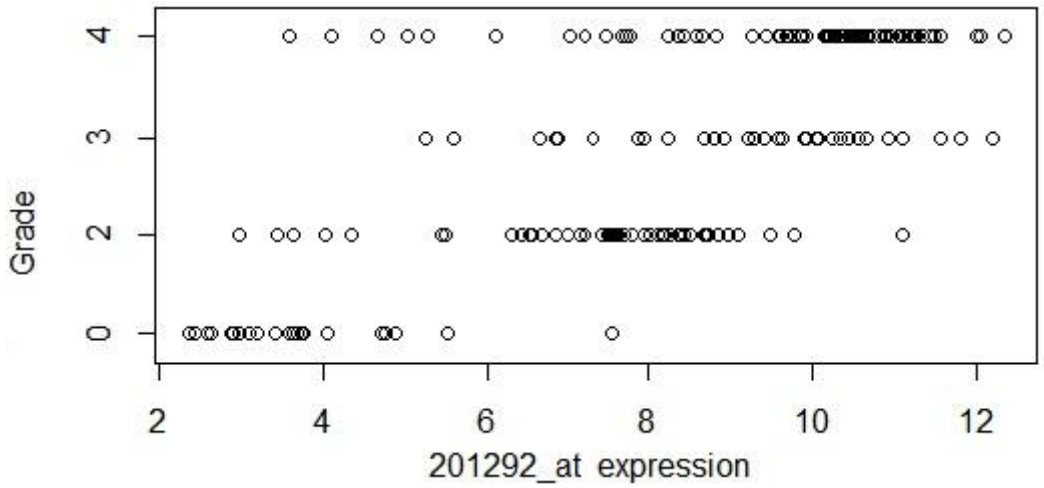


Figure 11. Expression/grade distribution for Affymetrix probeset 201292_at.

Four candidate algorithms were used, suitable for multi-level classification tasks: Ordinal Logistic Regression (OLR); Support Vector Machines (SVM); k-Nearest Neighbors (k-NN); and Linear Discriminant Analysis (LDA).

For the tests, datasets related with glioblastoma were selected, where it was possible to distinguish tumor grade. Three datasets came from GEO, built with Affymetrix HG-U133Plus2.0, HG-U133A and HG-U133B platforms: GDS1962; GDS1815_6; and GDS1975_6. Two datasets were taken from TCGA, built with Agilent platforms: TCGA_AGIL2; and TCGA_AGIL4. As GDS1962 is the most complete and balanced dataset, it was used for preliminary Affymetrix probesets tests. For Agilent, TCGA_AGIL2 g4 subset was trimmed, leaving only 40 g4 samples. The two Agilent datasets were joined on TCGA_AGIL2_4 (10 g0, 7 g2, 20 g3, 40 g4). Higher-level datasets were built from probeset data. The gene level datasets were produced averaging all probesets associated with each gene. The transcript datasets were produced by weighted average of the probesets associated with each transcript [115].

Pearson's correlation and eBayes filters were applied on these datasets and the probesets were sorted by rank. Over this, a cleaning process was performed to remove collinear

probesets, using a 0.9 Pearson's correlation cutoff. Finally, only a set of the 200 best ranked probesets were kept for further tests.

The wrapper method was tuned to make combinations until a maximum of 20 features. Also, the quantity of the best combinations preserved between cycle iteration was limited to 20. For k-NN, k was always set to 3. The main results are provided in Table 24, Table 25 and Table 26.

Table 24. Results for classifiers (columns), filtering methods (rows) at the probeset level. #P - number of probesets; ER - Error Rate.

		OLR	SVM	k-NN	LDA
Pearson Correlation	Affy	#P: 16 ER: 0.25	#P: 13 ER: 0.11	#P: 18 ER: 0.14	#P: 13 ER: 0.19
	Agil	#P: 02 ER: 0.13	#P: 6 ER: 0.04	#P: 06 ER: 0.01	#P: 05 ER: 0.04
eBayes	Affy	#P: 16 ER: 0.28	#P: 10 ER: 0.09	#P: 20 ER: 0.14	#P: 12 ER: 0.17
	Agil	#P: 05 ER: 0.17	#P: 09 ER: 0.03	#P: 06 ER: 0	#P: 04 ER: 0.03

Table 25. Results for classifiers (columns), filtering methods (rows) at the gene level. #G - number of genes; ER - Error Rate.

		OLR	SVM	k-NN	LDA
Pearson Correlation	Affy	#G: 16 ER: 0.26	#G: 19 ER: 0.08	#G: 14 ER: 0.14	#G: 19 ER: 0.17
	Agil	#G: 03 ER: 0.12	#G: 08 ER: 0.03	#G: 05 ER: 0.01	#G: 11 ER: 0
eBayes	Affy	#G: 16 ER: 0.23	#G: 17 ER: 0.09	#G: 18 ER: 0.14	#G: 19 ER: 0.17
	Agil	#G: 05 ER: 0.17	#G: 04 ER: 0.01	#G: 06 ER: 0	#G: 13 ER: 0

Table 26. Results for classifiers (columns), filtering methods (rows) at the transcript level. #T - number of transcripts; ER - Error Rate.

		OLR	SVM	k-NN	LDA
Pearson Corr.	Affy	#T: 18 ER: 0.25	#T: 14 ER: 0.1	#T: 20 ER: 0.14	#T: 17 ER: 0.17
	Agil	#T: 03 ER: 0.09	#T: 06 ER: 0.03	#T: 07 ER: 0.01	#T: 17 ER: 0.04
eBayes	Affy	#T: 18 ER: 0.23	#T: 18 ER: 0.07	#T: 17 ER: 0.13	#T: 13 ER: 0.16
	Agil	not converge	#T: 06 ER: 0.02	#T: 04 ER: 0.02	#T: 06 ER: 0

The differences between results associated with the filtering method are not significant, so one will be used arbitrarily. It is also possible to see that the accuracy is always better for Agilent datasets, and SVMs have the best performance when considered both platforms. In the matters of probeset, gene or transcript level, the former presented

slightly worse results. Between gene and transcript, it is not possible to distinguish a clear difference. Besides the identification of the best classification algorithm, it is important to compare the prediction features obtained for Affymetrix and Agilent.

If a transcript/gene is related to the grade phenotype, then it should be relevant for both platforms. To check this premise, eBayes was applied to rank features separately, and then pairwise Kendall's coefficient of concordance (W) was calculated. For genes we obtained W = 0.675, and for transcripts W = 0.711.

There is a considerable discordance about feature relevance between Affymetrix and Agilent. Deepening the analysis, the features selected as the best predictor with SVM for Affymetrix were used for Agilent tests and vice versa. With Affymetrix predictors, a LOOCV error of 0.10 for genes and 0.09 for transcripts, was obtained when these predictors were applied on Agilent. For Agilent predictors, an error of 0.29 for genes and 0.32 for transcripts was obtained, when applied on Affymetrix data. It is clear that predictors obtained from Agilent data tend to overfit more and loose generalization.

The next tests were made joining all datasets on a single one, using different methods for data integration (Table 27). SVMs are confirmed as the training algorithm that gives the best results and the division of the data by a reference set of values (grade) also improves accuracy. As the common grade between all datasets is g3, this was the reference grade used.

Joining heterogeneous datasets aims to produce a robust training set that allows to classify a new sample with a good rate of assertiveness. With these LOOCV tests, it is possible to preliminarily confirm that it is possible to do this with datasets from Agilent and Affymetrix.

Table 27. LOOCV error rate for Affymetrix and Agilent data integration.

		Raw	Norm	Linear	LinearLog2	Grade	GradeLog2
OLR	**Genes**	0.24	0.28	0.27	0.31	0.2	0.2
	Transcripts	0.24	0.3	0.3	0.26	0.19	0.23
SVM	**Genes**	0.11	0.09	0.11	0.09	0.08	0.12
	Transcripts	0.09	0.1	0.12	0.15	0.08	0.11
k-NN	**Genes**	0.13	0.16	0.19	0.15	0.14	0.17
	Transcripts	0.11	0.18	0.19	0.2	0.14	0.17
LDA	**Genes**	0.16	0.2	0.19	0.16	0.14	0.18
	Transcripts	0.16	0.19	0.22	0.19	0.15	0.18

To take the tests a little further, the same joined dataset was used to build a model to predict a new Affymetrix dataset, collected from TCGA, not used before in any stage. This new dataset has 10 g0 and 287 g4 samples so, to apply the reference grade method, all data was divided by grade 0. The results again confirmed the good performance of the method (Table 28).

Table 28. Confusion matrices for prediction of the new dataset.

Genes

		Predicted			
	grade	0	2	3	4
Observed	0	10	0	0	0
	4	0	6	5	266
error rate:		0.038			

Transcripts

		Predicted			
	grade	0	2	3	4
Observed	0	10	0	0	0
	4	0	4	8	265
error rate:		0.042			

5 Software development

This chapter provides a description of the software architecture, at the database level and at the 'business' level and their interactions. The tools and programming languages used in this work will be described, while some software packages utilized will also be mentioned.

A reference to the principal functions created at each stage of the software development will be made. It is not intended to deepen programming paradigms, but rather to show a solution adapted to the concrete challenge that was proposed.

The functions are collected in a reference manual that could be consulted in Appendix 2.

5.1 Requirements

In a software programming project, the definition of the requirements is typically the first step. These were the main requirements defined, to build the gene expression integration system that could address the research questions in the context of this thesis:

- Collect gene expression datasets from several different sources and in different formats;
- Integrate the previous gene expression datasets, possibly coming from different formats and manufacturers;
- Normalize expression datasets into a common structure;
- Collect phenotypic information that can be used on machine learning tasks;
- Normalize phenotypic data;
- Make a feature map between datasets with different feature identifiers;
- Apply machine learning algorithms over the integrated data;

- Develop open-source and free to use software;
- Implement a system able to run on a personal computer with limited resources.

5.2 Architecture

The architecture of the solution adopted, called "Integrated Gene Expression Information System" (IGEIS), is shown on Figure 12.

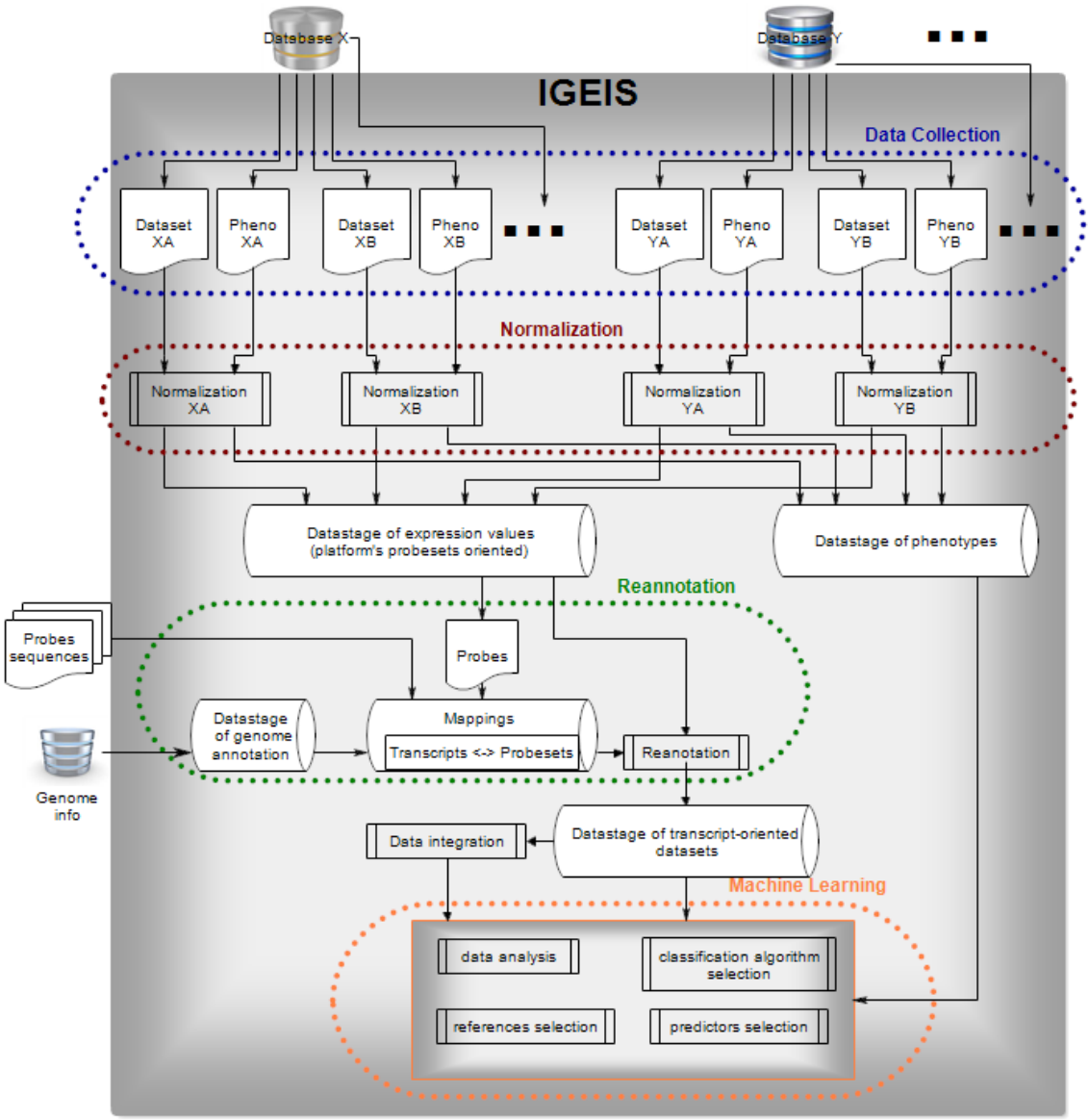


Figure 12. IGEIS general architecture

In Figure 12 it is possible to see the main architectural blocks that compose IGEIS. The "Data Collection", is the section of downloaded files from global databases. This also may include the "Normalization" section, where the data on files is treated before is inserted in the "Datastage of expression values" and "Datastage of phenotypes". On "Reannotation section", with external genome and probe sequences information, the datasets are transformed from a probeset-oriented annotation for a transcript-oriented annotation. The transformed datasets are stored in the "Datastage of transcript-oriented datasets". The datasets are then integrated using a set of integration methods. Over the integrated datasets several "Machine Learning" algorithms and tools are applied, to assess the quality of the model and to retrieve phenotype-gene expression relation patterns.

5.3 Tools and resources

This section provides a summary of the main tools and resources used in the software development process.

Hardware and operation system requirements

The IGEIS was developed over a windows 64bits operation system. Compatibility with other operation systems are not guaranteed.
Some functions, specially on the reannotation section, are very resources consumer, thus is recommended to run with a CPU multi-core and with a minimum of 8 GB RAM.

Wampserver

Wampserver is a Windows web development environment [116]. It includes the tools necessary to give support to database development, as the MySQL Database Management System. It also includes the interface phpMyAdmin, used in this work essentially to check if the SQL queries were producing the desired result.

Wampserver version: 2.1

- Apache version: 2.2.17
- PHP version: 5.3.4

- MySQL version: 5.1.53

MySQL Workbench

MySQL Workbench is a unified visual tool for data modeling and SQL development, among other features [117]. It was used to construct the Enhanced Entity-Relationship model.

MySQL Workbench version: 5.2.33 CE Revision 7508

BioMart

The BioMart project provides free software and data services to the scientific community to foster scientific collaboration and facilitate the scientific discovery process [118]. BioMart was used to extract genome data from the Ensembl database, using the online tool version and the biomaRt R package.

Biomart version: 0.7

DataPortal

The TCGA Data Portal provides a platform for researchers to search, download, and analyze data sets generated by TCGA [119]. It contains clinical information, genomic characterization data, and high level sequence analysis of the tumor genomes. It was used to download data from the TCGA database.

BRB Array tools

Biometric Research Branch (BRB) tools is an integrated package for the visualization and statistical analysis of DNA microarray gene expression data [120]. It was used to download the gene expression data from the Rembrandt database.

BLAT

BLAT (BLAST-Like Alignment Tool) is a very fast sequence alignment tool similar to BLAST, which author refers to being more accurate and 500 times faster than popular existing tools for mRNA/DNA alignments [121]. The claim that BLAT It was used to make a study of genes structure and probesets sequences alignment (Figure 6).

5.3.1 R system and packages

R is a free language and environment for statistical computing and graphics, similar to the S language. R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical functions organized into packages. R is open source and highly extensible, being open to community contributions through development of new packages [122].
The code in R environment can be run in blocks or line by line, allowing a great control of code execution.
In this language, we find all the basic components of the common languages and some specific features like the data frame structure. This type of object is a table but, contrarily to a matrix, allows columns with different types.
Other extremely important feature on R is the vectorization, which applies on operations with arrays as arguments. Vectorization is a built in feature that is more like a programming paradigm that boost the speed of operations execution, specially for multidimensional arrays. The better way to explain this concept is by example. I we have two matrices A and B and it is intended to sum then into a new matrix C, we could use the traditional programming code:

```
for(a in 1:NR){
        for(b in 1:NC){
                C[a][b] ← A[a][b] + B[a][b]
        }
}
```

The processing time of this traditional loops is considerable and escalates with the number of rows (NR) and columns (NC) of the matrices. In R this may be replaced by vectorization:

```
C ← A + B
```

This operation does the same that the loops before and in much less time.

This is a example for a simple mathematical function but also applies in more complex functions built over a vectorization base. For behind vectorization is code written on lower level languages, like C.

R was the working language and environment used for the main development of the methods proposed in this work, given its great base of statistic capabilities, also very useful for machine learning tasks. The open contribution system also allows R to contain a huge number of packages to support bioinformatics functions.

R version: 2.15.1 (64-bits)

Bioconductor

Bioconductor is an open source project that provides tools for the analysis and comprehension of high-throughput genomic data [123]. Installing Bioconductor on R gives access to hundreds packages for the analysis of DNA microarray, sequences, SNPs, and other data.

R Packages

The following packages were used on the functions of IGEIS.

- *class* - version: 7.3-5, date: 2012-10-03, title: Functions for Classification;
- *MASS* - version: 7.3-23, date: 2013-01-08; title: Support Functions and Datasets for Venables and Ripley's MASS;
- *e1071* - version: 1.6-3, date: 2014-02-13, title: Misc Functions of the Department of Statistics (e1071), TU Wien;
- *Biobase* - version: 2.18.0, date: 2012-10-02, title: Biobase: Base functions for Bioconductor;
- *limma* - version: 3.14.4, date: 2013/01/16, title: Linear Models for Microarray Data;
- *reshape* - version: 0.8.4, date: 2011-01-28, title: Flexibly reshape data;
- *reshape2* - version: 1.2.2, date: 2012-12-04, title: Flexibly reshape data: a reboot of the reshape package;
- *data.table* - version: 1.9.2, date: 2014-02-27, title: Extension of data.frame;
- *GEOquery* - version: 2.24.1, date: 2013-01-25, title: Get data from NCBI Gene Expression Omnibus (GEO);

- *RMySQL* - version: 0.7-5, date: 2010-07-13, title: R interface to the MySQL database;
- *XLConnect* - version: 0.2-7, date: 2014-01-03, title: Excel Connector for R;
- *Biostrings* - version: 2.26.3, date: 2013-01-27, title: String objects representing biological sequences, and matching algorithms;
- *BSgenome.Hsapiens.UCSC.hg19* - Version: 1.3.19, date: 2012-11-16, title: Homo sapiens (Human) full genome (UCSC version hg19);
- *plyr* - version: 1.8, date: 2012-12-06, title: Tools for splitting, applying and combining data;
- *biomaRt* - version: 2.14.0, date: 2012-10-02, title: Interface to BioMart databases (e.g. Ensembl, COSMIC ,Wormbase and Gramene).

5.4 Data collection and representation

The process of data collection is intricate and needs a different approach for almost each dataset. It is intended to obtain, in the final of this stage, a database with treated expression datasets with similar fields and normalized phenotypes.

The structure chosen for the database tables on IGEIS used rows representing samples and columns representing probesets. This is a not recommended table structure in the point of view of relational database modeling, since it gives place to a undefined number of columns to put on the database. However, it is a requirement in this case, because the row size allowed for each table could not stand with all the features of a gene expression dataset.

A data melt could be done where in one column we have samples names and in another we have expression values, but some packages (like *limma*) are already expecting the other format, which means that we would have to be melting and unmolding several times the datasets which is unthinkable since it is extremely time consuming.

5.4.1 GEO simple datasets

Expression

GEO entities are managed in R through the *GEOquery* package. The datasets (GDS) are obtained with the function *getGDSDataset( )*, which returns a data frame with the format seen on Figure 13.

probeset_id	GSM48598	GSM48617	GSM48600	GSM48601
1000_at	2617.74	2261.21	1911.29	1461.22
1001_at	41.642	85.54	167.458	57.583
1002_f_at	54.142	32.64	<NA>	127.156
1003_s_at	235.692	128.239	<NA>	<NA>

Figure 13. Example of a section of an expression data frame.

The *normalizeExpressionDF( )* function is applied over this data frame, to remove control probesets. These probesets are identified by the "AFFX-" prefix, for Affymetrix, and their purpose is to be used for batch effect reduction, since the sequences that they read were identified as expression invariants between tissues. However, this not always occurs and, moreover, these probesets often present NULL values, making them useless for their purpose. The function *normalizeExpressionDF( )* calls *handleNAs( )* to eliminate NULL values, being established a minimum of 1 to the expression values, which prevents that $\log_2$ transformation, performed next, presents negative results. The handle NAs algorithm is to be applied to a matrix *M* of numeric values and tries to preserve the information relation between attributes:

for each column C_i
  order other columns C_j by correlation with C_i
  get linear models for $C_i \sim C_j$ (most correlated)
  for each line r of C_i
    if V_{ri} is a missing value
      if V_{rj} value is not missing
        $V_{ri} = a_{ij} * V_{rj} + b_{ij}$

$$V_{ri} = \begin{cases} V_{ri} & , V_{ri} > \text{minimum(M)} \\ \text{minimum(M)} & , V_{ri} <= \text{minimum(M)} \end{cases}$$

      else

```
                                get next correlated column C_{j+1}
                                repeat if
                        end if
                        if all V_{r j} are missing values → eliminate row
                end if
        end for each
end for each
```

Expression dataset is inserted on database "raw_expressions" with the *dbInsertExpDF( )* function. Here, the original dataset id is used with the suffix "_exp" to name the corresponding table on the database. The number of attributes on the database table is variable, corresponding to the first column for probesets names plus one column for each sample on the original dataset. Thus, the table is created by *dbInsertExpDF( )* function dynamically, setting 'probeset_id' as primary key, of type VARCHAR, and adding as much attributes as samples, each of FLOAT type.

Phenotype

The phenotype are obtained with the *getGDSPheno( )* function, that extracts it from the GDS file (Figure 14).

sample	other	disease.state	description
GSM48598	cell line	control	Value for GSM48598: KHG-GliomaCellCRL-1aAv2-s2; src: Human cell line
GSM48617	cell line	control	Value for GSM48617: KHG-GliomaCell87-1aAv2-s2; src: Human cell line
GSM48600	tumor	low grade astrocytoma	Value for GSM48600: KHG-LG-3Av2-s2; src: Human brain tumor
GSM48601	tumor	low grade astrocytoma	Value for GSM48601: KHG-LG-6Av2-s2; src: Human brain tumor

Figure 14. Example of a section of a phenotype data frame.

The *normPheno( )* function is used to translate original attributes to IGEIS standardized attributes. These transformations are different for each dataset, so the system was projected to accept an external spreadsheet, defined by the user that submits the dataset to IGEIS, with the rules for this dataset. This document is composed by two sheets, the first with attributes (Table 29) and the second with data (Table 30). The first is used to select from which original columns to extract information for the new columns, while the second is to define the transformations on the original column values to obtain the final values on the new column.

Table 29. Attributes table for GDS2853.

phenotype	new_phenotype	db_attribute_constraints
sample	sample	varchar(100)
other	source	varchar(100)
disease.state	tumor_type	varchar(100)
disease.state	who_grade	varchar(100)
description	description	text

Table 30. Data table for GDS2853.

attribute	origin	final
tumor_type	control	normal
tumor_type	low grade astrocytoma	astrocytoma
tumor_type	high grade astrocytoma	astrocytoma
tumor_type	low grade glioblastoma	astrocytoma
who_grade	control	0
who_grade	low grade astrocytoma	low_grade(1-2)
who_grade	high grade astrocytoma	high_grade(3-4)
who_grade	low grade glioblastoma	low_grade(1-2)
source	cell line	cell_line
source	Human brain tumor	tumor

The normalization algorithm used is the following:

```
oldAttrib <- get phenotype
newAttrib <- get new_phenotype
create new phenoData
for each attrib1 in newAttrib and attrib2 in oldAttrib
        add attrib1 column to phenoData
        move attrib2 column values to attrib1 column
end for

for each new attribute column
        pattern <- get origin
        replaceValue <- get final
        replace pattern with replaceValue on every elements of the column
end for
```

Note that the new attributes that do not have transformations explicit in the sheet maintain the original source format. This is the case of description, that generally is a dump of an already existing description.

The *insertPheno( )* function stores the phenotype in the database, on a table with name "<gdsID>_pheno". The data types for table construction are gathered from the rules spreadsheet.

5.4.2 GEO complementary datasets

The function *getDualGDSDataset( )* is used to fetch the datasets, with recurs to *getGDSDataset( )*, being retrieved a probeset concatenated dataset.

The rest of the process is similar to the previous section, with the application of *normalizeExpressionDF( )* and *dbInsertExpDF( )*.

The phenotype data is equal for the two datasets, so it is collected only from one of them, applying consecutively *getGDSPheno( )*, *normPheno( )* and *insertPheno( )*.

5.4.3 GEO series

The *getGSEDataset( )* function returns the expression data frame associated to a given series. As there are series with more than one dataset associated, it is possible to choose the dataset retrieved, specifying the position on 'esetNumb' argument. The *getGEO( )* function called inside *getGSEDataset( )* exports an *ExpressionSet* object, from which it is possible to extract an expression values table.

The expression data passes through *normalizeExpressionDF( )*, to verify if missing values exist (represented as NA in R), and then is inserted on the database (*dbInsertExpDF( )*).

The manipulation of the phenotype data is identical of the previous, with the difference that the data is gathered from a series structure: *getGSEPheno( )* → *normPheno( )* → *insertPheno( )*.

5.4.4 TCGA

Expression

Expression datasets from TCGA are obtained with the *getTCGADataset( )* function. It is only allowed to load a dataset from previous downloaded files and there is not a common nomenclature for the file name, path or structure, so this information has to be passed in the arguments. In Figure 15, the folders structure and nomenclature containing TCGA expression files is shown.

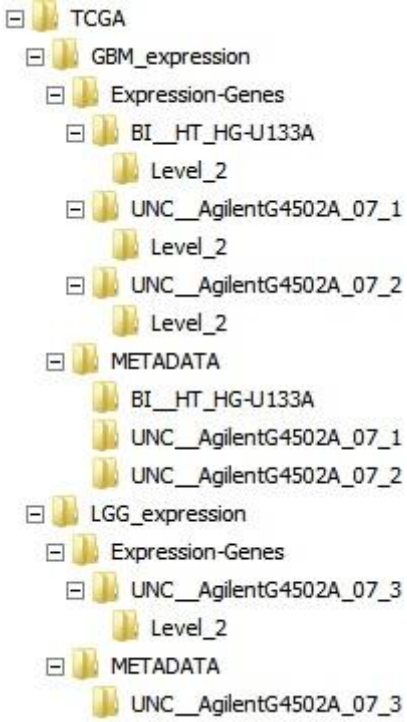


Figure 15. Structure of TCGA expression folders.

The data is downloaded on a disease base, so the folders are named firstly by the phenotype attribute used on the search. The "Expression-Genes" folder contains expression data, and is divided by the different microarray platforms used to acquire data. Inside these folders, there are subfolders for each level of data (probe, probeset or gene). There is also a "METADATA" folder, with meta-information about expression data.

"Level_2" folders have probeset-based datasets, but each sample is in a separate file, (e.g. "550001402607203260701l.A07.level2.data.txt"). The identifiers used for TCGA samples are aliquot barcodes, so a translation is done, using the corresponding Sample and Data Relationship Format (sdrf) file, located on the metadata section. When the

experiment has two channels, the sdrf file presents information for the two channels in an interleaved way. As the RNA in test is on the cy5 channel, the data is filtered to leave just the rows of interest.

Before applying barcodes as sample identifiers, it is needed to convert '-' character to '_', since the first one is not allowed in attributes names on a MySQL table.

Datasets with a large number of samples were split in two, with approximately the same number of samples.

The expression data frame obtained is normalized (*normalizeExpressionDF( )*) and inserted on the database (*dbInsertExpDF( )*).

Phenotype

The TCGA phenotype files are structured based on disease (as shown in Figure 16).

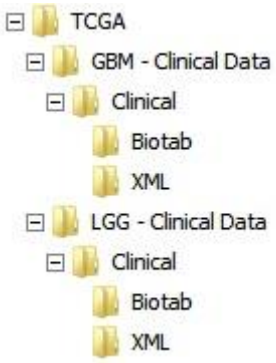


Figure 16. TCGA phenotype folders.

The data is disposed in clinical files presented on plain text (Biotab) or XML format. The Biotab folder has several files associated with the clinical history of patients, as disease progression and applied treatments, being the clinical-patient (e.g. "clinical_patient_gbm.txt") the most informative for this work.

An aliquot barcode has the complete identification of patient, disease condition, etc. An entry on the clinical file is identified only by the patient section of the barcode (Figure 17), so that part was extracted from the aliquot barcodes (e.g. TCGA_02_0071_01A_01R_0202_01 → TCGA_02_0071) .

```
  bcr_patient_barcode age_at_initial_pathologic_diagnosis anatomic_neoplasm_subdivision
1       TCGA-12-0780                                 51                [Not Available]
2       TCGA-12-0657                                 69                [Not Available]
                    bcr_patient_uuid date_of_form_completion days_to_birth days_to_death
1 120e257d-1cf1-421f-9a6b-6a5681eaea2e              2008-12-10        -18646           452
2 a9a0bad8-6074-4a7c-9f9e-7dbc1de01def              2008-12-11        -25413             3
  days_to_initial_pathologic_diagnosis days_to_last_followup eastern_cancer_oncology_group              ethnicity
1                                    0                   434               [Not Available] NOT HISPANIC OR LATINO
2                                    0                     3               [Not Available] NOT HISPANIC OR LATINO
  gender              histological_type history_of_neoadjuvant_treatment icd_10 icd_o_3_histology icd_o_3_site
1 FEMALE Untreated primary (de novo) GBM                              No  C71.9            9440/3        C71.9
2   MALE Untreated primary (de novo) GBM                              No  C71.9            9440/3        C71.9
  informed_consent_verified initial_pathologic_diagnosis_method karnofsky_performance_score patient_id
1                       YES                  Excisional Biopsy                          80        780
2                       YES                  Excisional Biopsy                [Not Available]     657
  performance_status_scale_timing person_neoplasm_cancer_status prior_glioma  race tissue_source_site
1            Pre-Adjuvant Therapy                    WITH TUMOR           NO WHITE                12
2                 [Not Available]                    WITH TUMOR           NO WHITE                12
  tumor_tissue_site vital_status year_of_initial_pathologic_diagnosis
1             Brain     DECEASED                                2006
2             Brain     DECEASED                                1996
```

Figure 17. TCGA Clinical file example, with two entries.

It is not clear why this format is adopted for a clinical file, because there may be multiple samples related to the same patient. A sample taken from a tumor and a sample taken from blood, from the same patient, have an identical patient section, but a different aliquot barcode. By examination of the clinical file, discrepancies were found, where some entries are identified as "non tumor", when the files were obtained with the "GBM" filter. To regulate these cases, a merge was made between the clinical entries and the aliquot barcode, repeating each patient entry for each aliquot corresponding to that patient. The aliquots were collected from the previously stored expression data.

On the normalization process (*normPheno( )*), the ambiguous phenotype information was defined based on the sample section of the aliquot barcode (Table 31) and the source study of the files.

Table 31. All possible values for sample field on a aliquot barcode and the corresponding phenotype meaning.

Sample	Phenotype
01	Primary solid Tumor
02	Recurrent Solid Tumor
03	Primary Blood Derived Cancer - Peripheral Blood
04	Recurrent Blood Derived Cancer - Bone Marrow
05	Additional - New Primary
06	Metastatic
07	Additional Metastatic
08	Human Tumor Original Cells
09	Primary Blood Derived Cancer - Bone Marrow
10	Blood Derived Normal
11	Solid Tissue Normal

12	Buccal Cell Normal
13	EBV Immortalized Normal
14	Bone Marrow Normal
20	Control Analyte
40	Recurrent Blood Derived Cancer - Peripheral Blood
50	Cell Lines
60	Primary Xenograft Tissue
61	Cell Line Derived Xenograft Tissue

Phenotype tables, after normalization with *normPheno( )*, were put on the database, using the *insertPheno( )* function.

5.4.5 Rembrandt

Expression

The expression collection for this database was made with *getRMBDataset( )* function, over pre-downloaded files. Rembrandt expression data is at probe level and refers only to U133 plus 2 Affymetrix platform. Thus, each sample is stored in a CEL file. The CEL files were read with the *ReadAffy( )* function from the *affy* package, in batches of 50, since R cannot stand with much heavier vectors. The RMA method was applied over each batch to obtain an *ExpressionSet* object and then all of them were combined on a single *ExpressionSet*.

Sample names were normalized, replacing '-' character for '_', trimming file extension and concatenating the "R" prefix to prevent attribute names to start with a number.

Expression data frames are normalized with the function *normalizeExpressionDF( )* and inserted in the database using the function *dbInsertExpDF( )*.

Phenotype

Rembrandt phenotype data comes on a spreadsheet, being used the function *getRMBPheno( )*, resorting to the *XLConnect* package to load it. Given the dimension of the sheet, and a poor garbage collection mechanism, *XLConnect* may show an error of lack of memory heap for java or a timeout on the reading process. To ease these problems the line "options(java.parameters = "-Xmx2048m")" is used to increase

allocated RAM and "options(java.parameters = "-XX:-UseGCOverheadLimit")" to turn off the timeout [124].

Duo to unconformities on the data, the process of acquisition by *getRMBPheno( )* has to be hard coded. It was used to rename samples E09624 and E10013 that were wrongly identified as 0308diff1_U133P2 and 1228dif1_U133P2, and for sample E10512, which does not have a phenotype on the spreadsheet. As this sample was collected from a GBM source, a specific line was created, with the known information,

Besides that, the '_T' suffix and the 'MD' source identification have to be removed from samples names, because they do not exist on the expression side. The '.' character was replaced by '_' and the 'R' prefix was added.

The phenotype table, after normalization with the function *normPheno( )*, was put on the database using the *insertPheno( )* function.

5.5 Reannotation

This section describes the translation process from probeset identification to transcript identification with emphasis in the database modeling.

5.5.1 Probe sequences collection

Probe sequences were collected by vendor, gathering all data found. The result were several files, each corresponding to one platform. The load process for Affymetrix and Agilent was made with the *getProbesSequences( )* function, which extracts all probe-sequence pairs from files and discards repeated ones.

Illumina files were structurally different from the others and present errors when trying to read to a data frame. Thus, a specific function was developed for these data (*getIlluProbesSequences( )*), that extracts information directly from a GEO Illumina platform.

The insertion on the database is made with the function *insertSeqs( )*, where the structure of the table is specified (shown in Figure 18) and the 'probe_id' is defined as auto increment.

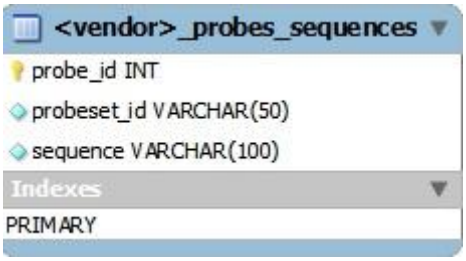


Figure 18. Database table to store vendor probes sequences.

5.5.2 Probes matches

The matches are done for each '<vendor>_probes_sequences' table, with the function *probesMatches( )*. Resorting to the *Biostrings* package, the sequences are converted to *DNAStringSet* objects for easier and more efficient manipulation, becoming part of a DNA dictionary. This dictionary is complemented to obtain a representation for each strand. Both representations are then aligned to the UCSC human genome (Figure 19), allowing a maximum of 1 base pair (bp) mismatch. Only the common 24 chromosomes (1 to 22, X and Y) were considered for the match.

```
Human genome
|
| organism: Homo sapiens (Human)
| provider: UCSC
| provider version: hg19
| release date: Feb. 2009
| release name: Genome Reference Consortium GRCh37
|
| single sequences (see '?seqnames'):
|   chr1                  chr2                  chr3                  chr4
|   chr5                  chr6                  chr7                  chr8
|   chr9                  chr10                 chr11                 chr12
|   chr13                 chr14                 chr15                 chr16
|   chr17                 chr18                 chr19                 chr20
|   chr21                 chr22                 chrX                  chrY
|   chrM                  chr1_gl000191_random  chr1_gl000192_random  chr4_ctg9_hap1
|   chr4_gl000193_random  chr4_gl000194_random  chr6_apd_hap1         chr6_cox_hap2
|   chr6_dbb_hap3         chr6_mann_hap4        chr6_mcf_hap5         chr6_qbl_hap6
|   chr6_ssto_hap7        chr7_gl000195_random  chr8_gl000196_random  chr8_gl000197_random
|   chr9_gl000198_random  chr9_gl000199_random  chr9_gl000200_random  chr9_gl000201_random
|   chr11_gl000202_random chr17_ctg5_hap1       chr17_gl000203_random chr17_gl000204_random
|   chr17_gl000205_random chr17_gl000206_random chr18_gl000207_random chr19_gl000208_random
|   chr19_gl000209_random chr21_gl000210_random chrUn_gl000211        chrUn_gl000212
|   chrUn_gl000213        chrUn_gl000214        chrUn_gl000215        chrUn_gl000216
|   chrUn_gl000217        chrUn_gl000218        chrUn_gl000219        chrUn_gl000220
|   chrUn_gl000221        chrUn_gl000222        chrUn_gl000223        chrUn_gl000224
|   chrUn_gl000225        chrUn_gl000226        chrUn_gl000227        chrUn_gl000228
|   chrUn_gl000229        chrUn_gl000230        chrUn_gl000231        chrUn_gl000232
|   chrUn_gl000233        chrUn_gl000234        chrUn_gl000235        chrUn_gl000236
|   chrUn_gl000237        chrUn_gl000238        chrUn_gl000239        chrUn_gl000240
|   chrUn_gl000241        chrUn_gl000242        chrUn_gl000243        chrUn_gl000244
|   chrUn_gl000245        chrUn_gl000246        chrUn_gl000247        chrUn_gl000248
|   chrUn_gl000249
```

Figure 19. UCSC Human Genome, version hg19.

The alignment and annotation processes are massive time and resources consuming. The first attempts of implementation did not complete the execution, being the processing time estimate in the order of weeks. Thus, the code structure was reorganized to reduce processing time, establishing the final records on less than a day.
Since the data is saved iteratively in the database, by *probesMatches( )*, the database table (Figure 20) must be created before, using the function *createProbesMatchesTable( )*. It is not a good structure in terms of clean code organization, but database insertions on the middle of the matching algorithm help to reduce processing time.

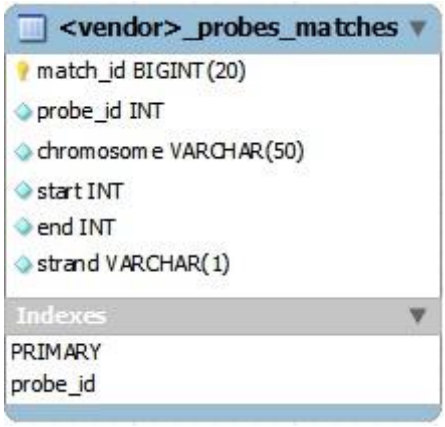


Figure 20. Database table to store probes matches.

5.5.3 Genes annotation

The *genesAnnot( )* function is used to gather gene annotation. All the necessary information was on the Ensembl database and it already had been downloaded with the BioMart online tool, so this function only fetches the information from file and selects that of interest. Afterwards, a table is created in the database (Figure 21), with the *createGenesAnnotTable( )* function, and the data is stored.
To uniquely identify the genes, the notation adopted on Ensembl was used. The name is the common designation of the gene (e.g. ZFYVE20), start and end marks the limits of the gene on the chromosome, and transcript_count is the number of different transcripts identified by Ensembl for each gene. The description gives non-normalized extra information about the gene (e.g. "microRNA 30c-2 [Source:HGNC Symbol;Acc:31627]").

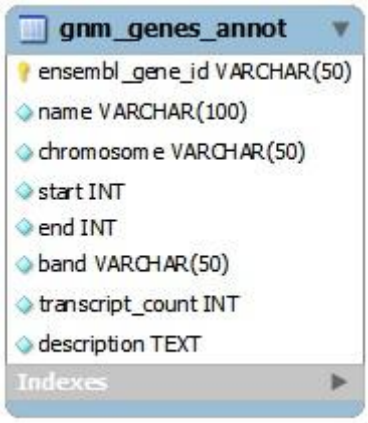


Figure 21. Database table to store genes annotation.

5.5.4 Mapping probes to genes

The mapping db table with the function *createProbesGenesTable( )* (Figure 22).



Figure 22. Database table with probes to genes mapping.

The *mapProbesGenes( )* function crosses information between probes matches and gene annotation, to find where there is a match inside the gene boundaries.

5.5.5 Mapping probesets to genes

The *mapProbesetsGenes( )* function crosses information from probe sequences and probes to genes mapping tables to establish a probeset-gene map. A probeset is associated with a gene if at least one of its probes is mapped to that gene. So, there may be cases of probesets associated with a gene only by one probe. To monitor that, a field 'absolut_freq' was added, that is a ratio between probeset matches on gene and total

matches for that probeset. In this way, in the future, probesets with a low frequency could be discarded.

The *probesetsGenesTable( )* function creates the probesets-genes map table (Figure 23) and fills it.

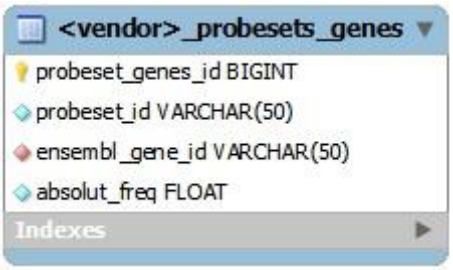


Figure 23. Database table with probesets to genes mapping.

5.5.6 Exons Annotation

This annotation is made with *exonsAnnot( )* function. For exons, something different from gene annotation was attempted, by using *biomaRt* package to load Ensembl data directly on R, instead of the online tool. While this is the logical choice, to use a tool already integrated on R, the genes annotation made earlier was made with the online tool because the *getMB( )* function has some drawbacks in terms of downloading time. Some fields on this result set are even not normalized, which demands a text treatment for each entry, leading to a huge processing time. This process was used here only after verifying that there were no entries with compound data, identified by the ';' separator, on each field (e.g.: `all(!grepl(";", exon.annotation$gene_exon)`). Duplicated entries were also detected, which were removed.

The data frame created by *exonsAnnot( )* is stored on the database table 'gnm_exons_annot' (Figure 24), with *exonsAnnotTable( )* function.

The 'sequence' field on table refers to de DNA sequence of the exon.



Figure 24. Database table to store exons annotation.

5.5.7 Mapping probes to exons

This process is similar to the one in the mapping of probes to genes. Table '<vendor>_probes_exons' (Figure 25) is created with function *createProbesExonsTable( )* and filled with *mapProbesExons( )* function.

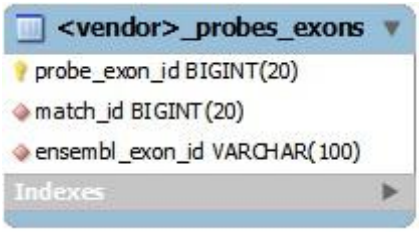


Figure 25. Database table with probes to exons mapping.

5.5.8 Transcripts annotation

Similar to genes annotation, *transcriptsAnnot( )* function gets data from an annotation file, while *transcriptsAnnotTable( )* creates and fills 'gnm_transcripts_annot' table (Figure 26).

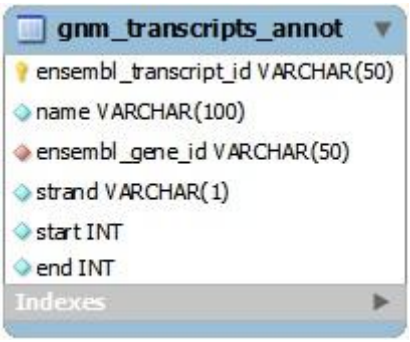


Figure 26. Database table to store transcripts annotation.

5.5.9 Mapping exons to transcripts

The database table is created with function *createExonsTranscriptsTable( )* and is named 'gnm_exons_transcripts' (Figure 27).



Figure 27. Database table with exons to transcripts mapping.

The *mapExonsTranscripts( )* function gets the information through a previous downloaded file, from Ensembl. The non common chromosome data is filtered, repeated entries are eliminated and it is verified if there are no composite fields.

5.5.10 Mapping probesets to transcripts

The database table is created with function *createProbesetsTranscriptsTable( )* and is named '<vendor>_probesets_transcripts' (Figure 28).

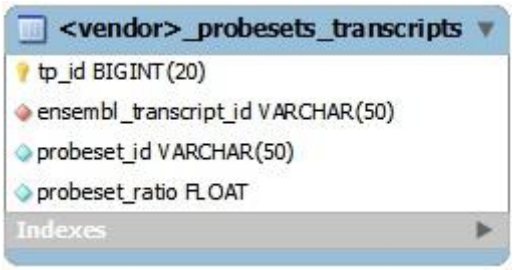


Figure 28. Database table with probesets to transcripts mapping.

The mapP*robesetsTranscripts( )* function is used here to cross information from probes-exons mapping, probes sequences, exons-transcripts mapping and obtain the number of times that the probes of some probeset match exons belonging to some transcript. This information is used to map probesets to transcripts and also to calculate probeset ratio (Eq. 14).

5.5.11 Rebuild expression datasets

The probeset-transcript mapping is applied here to rebuild expression datasets into a transcript-based annotation.

The *rebuildDatasets( )* function loads an expression data table from 'raw_expressions' and merges it with probeset-transcript mapping. The map entries that present a ratio lower than a specified threshold (value used was 0.5) are discarded, to avoid probeset-transcript associations with few probe readings, which would cause more damage than benefit of the final transcript expression value.

The expression value recalculation based on Eq. 15 requires a great amount of processing resources, so the *melt( )* function of package reshape was used to concatenate all sample columns on a single column and then apply the calculations. Even in this way, some datasets have to be split in subsets of no more than 250 samples to prevent system crashes along the process. The dataset is then reshaped to the original structure. It was also noted a great amount of RAM sequestered after the process, so a garbage collection function was included in the end.

A new database (hmg_annot_express) was created for rebuilt datasets. The new tables were created and filled using *insertRebDatasets( )* function, and their names differ from the formers on the suffix, where '_exp' became '_hexp' (e.g. gds1962_hexp).

5.5.12 The reannotation Enhanced Entity-Relationship diagram

The "Reannotation" stage is the only that is gone be shown the EER diagram, since the other stages are more data repositories without little or none relationship between tables. In Figure 29 is possible to see the tables associated with genome annotation, with the prefix "gnm_" and their relationships with the mapping tables specific for each manufacturer (Affymetrix, Agilent and Illumina), identified with the prefix "<vendor>_".

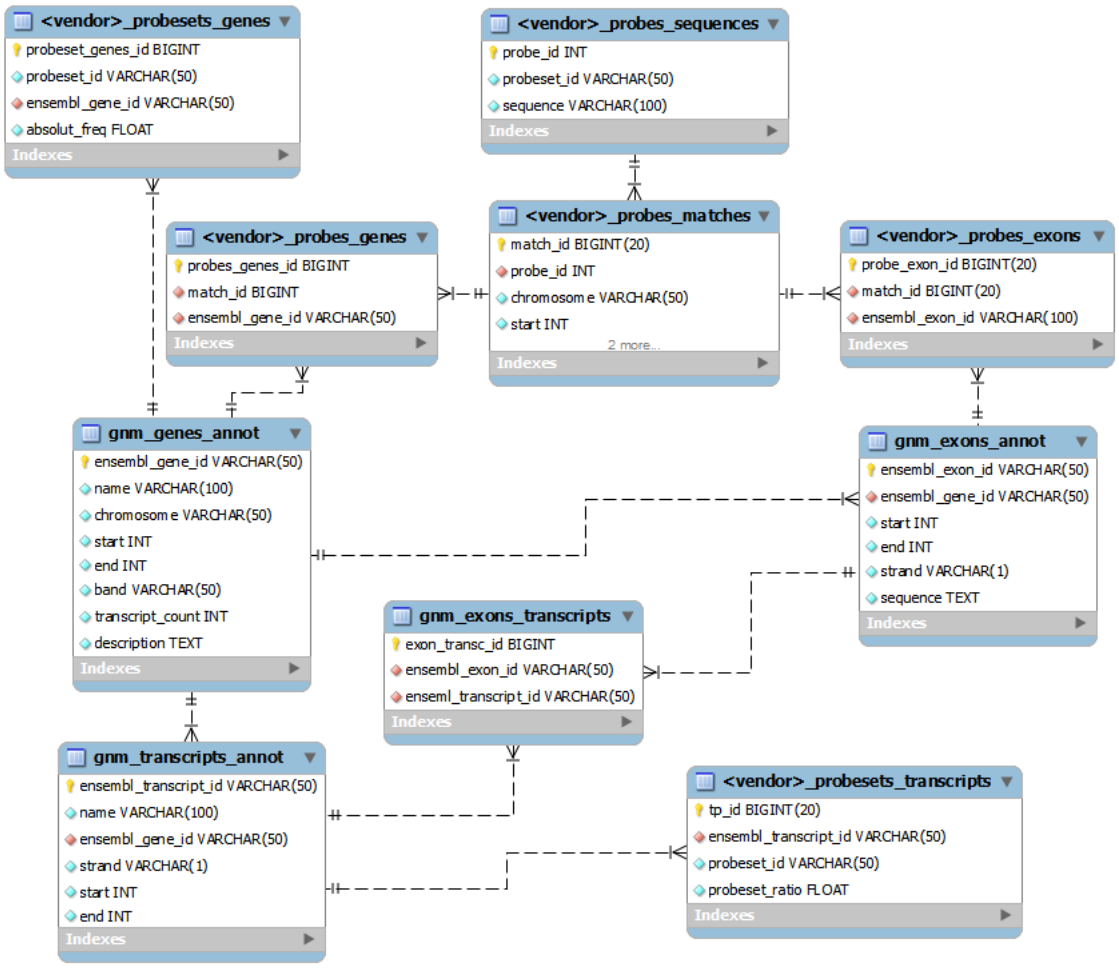


Figure 29. The reannotation Enhanced Entity-Relationship diagram.

5.6 Data integration

Datasets are joined using various methods that are applied by the following functions. The *joinRaw( )* function joins datasets as they are, without any transformation.

The *joinNorm( )* function applies a normalization based on mean annulment and standard deviation correction.

The *joinLinear( )* function finds the reference features with lower inter-class variation and use them to correct batch effect between datasets.

The *joinLinearLog2( )* function is the same as *joinLinear( )*, but applied to $\log_2$ scaled data.

The *joinGradeRef( )* function uses a class as reference between the datasets. The other class values are divided by the reference class to achieve a comparable scale between datasets. Note that is only possible to apply this function where all the datasets to integrate have the reference class.

The *joinGradeRefLog2( )* function is the same as *joinGradeRef ( )*, but applied to $\log_2$ scaled data.

5.7 Feature selection and classification

There are already classification functions with built in feature selection, but the huge number of variables present on a gene expression dataset is too much for the majority of them.

The *caret* package was tested for this task, since it includes the *foreach* package to implement parallel processing [125][126]. The main idea is to distribute processing burden for all processor cores to boost performance. However, this package was designed for Linux distributions, so the advantages disappear on Windows systems like the one used for this work. An adapted version of *foreach* was tried, that was able to use the four cores, but that was a greedy option that does not leave any processor capacity free for other tasks and results on short time in computer overheating and turn off. Thus, it was preferred to specify a recursive feature elimination algorithm from base.

Machine learning data mart

As classification focus on tumor WHO grade, a file based data mart was created, consisting on datasets with this classification incorporated and with features as columns. The samples appear as rows, being the sample identifiers the row names (Figure 30).

This saves time when it is needed to use a dataset for machine learning (ML), since the datasets in our database would need always a reshape on the process. These files are on RData format for faster loading (they are not for use outside R) and each file corresponds to one dataset. The dataset is loaded as a *data.frame* object.

	who_grade	1007_s_at	1053_at	117_at	121_at	1255_g_at
GSM97803	0	12.2091	8.44253	8.16942	10.57670	7.67807
GSM97804	0	11.4838	8.47167	7.63662	10.80880	6.89845
GSM97805	0	12.3854	8.32148	7.51412	10.87650	7.49346
GSM97878	2	13.0007	8.17243	8.82623	10.63420	2.67807
GSM97913	2	12.5523	8.68825	11.87210	10.79690	3.78660
GSM97802	3	12.7435	9.52239	8.03122	10.76560	8.30697
GSM97810	3	13.0111	8.54612	7.22400	9.92496	6.62936
GSM97815	3	12.9688	8.58421	7.96751	10.70420	7.07039
GSM97837	3	13.3375	8.23745	9.63081	10.61840	5.47249

Figure 30. Example of the structure of a machine learning dataset.

Filter methods

Three methods for filtering were established: multiple ANOVA; simple correlation; and empirical Bayes.

For multiple ANOVA, the function *getDfBestAnova( )* calculates ANOVA for each feature on the dataset, and returns a vector with the features ordered by p-value, representing the ones that have best inter-grade variation when compared to intra-grade variation.

For simple correlation, the function *getAbsCorrelated( )* performs Pearson's correlation between each feature and grade, returning a vector of features ordered by correlation coefficient. It is possible to choose a minimum correlation coefficient and/or a maximum number of features to return.

For empirical Bayes, the function *eBayesFeatureFilter( )* uses *limma* and *Biobase* packages to order the features by p-value on differential expression. For this, it is necessary to create an *ExpressionSet* (eSet) object with the dataset and a linear model over eSet and a model matrix. The linear model is then used to calculate empirical Bayes results all over the possible contrasts on dataset (e.g. grade2-grade0, grade3-grade2, etc.).

Linearly correlated features are an annoying problem on machine learning algorithms, since they represents more uninformative entries leading to worst results. Some of them are prepared to deal with this problem, but others simply does not accept collinear

variables as inputs. Since a good function on R to deal with this was not found, one was built. *getNonCollinear( )* function builds a set on non correlated features, on an accumulatively manner. The first feature enter directly for the set. The second is tested to verify if is correlated with the first one above a certain threshold. If it is, it is discarded, if not, it is added to the set. The other features are tested over all the features already on the set.

Wrapper methods

Wrapper methods are based on applying recursively a machine learning algorithm to measure the performance of a set of predicting features, starting from one to an established maximum of features. For these tasks, the functions used were: *getBestPredictorsOrdLogReg( )*, *getBestPredictorsSVM( )*, *getBestPredictorsKnn( )* and *getBestPredictorsLDA( )*, which have the ML algorithm used explicit on their names.
getRefs(), based on mANOVA and Person's correlation, is used to retrieve features whose expression maintains relatively stable between classes.

6 Conclusions

As it has been stressed along this work, gene expression data integration is an extensive multi-task process where it is not always possible to achieve a good quality assessment. Many authors have addressed this subject, but typically only focusing on some parts of the problem. The majority of the works addresses data integration with equivalent platforms, generally from Affymetrix. The multi-platform approach, even with platforms from the same vendor, is a more complex problem and thus, less addressed. The number of gene expression integration works decreases even more when the focus lies on multi-platform scenarios from different vendors.

The main aim of this work was to build a gene expression integration system from scratch, only with data from open-access databases, which was able to take these datasets, normalize their information, operate structural adaptations for integration capabilities and then provide some tools for model quality assessment.

The "Data Collection" stage was one of the most complex parts of the work, since it was needed to acquire an extensive knowledge about the biological/clinical aspects related to the data meaning and the organization of the source databases. The normalization of the phenotypic data has proven to be one of the most time consuming tasks, since the original data fields were not clearly explained neither it was possible the access direct information about their content.

The "Reannotation" process is a crucial step in the integration of gene expression data from different vendors. The original probeset-oriented annotation makes impossible the direct integration of data due to the use of different identifiers. Since each manufacturer decides which regions of the genome should be under measurement, and this is not a static assignment, a flexible method of reannotation is a great achievement. Gene-level

integration can generate divergent results, given alternative splicing. Confining expression information to a gene level basis means losing precious information about the different biological functions of genes. The separate analysis of microarray data followed by result integration is a common alternative, but suffers from the problem of few samples per experiment, which can lead to skewed conclusions.

The option for a transcript-oriented integration, presented in this thesis, preserves the proteomic structure, allowing a biological meaningful integration between data sets of different vendors. This approach improves the strength of biomarkers, by allowing to exclude those with divergent behavior between platforms, and associating each one to a specific function.

The data integration with datasets whose genesis was on different types, tools and methodologies for gene expression measurement is a step that could not make sense at a first glance. Even the general mathematic description that is said to be followed by data is not a reliable assumption, since all the processing applied on datasets to achieve, for example, background correction, give them particular characteristics that are not possible to isolate on the probeset-oriented values. To address this integration challenge, the key feature to be possible to join two heterogeneous datasets is the reference. Without a reference that could make a bridge between levels of two datasets, the data integration result only could achieve a good result by chance.

Glioblastoma is a disease that urgently needs new prognostic methods that provide an extra aid to the actual efforts. The actual methods are extremely invasive, and are taken in stages where there are little or none measures that could be taken to treat the tumor. Glioblastoma is the worst of the astrocytomas grades, used as case study for this thesis. It provided a categorical variable as output for supervised learning, involving choosing the best set of brain tumor grade predictors and the best algorithms to train them. This variable also was used to assess the quality of the results.

Using feature selection, it was possible to see that the filter methods reduce drastically the initial dimension of the datasets, and great differences among the efficiency between methods were not detected. SVMs stood out as the best classifier for the integrated data, together with the use of a reference grade to bring data to equivalent scales between

datasets. The level of data to use on the integration, gene or transcript, needs a more profound interpretation. The probesets used for expression measurement are exon centric and do not read all exons of a gene. Since the same exon could be part of several different transcripts, it is common to find many of them with the same expression value, which causes entropy to learning algorithms. Even filtering the 'clones', in many cases the expression value of a transcript is the same as the correspondent gene, when all the probesets measuring the gene also measure that transcript. It is than clear that it is necessary to upgrade the expression measurement method to be possible to use transcript levels effectively.

This thesis contributes to show that it is possible to integrate data from different microarray platforms, which increase the predictors capability to generalize, which was demonstrated with the application of the models on a complete new dataset, with good results. The integration leads to better biomarker discovery, based on broader universe of classified samples, that in turn increase the strength of possible diagnostic for this case study or similar.

The informatics tools needed to perform the integration and access data quality have been developed. It is possible to perform machine learning tasks to extract information from data, as was intended. The platform was successfully applied on glioblastoma case study, which opens doors to a future set of clinical tests based on the results.

This work originated two research articles, one focused on the Reannotation process [115] and other on integration and quality assessment of the integrated data [127] (See Appendix 1).

The IGEIS provides a structured base system for future data integration studies. It must be possible to easily apply the concept to a new case study, with the needed adaptations to the case-dependent Data Collection stage. As a future work, a good interface would be of interest, since the developed tools are at the business level, given that was not a requirement for this work.

The Illumina datasets take part on the process only until the phase of datasets restructuring, due to the lack of explicit clinical information to define the phenotype. Nonetheless, the base of the structure for this vendor is already made and, therefore, it

would be interesting to test a representative dataset over the methods used for Agilent and Affymetrix .

The homogeneity and quality of the data was always a doubt, which in turn may rise some questions about the assertiveness of the results. In future work, a test point between data collection and reannotation would be recommended. This stage could be based on unsupervised learning, like clustering, aiming to detect subtypes of phenotypic attributes that were not distinguished in this work and may present subtle variations on expression patterns (e.g. subtypes mesenchymal, proneural, proliferative, etc.). The 'normal' samples also should be tested, since their biological sources differ between datasets, which rise questions about their use as references. They were used when no other common reference existed between datasets.

A survival study also would be of interest. These data were collected and a set of graphical analysis would be not difficult to be done. That was not included in this work only for lack of time.

References

[1] George Sarton, *Introduction to the History of Science*.: Krieger Pub Co, 1975.

[2] Markham Geller, *Ancient Babylonian Medicine: Theory and Practice*.: Wiley-Blackwell, 2010.

[3] Vivian Nutton, *Ancient Medicine*.: Routledge, 2012.

[4] William Bateson and Gregor Mendel, *Mendel's Principles of Heredity*.: Dover Publications, 2010.

[5] Catherine Wilson, *The Invisible World: Early Modern Philosophy and the Invention of the Microscope*.: Princeton University Press, 1997.

[6] Leslie Pray, "Discovery of DNA Structure and Function: Watson and Crick," *Nature Education*, vol. 1, no. 1, p. 100, 2008.

[7] Allan Maxam and Walter Gilbert, "A new method for sequencing DNA," *Proceedings of the National Academy of Sciences of USA*, vol. 74, no. 2, pp. 560-564, 1977.

[8] Nyrén Pal, "The history of pyrosequencing," *Methods in Molecular Biology*, vol. 373, no. 1, pp. 1-14, 2007.

[9] Edwin Southern, "Detection of specific sequences among DNA fragments separated by gel electrophoresis," *Journal of Molecular Biology*, vol. 98, no. 3, pp. 503-517, 1975.

[10] Ashraf Ewis et al., "A history of microarrays in biomedicine," *Expert Review of Molecular Diagnostics*, vol. 5, no. 3, pp. 315-328, 2005.

[11] National Center for Biotechnology Information. (2007, August) Gene Expression Omnibus. [Online]. http://www.ncbi.nlm.nih.gov

[12] Sorin Drăghici, *Statistics and Data Analysis for Microarrays Using R and Bioconductor*, 2nd ed.: CRC Press, 2011.

[13] Johan Christersson et al., "Non-coding antisense transcription detected by conventional and single-stranded cDNA microarray," *BMC Genomics*, vol. 8, no. 295, 2007.

[14] Maurice Ling, Yuguang Ban, Hongxiu Wen, San Wang, and Steven Ge, "Conserved expression of natural antisense transcripts in mammals," *BMC Genomics*, vol. 14, no. 1, p. 243, 2013.

[15] Hui Yu et al., "Transcript-level annotation of Affymetrix probesets improves the interpretation of gene expression data," *BMC Bioinformatics*, vol. 8, no. 1, 2007.

[16] Luis Rueda and Adnan Ali, *Microarray Image and Data Analysis: Theory and Practice*.: CRC Press, 2014.

[17] Jack Newton, "Analysis of Microarray Gene Expression Data Using Machine Learning Techniques," 2002.

[18] Alper Küçükural, Reyyan Yeniterzi, Süveyda Yeniterzi, and O. Sezerman, "Evolutionary Selection of Minimum Number of Features for Classification of Gene Expression Data Using Genetic Algorithms," *In proceedings of the 9th annual conference on Genetic and evolutionary computation, GECCO*, vol. 9, pp. 401-406, 2007.

[19] Qingzhong Liu, A.H. Sung, Jianyun Xu, Jianzhong Liu, and Zhongxue Chen, "Microarray Gene Expression Classification Based on Supervised Learning and Similarity Measures," *IEEE International Conference on Systems, Man and Cybernetics, SMC '06.*, vol. 6, pp. 5094-5099, 2006.

[20] Gregor Štiglic and Peter Kokol, "Using Multi-Agent System for Gene Expression Classification," *In Proceedings of 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2, pp. 2952-2955, 2004.

[21] Sabah Khalid, Mohsin Khan, Ping Wang, Xiaohui Liu, and Su Li, "Application of Bioinformatics in the Design of Gene Expression Microarrays," in *2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2006)*, Paphos, 2006, pp. 146-160.

[22] Ana C. Lorena, G. Ivan Costa, and Marcilio C.P. Souto, "On the complexity of gene expression classification data sets," *8th International Conference on Hybrid Intelligent Systems*, pp. 825-830, 2008.

[23] T.R. Golub et al., "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, vol. 286, no. 5439, pp. 531-537, 1999.

[24] Jane Liu et al., "Multiclass cancer classification and biomarker discovery using GA-based algorithms," *Bioinformatics*, vol. 21, no. 11, pp. 2691-2697, 2005.

[25] Sung Bae Cho and Jungwon Ryu, "Classifying Gene Expression Data of Cancer Using Classifier Ensemble with Mutually Exclusive Features," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1744-1753, 2002.

[26] Monika Ray, Johannes Freudenberg, and Weixiong Zhang, "A Comprehensive Analysis of the Effect of Microarray Data Preprocessing Methods on Differentially Expressed Transcript Selection," in *Methods in Microarray Normalization*, Phillip Stafford, Ed.: CRC Press, 2008.

[27] Affymetrix, *GeneChip Expression Analysis Technical Manual*, 3rd ed.: Affymetrix Inc., 2009.

[28] Agilent Technologies, *Two-Color Microarray-Based Gene Expression Analysis*, 65th ed.: Agilent Technologies, Inc., 2010.

[29] Agilent. (2015, March) Universal Reference RNAs - Details & Specifications. [Online]. http://www.genomics.agilent.com/article.jsp?pageId=1452

[30] Marianna Zahurak et al., "Pre-processing Agilent microarray data," *BMC Bioinformatics*, vol. 8, no. 142, 2007.

[31] Genomics and Bioinformatics Group. (2015, March) Microarray Data Analysis. [Online]. http://discover.nci.nih.gov/microarrayAnalysis/Two.Color.Preprocessing.jsp

[32] Benoît Ballester, Nathan Johnson, Glenn Proctor, and Paul Flicek, "Consistent annotation of gene expression arrays," *BMC Genomics*, vol. 11, no. 294, 2010.

[33] Eric Rouchka, Abhijit Phatak, and Amar. Singh, "Effect of single nucleotide polymorphisms on Affymetrix match-mismatch probe pairs," *Bioinformation*, vol. 2, no. 9, pp. 405-411, 2008.

[34] Affymetrix, *Statistical Algorithms Description Document*.: Affymetrix, Inc., 2002.

[35] Ben Bolstad, "Preprocessing and normalization for Affymetrix GeneChip

Expression Microarrays," in *Methods in Microarray Normalization*, Phillip Stafford, Ed.: CRC Press, 2008.

[36] Stuart Pepper, Emma Saunders, Laura Edwards, Claire Wilson, and Crispin Miller, "The utility of MAS5 expression summary and detection call algorithms," *BMC Bioinformatics*, vol. 8, p. 273, 2007.

[37] Pedro Romero, "Pre-processing and differential expression analysis of Agilent microRNA arrays using the AgiMicroRna Bioconductor library," *BMC Genomics*, vol. 12, p. 64, 2011.

[38] Isaac Kohane, Alvin Kho, and Atul Butte, *Microarrays for an Integrative Genomics*.: MIT Press, 2005.

[39] Illumina, BeadStudio Normalization Algorithms for Gene Expression Data, 2011, Technote.

[40] Dov Stekel, *Microarray Bioinformatics*, 1st ed.: Cambridge University Press, 2003.

[41] Hinanit Koltai and Carmiya Baror, "Specificity of DNA microarray hybridization characterization, effectors and approaches for data correction," *Nucleic Acids Research*, vol. 36, no. 7, pp. 2395-2405, 2008.

[42] Steen Knudsen, *Guide to Analysis of DNA Microarray Data*, 2nd ed.: John Wiley & Sons, 2004.

[43] Jason Hsu, *Multiple Comparisons: Theory and Methods*.: Chapman and Hall, 1996.

[44] J. Bland and Douglas Altman, "Multiple significance tests: the Bonferroni method," *Statistic Notes*, vol. 310, no. 6973, p. 170, 1995.

[45] Zbyněk Šidák, "Rectangular Confidence Regions for the Means of Multivariate Normal Distributions," *Journal of the American Statistical Association*, vol. 62, no. 318, pp. 623-633, 1967.

[46] Hervé Abdi, "Holm's Sequential Bonferroni Procedure," in *Encyclopedia of Research Design*, Neil Salkind, Ed.: Sage Publications, 2010.

[47] Aidong Zhang, *Advanced Analysis Of Gene Expression Microarray Data*.: World Scientific Publishing, 2006.

[48] Zoubin Ghahramani, "Unsupervised Learning," *Advanced Lectures on Machine*

Learning, vol. 3176, pp. 72-112, 2004.

[49] Leping Li, Clarice Weinberg, Thomas Darden, and Lee Pedersen, "Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method," *Bioinformatics*, vol. 17, no. 12, pp. 1131-1142, 2001.

[50] Gunnar Carlsson and Facundo Mémoli, "Characterization, Stability and Convergence of Hierarchical Clustering Methods," *Journal of Machine Learning Research*, vol. 11, pp. 1425-1470, 2010.

[51] Tapas Kanungo, Nathan Netanyahu, and Angela Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 24, no. 7, pp. 881-892, 2002.

[52] S. Ramaswamy et al., "Multiclass cancer diagnosis using tumour gene expression signatures," *Proceedings of the National Academy of Science*, vol. 98, no. 26, pp. 15149-15154, 2001.

[53] Ian Jolliffe, *Principal Component Analysis*, 2nd ed.: Springer, 2002.

[54] Pádraig Cunningham, Matthieu Cord, and Sarah Delany, , Matthieu Cord and Pádraig Cunningham, Eds.: Springer, 2008, ch. 2, pp. 21-49.

[55] Jianping Huang, Hong Fang, and Xiaohui Fan, "Decision forest for classification of gene expression data," *Computers in Biology and Medicine*, vol. 40, no. 8, pp. 698-704, 2010.

[56] Shin Ando and Hitoshi Iba, "Classification of Gene Expression Profile Using Combinatory Method of Evolutionary Computation and Machine Learning," *Genetic Programming and Evolvable Machines*, vol. 5, no. 2, pp. 145-156, 2004.

[57] Li Chuang, Kuo Wu, and Cheng Yang, "A Hybrid Feature Selection Method Using Gene Expression Data," in *Ninth IEEE International Conference on Bioinformatics and Bioengineering, BIBE*, Taichung, 2009.

[58] Zvi Boger, "Artificial Neural Networks Methods for Identification of the Most Relevant Genes from Gene Expression Array Data," *Proceeding of the International Joint Conference on Neural Networks, IJCNN*, vol. 4, pp. 3095-3100, 2003.

[59] Wutao Chen, Huijuan Lu, Mingyi Wang, and Cheng Fang, "Gene Expression

Data Classification Using Artificial Neural Network Ensembles Based on Samples Filtering," in *International Conference on Artificial Intelligence and Computational Intelligence, AICI*, Shanghai, 2009.

[60] Lipo Wang, Feng Chu, and Wei Xie, "Accurate Cancer Classification Using Expressions of Very Few Genes," *IEEE/ACM Tansactions on Computational Biology and Bioinformatics, TCBB*, vol. 4, no. 1, pp. 40-53, 2007.

[61] Yonghong Peng, Zhiqing Wu, and Jianmin Jiang, "A novel feature selection approach for biomedical data classification," *Biomedical Informatics*, vol. 43, no. 1, pp. 15-23, 2010.

[62] Michael Brown et al., "Knowledge-based analysis of microarray gene expression data by using support vector machines," *In Proceedings of the National Academy of Science*, vol. 97, no. 1, pp. 262-267, 2000.

[63] David Hosmer, Stanley Lemeshow, and Rodney Sturdivant, *Applied Logistic Regression*, 3rd ed.: Wiley, 2013.

[64] Alfredo Benso, Stefano Di Carlo, Gianfranco Politano, and Luca Sterpone, "A Graph-Based Representation of Gene Expression Profiles in DNA Microarrays," in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Sun Valley, USA, 2008.

[65] J. Deutsch, "Evolutionary algorithms for finding optimal gene sets in microarray prediction," *Bioinformatics*, vol. 19, no. 1, pp. 45-52, 2003.

[66] C. Ooi and Patrick Tan, "Genetic algorithms applied to multi-class prediction for the analysis of gene expression data," *Bioinformatics*, vol. 19, no. 1, pp. 37-44, 2003.

[67] Kalyanmoy Deb and A. Reddy, "Reliable classification of two-class cancer data using evolutionary algorithms," *BioSystems*, vol. 72, no. 1-2, pp. 111-129, 2003.

[68] Keivan Kianmehr, Mohammed Alshalalfa, and Reda Alhajj, "Fuzzy clustering-based discretization for gene expression classification," *Knowldedge and Information Systems*, vol. 24, no. 3, pp. 441-465, 2009.

[69] Michalis Zervakis et al., "Outcome prediction based on microarray analysis: a critical perspective on methods," *BMC Bioinformatics*, vol. 10, no. 53, 2009.

[70] Kyung Kim and Sung Cho, "DNA gene expression classification with ensemble classifiers optimized by speciated genetic algorithm," in *Pattern Recognition and*

Machine Intelligence, LNCS. Kolkata: Springer Berlin Heidelberg, 2005, vol. 3776, pp. 649-653.

[71] Thanyaluk Umpai and Stuart Aitken, "Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes," *BMC Bioinformatics*, vol. 6, no. 148, 2005.

[72] Koen Bossers et al., "Intensity-based analysis of dual-color gene expression data as an alternative to ratio-based analysis to enhance reproducibility," *BMC Genomics*, vol. 11, no. 112, 2010.

[73] Michel Bellis, "Mapping of Affymetrix probe sets to groups of transcripts using transcriptional networks," Cornell University, arXiv:1201.2033v2, 2012.

[74] Elena Kostadinova, "Data Integration: An Approach to Improve The Preprocessing and Analysis of Gene Expression Data," *Union of Scientists in Bulgaria*, vol. 6, pp. 120-133, 2013.

[75] Chen Meng, Bernhard Kuster, Aedín Culhane, and Amin Gholami, "A multivariate approach to the integration of multi-omics datasets," *BMC Bioinformatics*, vol. 15, pp. 162-175, 2014.

[76] Anna Papiez, Paul Finnon, Christophe Badie, Simon Bouffler, and Joanna Polanska, "Integrating Expression Data from Different Microarray Platforms in Search of Biomarkers of Radiosensitivity," in *IWBBIO*, vol. 1, Granada, 2014, pp. 484-493.

[77] Chris Willis, Kyle Whyte, Melinda Baker, and Richard Pestell, "Integrating transcriptomic data using metacore pathway analysis to identify novel biomarkers of bevacizumab target engagement," in *American Association for Cancer Research*, vol. 74, San Diego, 2014.

[78] Aedín C. Culhane, Guy Perrière, and Desmond G. Higgins, "Cross-platform comparison and visualisation of gene expression data using co-inertia analysis," *BMC Bioinformatics*, vol. 4, no. 1, p. 59, 2003.

[79] Yong Woo et al., "A comparison of cDNA, oligonucleotide, and Affymetrix GeneChip gene expression microarray platforms," *Journal of Biomolecular Techniques*, vol. 15, no. 4, pp. 276–284, 2004.

[80] Anna-Kaarina Jarvinen et al., "Are data from different gene expression microarray platforms comparable?," *Genomics*, vol. 83, pp. 1164–1168, 2004.

[81] Jung Kyoon Choi, Ungsik Yu, Sangsoo Kim, and Ook Joon Yoo, "Combining multiple microarray studies and modeling interstudy variation," *Bioinformatics*, vol. 19, no. 1, pp. 84–90, 2003.

[82] Bingmei Zhu, Guichen Ping, Yasuo Shinohara, Yong Zhang, and Yoshinobu Baba, "Comparison of gene expression measurements from cDNA and 60-mer oligonucleotide microarrays," *Genomics*, vol. 85, pp. 657–665, 2005.

[83] Chintanu Kumar Sarmah and Sandhya Samarasinghe, "Microarray gene expression: A study of between-platform association of Affymetrix and cDNAarrays," *Computers in Biology and Medicine*, vol. 41, pp. 980-986, 2011.

[84] Cynthia Stretch et al., "Effects of sample size on differential gene expression, rank order and prediction accuracy of a gene signature," *PLoS ONE*, vol. 8, no. 6, 2013.

[85] David A. Elizondo, Benjamin N. Passow, Ralph Birkenhead, and Andreas Huemer, "Dimensionality reduction and microarray data," in *Principal Manifolds for Data Visualization and Dimension Reduction*. Leicester, UK: Springer Berlin Heidelberg, 2008, pp. 293-308.

[86] Chao Chen et al., "Removing Batch Effects in Analysis of Expression Microarray Data: An Evaluation of Six Batch Adjustment Methods," *PLoS ONE*, vol. 6, no. 2, 2011.

[87] W. Johnson, Cheng Li, and A. Rabinovic, "Adjusting batch effects in microarray expression data using empirical Bayes methods," *Biostatistics*, vol. 8, no. 1, pp. 118-127, 2007.

[88] Eli Eisenberg and Erez Levanon, "Human housekeeping genes are compact," *Trends in Genetics*, vol. 19, no. 7, pp. 362-365, 2003.

[89] Johann Bartsch and Terence Speed, "Using control genes to correct for unwanted variation in microarray data," *Biostatistics*, vol. 13, no. 3, pp. 539-552, 2012.

[90] D. Louis, E. Holland, and J. Cairncross, "Glioma Classification: A Molecular Reappraisal," *The American journal of pathology*, vol. 159, no. 3, pp. 779-786, 2001.

[91] Dimitri Agamanolis. (2014, October) Tumors of the central nervous system. [Online]. http://neuropathology-web.org/chapter7/chapter7bGliomas.html

[92] B. M. et al. Costa, "Reversing HOXA9 oncogene activation by PI3K inhibition:

epigenetic mechanism and prognostic significance in human glioblastoma," in *Research, vol. 70, Issue 2, pp. 453-62*, 2010.

[93] Mairéad McNamara, Solmaz Sahebjam, and Warren Mason, "Emerging Biomarkers in Glioblastoma," *Cancers*, vol. 5, pp. 1103-1119, 2013.

[94] Heidi Phillips et al., "Molecular subclasses of high-grade glioma predict prognosis, delineate a pattern of disease progression, and resemble stages in neurogenesis," *Cancer Cell*, vol. 9, no. 3, pp. 157-173, 2006.

[95] Ron Edgar, Michael Domrachev, and Alex Lash, "Gene Expression Omnibus: NCBI gene expression and hybridization array data repository," *Nucleic Acids Research*, vol. 30, no. 1, pp. 207-210, 2002.

[96] Sean Davis and Paul Meltzer, "GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor," *Bioinformatics*, vol. 23, no. 14, pp. 1846-1847, 2007.

[97] Tanya Barret and Ron Edgar, "Mining Microarray Data at NCBI's Gene Expression Omnibus," in *Gene Mapping, Discovery, and Expression. Methods and Protocols*, M. Bina, Ed.: Human Press, 2006, ch. 14, pp. 175-190.

[98] The Cancer Genome Atlas Research Network et al., "The Cancer Genome Atlas Pan-Cancer analysis project," *Nature Genetics*, vol. 45, no. 10, pp. 1113–1120, 2013.

[99] Paul Flicek et al., "Nucleic Acids Research," Ensembl, Technical report doi:10.1093/nar/gks1236, 2013.

[100] Soumen Khatua et al., "Overexpression of the EGFR/FKBP12/HIF-2alpha pathway identified in childhood astrocytomas by angiogenesis gene profiling," *Cancer Research*, vol. 63, no. 8, pp. 1865-1870, 2003.

[101] Tsunglin Liu et al., "Detection of a microRNA signal in an in vivo expression set of mRNAs," *PLoS ONE*, vol. 2, no. 8, 2007.

[102] Lixin Sun et al., "Neuronal and glioma-derived stem cell factor induces angiogenesis within the brain," *Cancer Cell*, vol. 9, no. 4, pp. 287-300, 2006.

[103] William Freije et al., "Gene expression profiling of gliomas strongly predicts survival," *Cancer Research*, vol. 64, no. 18, pp. 6503-6510, 2004.

[104] Maria Carro et al., "The transcriptional network for mesenchymal transformation

of brain tumours," *Nature*, vol. 463, no. 7279, pp. 318-325, 2010.

[105] Douglas Ross et al., "Systematic variation in gene expression patterns in human cancer cell lines," *Nature Genetics*, vol. 24, no. 3, pp. 227-234, 2000.

[106] Rickard Sandberg and Ingemar Ernberg, "Assessment of tumor characteristic gene expression in cell lines using a tissue similarity index (TSI)," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 6, pp. 2052-2057, 2005.

[107] Affymetrix, "GeneChip® Human Genome Arrays - Data Sheet," 2004.

[108] Affymetrix, "Array Design for the GeneChip® Human Genome U133 Set ," Affymetrix, Technical Note Part Number 701133 Rev. 2, 2007.

[109] Pan Du, Warren Kibbe, and Simon Lin, "lumi: a pipeline for processing Illumina microarray," *Bioinformatics*, vol. 24, no. 13, pp. 1547-1548, 2008.

[110] Yingdong Zhao and Richard Simon, "BRB-ArrayTools Data Archive for human cancer gene expression: a unique and efficient data sharing resource," *Cancer Informatics*, vol. 6, pp. 9-15, 2008.

[111] Giulio Genovese, Robert E. Handsaker, Heng Li, Eimear E. Kenny, and Steven A. McCarroll, "Mapping the human reference genome's missing sequence by three-way admixture in latino genomes," *The American Journal of Human Genetics*, vol. 93, pp. 1-11, 2013.

[112] George Asimenos. (2013, June) Human Genome. [Online]. https://wiki.dnanexus.com/Scientific-Notes/human-genome

[113] Francesco Ferrari et al., "Novel definition files for human GeneChips based on GeneAnnot," *BMC Bioinformatics*, vol. 8, no. 1, p. 446, 2007.

[114] G. K. Smyth, "Limma: linear models for microarray data," in *Bioinformatics and Computational Biology Solutions using R and Bioconductor.* New York: Springer, 2005.

[115] Eduardo Valente and Miguel Rocha, "Transcript-based Reannotation for Microarray Probesets," in *ACM Symposium on Applied Computing*, Salamanca, 2015.

[116] Romain Bourdon. WampServer. [Online]. http://www.wampserver.com/en/

[117] MySQL. (2015) MySQL Workbench. [Online].

https://www.mysql.com/products/workbench/

[118] Scientific Community. (2015, January) BioMart. [Online]. http://www.biomart.org/

[119] TCGA. (2015) TCGA Data Portal. [Online]. https://tcga-data.nci.nih.gov/tcga/

[120] National Cancer Institute. (2015, January) BRB-ArrayTools. [Online]. http://linus.nci.nih.gov/BRB-ArrayTools.html

[121] James Kent, "BLAT - The BLAST-Like Alignment Tool," *Genome Research*, vol. 12, no. 4, pp. 656-664, 2002.

[122] W. Venables, D. Smith, and R Core Team, *An Introduction to R. Notes on R: A Programming Environment for Data Analysis and Graphics*.: R Core Team, 2015.

[123] Open source. (2015, April) Bioconductor. [Online]. http://www.bioconductor.org/

[124] Oracle. Java SE 6 HotSpot[tm] Virtual Machine Garbage Collection Tuning. [Online]. http://www.oracle.com/technetwork/java/javase/gc-tuning-6-140523.html#par%5Fgc.oom

[125] Max Kuhn. (2014, December) Caret. [Online]. http://topepo.github.io/caret/parallel.html

[126] Max Kuhn, "Building Predictive Models in R Using the caret Package," *Journal of Statistical Software*, vol. 28, no. 5, 2008.

[127] Eduardo Valente and Miguel Rocha, "(In Press) Microarray gene expression data integration: an application to brain tumor grade determination," in *Practical Applications of Computational Biology & Bioinformatics*, Salamanca, 2015.

Appendix A

Publications derived from this thesis

Eduardo Valente and Miguel Rocha, "Transcript-based Reannotation for Microarray Probesets," in *ACM Symposium on Applied Computing*, Salamanca, 2015.

Abstract

DNA microarrays are one of the most used technologies for gene expression measurement. However, there are several distinct microarray platforms, from different manufacturers, each with its own measurement protocol, resulting in data that can hardly be compared or directly integrated. Data integration from multiple sources aims to improve the assertiveness of statistical tests, reducing the data dimensionality problem. This work intends to establish a basis for the integration of gene expression measurements from several manufacturers, a problem that can be addressed at different levels. We will focus on the reannotation process, a cornerstone of multi-platform integration. The proposed approach is based on a reannotation from probesets to transcripts, preserving valuable information for further analysis. Gene expression data from glioblastoma studies will be used as case studies, considering data from Agilent and Affymetrix platforms

Eduardo Valente and Miguel Rocha, "(In Press) Microarray gene expression data integration: an application to brain tumor grade determination," in *Practical Applications of Computational Biology & Bioinformatics*, Salamanca, 2015.

Abstract

World Health Organization ranks brain tumors in four stages, being the fourth grade the most aggressive. Glioblastoma, a fourth grade tumor, is one of the most severe human diseases that almost inevitability leads to death. Physicians address the classification in grades through direct inspection. Indeed, there is a need for good automatic predictors of tumor grade, which are not affected by human misclassification errors and that can be

made with less invasive diagnostic tools. This work address the stages involved in the process of selecting a good tumor grade predictor, based on microarray gene expression data. In this work, the information integration from heterogeneous platforms is highlighted, evidencing the particularities of choosing approaches working at gene, transcript or probeset levels. Distinct machine learning algorithms and integration methods are tested, analyzing their ability to produce a good set of predictors for tumor grade

Appendix B

Reference Manual

Data collection

getGDSDataset

Title

Get a dataset from GEO

Required packages

GEOquery

Arguments

gdsID (character)

Name of the dataset on the GEO database. Identifies if it requires to download it from online source.

gdsPath (character)

Absolute path to previous stored dataset file.

Details

The *getGEO* function is used to get a GEO entity. The source could be a previous downloaded file, or the dataset id and, in this case, it is tried to download directly the file from GEO.

The dataset is then converted to an R data frame, and the field "IDENTIFIER", which contains the gene associated with each probeset, is removed.

Value

data.frame

An expression table, where the first column has the probeset ids and the other columns refer to samples.

getDualGDSDataset

Title

Get two GEO complementary datasets and convert into one.

Required packages

GEOquery

Arguments

gdsID1 (character)

Name of the first dataset on GEO database. Identifies if it requires to download it from online source.

gdsID2 (character)

Name of the second dataset on GEO database.

gdsPath1 (character)

Absolute path to previously stored first dataset file.

gdsPath2 (character)

Absolute path to previously stored second dataset file.

Details

Get two complementary datasets, clean batch effect and concatenate them by row, increasing the number of probesets.

Value

data.frame

A concatenated expression table.

getGSEDataset

Title

Get expression dataset from a GEO series.

Required packages

GEOquery

Arguments

gseID (character)

Identification of the series on GEO database.

gsePath (character)

Absolute path to previously stored series file.

esetNumb (numeric)

Position of the dataset intended to retrieve. Default is 1.

Details

A series may contain several datasets associated, on the form of an ExpressionSet. So, to retrieve a specific dataset, it is needed to indicate the position of the dataset (esetNumb) in the series. The default is 1, since the majority of series only have one dataset associated.

Value

data.frame

An expression table.

getTCGADataset

Title

Get expression dataset from TCGA files.

Arguments

dsetID (character)

Identification of the dataset on TCGA files.

tumorLevel (character)

Identifier of the study to which the files are related.

vendor (character)

The microarray manufacturer. Used to define if the platform is 1 or 2 channel.

verbose (logical)

Where to display or not algorithm information during function runtime execution.

Details

This function is designed to extract an expression data frame from files downloaded through TCGA Data Portal. It is considered that the structure of these files and containing folders is not modified.

Value

data.frame

An expression table.

getRMBDataset

Title

Get expression dataset from Rembrandt files.

Required packages

affy

Arguments

expDirPath (character)

Path to directory where Rembrandt CEL files are stored.

verbose (logical)

Where to display or not algorithm information during function runtime execution.

Details

The CEL files are converted on an ExpressionSet object with all the samples. Conversion is made using the RMA algorithm, with batches of 50 samples. Samples names are normalized to correspond to phenotype samples.

Value

data.frame

An expression table.

normalizeExpressionDF

Title

Perform 'cleaning' operations over an expression data frame

Arguments

dataset (data.frame)

The data frame to be cleaned.

do.log2 (boolean)

Whether or not to perform $\log_2$ transformation.

verbose (boolean)

Whether or not to show information about algorithm execution.

Details

Prepare a GEO dataset to be introduced on a database.

Remove control probesets, handle NAs and perform $\log_2$ transformation.

Value

data.frame

A normalized expression data frame.

handleNAs

Title

Remove missing values (NA) from data.

Arguments

expData (numeric data.frame or matrix)

Expression dataset with NA values.

verbose (logical)

Whether or not show information about algorithm execution.

Details

Replace missing values (NAs) based on the best correlated column of the rest of dataset. A linear model is used from the most correlated column to the one with NAs to predict the new value. If it is not possible to find a new value, it is tried with the next most correlated column, until one is found without an NA on that position. If it is not found, the entire row is removed.

Value

data.frame

An expression data frame without NAs.

getCorrelated

Title

Find the correlation of one column with columns of a data frame

Arguments

column (numeric vector)

The column to compare with all columns on a data frame.

data (numeric data.frame)

The data frame.

Details

It is calculated the correlation between the vector and each column of the data frame. The names of the data frame columns are ordered by descending correlation.

Value

character

A vector with the names of the columns ordered by correlation.

dbInsertExpDF

Title

Insert an expression dataset on the local database.

Required packages

RMySQL

Arguments

gdsID (character)

Name of the dataset on GEO database.

Details

Stores dataset on local database. The name of the table is the original dataset ID with the suffix "_exp".

Value

none

getGDSPheno

Title

Get phenotype data.

Required packages

GEOquery

Arguments

dSetName (character)

Name of the dataset on GEO database.

gdsPath (character)

Absolute path to previous stored dataset file.

Details

Extract phenotype data from GDS file.

Value

data.frame

Table with phenotype.

getGSEPheno

Title

Get phenotype dataset from a GEO series.

Required packages

GEOquery

Arguments

gseID (character)

Identification of the series on GEO database.

gsePath (character)

Absolute path to previous stored series file.

esetNumb (numeric)

Position of the dataset intended to retrieve. Default is 1.

Details

A series may contain several datasets associated, on the form of an ExpressionSet. So, to retrieve a specific dataset, it is needed to indicate the position of the dataset (esetNumb) in the series. The default is 1, since the majority of series only have one dataset associated.

Value

data.frame

A phenotype table.

getTCGAPheno

Title

Get phenotype dataset from TCGA files.

Arguments

grade (character)

Identification of the disease from where to get clinical file.

tableName (character)

Name given to corresponding expression table on database.

Details

A phenotype table is created based on information from the clinical file downloaded from TCGA, which is complemented with aliquot barcodes information.

Value

data.frame

A phenotype table.

getRMBPheno

Title

Get phenotype data from Rembrandt files.

Required packages

XLConnect

Arguments

phenoFilePath (character)

Absolute path to spreadsheet file with the Rembrandt phenotype data.

Details

Extracts phenotypic data from a spreadsheet. Sample names are normalized to meet the one in the expression data.

Value

data.frame

A phenotype table.

normPheno

Title

Phenotype normalization.

Required packages

XLConnect

Arguments

dSetName (character)

Name of the dataset on GEO database.

phenoTable (data.frame)

Table with the original phenotype attributes.

phenoMetaPath

Path to the spreadsheet with normalization rules.

Details

Translates original phenotype attributes into new standardized attributes, based on conversion rules from a spreadsheet. The spreadsheet has the name "<gdsID>_PhenoMeta.xlsx".

Value

data.frame

The normalized phenotype.

insertPheno

Title

Insert phenotype data in the database.

Required packages

RMySQL; XLConnect

Arguments

dSetName (character)

Name of the dataset.

phenoDf (data.frame)

Table with the phenotype attributes.

phenoMetaPath

Path to the spreadsheet with normalization rules.

Details

Create a table in the database, with the attributes present in phenoDf, to which are applied the types explicit in phenoMetaPath, and then fill the table with phenotype data.

Value

none

Reannotation

mapProbesetsTranscripts

Title

Establish an association between probesets and transcripts.

Required packages

RMySQL, plyr

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

Details

Find where each probeset has probes which match on genome overlap a transcript area and, in those cases, establish a probeset-transcript link. It is calculated the ratio, that is the number of matches a probeset transcript has over all the matches of that probeset. The information is inserted in database.

Value

data.frame

A table with probesets-transcripts mapping.

createProbesetsTranscriptsTable

Title

Create probesets-transcripts mapping table.

Required packages

RMySQL

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

Details

Create table '<vendor>_probesets_transcripts' with probesets-transcripts mapping.

Value

none

mapExonsTranscripts

Title

Establish an association between exons and transcripts.

Arguments

annotFilePath (character)

Path to file with information needed to mapping.

Details

Reads mapping information from file and performs operations of data cleaning and selection.

Value

data.frame

A table with exons-transcripts mapping.

exonsTranscriptsTable

Title

Create and fill exons-transcripts mapping table.

Required packages

RMySQL

Arguments

exTrcs (data.frame)

A table with exons-transcripts mapping.

Details

Create table 'gnm_exons_transcripts' and fills it with exons-transcripts mapping information.

Value

none

transcriptsAnnotTable

Title

Create and fill transcripts annotation table.

Required packages

RMySQL

Arguments

transcriptsAnnot (data.frame)

Table with annotation data.

Details

Creates the table "gnm_transcripts_annotation" in the database and fills it with data from transcriptsAnnot data frame.

Value

none

transcriptsAnnot

Title

Transcripts annotation.

Arguments

annotFilePath (character)

Path to file where is stored the annotation information.

Details

Get transcript annotation from file, extracting only the desired fields.

Value

data.frame

Transcripts annotation table.

mapProbesExons

Title

Establish an association between probes and exons.

Required packages

RMySQL

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

Details

Find where each probe matches on genome overlap an exon area and, in those cases, save that link on the database.

Value

none

createProbesExonsTable

Title

Create table for probes-exons mapping.

Required packages

RMySQL

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

Details

Create a static database table for probes-exons mapping data, with the name '<vendor>_probes_exons'.

Value

none

exonsAnnotTable

Title

Create and fills exons annotation table.

Required packages

RMySQL

Arguments

exonsAnnot (data.frame)

Table with annotation data.

Details

Creates the table "gnm_exons_annotation" in the database and fills it with data from exonsAnnot data frame.

Value

none

exonsAnnot

Title

Exons annotation.

Arguments

none

Details

Get gene annotation from Ensembl online database and processes it to remove duplicate entries and normalize fields.

Value

data.frame

exons annotation table.

mapProbesetsGenes

Title

Establish an association between probesets and genes.

Required packages

RMySQL, plyr

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

Details

Find where each probeset has probes which match on genome overlap a gene area and, in those cases, establish a probeset-gene link. It is calculated a frequency that is the number of times a probeset matches a gene over all the matches of that probeset.

Value

data.frame

A table with probesets-genes mapping.

probesetsGenesTable

Title

Create and fill probesets-genes mapping table.

Required packages

RMySQL

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

pbstsGenesMap (data.frame)

Table with probesets-genes mapping.

Details

Create and fill table '<vendor>_probesets_genes' with probesets-genes mapping.

Value

none

mapProbesGenes

Title

Establish an association between probes and genes.

Required packages

RMySQL

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

Details

Find where each probe matches on genome overlap a gene area and, in that cases, save that link on the database.

Value

none

createProbesGenesTable

Title

Create table for probes-genes mapping.

Required packages

RMySQL

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

Details

Create a static database table for probes-genes mapping data, with the name '<vendor>_probes_genes'.

Value

none

createGenesAnnotTable

Title

Create genes annotation table.

Required packages

RMySQL

Arguments

genesAnnot (data.frame)

Table with annotation data.

Details

Creates the table "gnm_genes_annotation" in the database and fills it with data from genesAnnot data frame.

Value

none

genesAnnot

Title

Genes annotation.

Arguments

annotFilePath (character)

Path to file where is stored the annotation information.

Details

Get gene annotation from file, extracting only the desired fields.

Value

data.frame

Genes annotation table.

probesMatches

Title

Probes sequences alignment with human genome.

Required packages

RMySQL, Biostrings, BSgenome.Hsapiens.UCSC.hg19

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

Details

Align probes sequences with the genome and inserts on <vendor>_probes_matches DB table the info probe_id, chromosome, start, end, strand. Table has to be already created.

Value

none

createProbesMatchesTable

Title

Create table for probes matches.

Required packages

RMySQL

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

Details

Create a static database table for probes matches data, with the name <vendor>_probes_matches.

Value

none

insertSeqs

Title

Insert probes sequences information in database.

Required packages

RMySQL

Arguments

gplID (character)

Platform ID.

verbose (logical)

Whether or not to show information about algorithm execution.

Details

Get Illumina probe sequences information directly from a GEO platform.

Value

data.frame

A table with one column for probesets ids and other for DNA sequences.

getIlluProbesSequences

Title

Get Illumina probes sequences from GEO platform.

Required packages

GEOquery

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

probeSequences (data.frame)

A table with one column for probesets ids and other for DNA sequences.

Details

Insert probe sequences information in the database. The table is automatically named <vendor>_probes_sequences and it is created a primary key attribute, probe_id, defined as auto increment.

Value

none

getProbesSequences

Title

Get probes sequences from platform files.

Arguments

vendor (character)

A microarrays manufacturer abbreviation (affy or agil).

verbose (boolean)

Whether or not to show information about algorithm execution.

Details

Get probe sequences information from pre-downloaded files with extension 'txt' for Agilent or '.probe_tab ' for Affymetrix.

Value

data.frame

A table with one column for probesets ids and another for DNA sequences.

rebuildDatasets

Title

Rebuild expression datasets.

Required packages

reshape, reshape2, data.table

Arguments

dSetID (character)

Identification of the dataset.

vendor (character)

A microarrays manufacturer abbreviation (affy, agil or illu).

minRatio (numeric)

minimum acceptable probeset-transcript ratio.

verbose (logical)

Whether or not show information about algorithm execution.

Details

Create new transcript-based expression dataset, from a probeset-based source. Eliminates entries that have a probeset-transcript ratio less then minRatio.

Value

data.frame

Table with transcript-oriented expression data.

insertRebDatasets

Title

Insert rebuilt expression datasets.

Required packages

RMySQL

Arguments

dSetID (character)

Identification of the dataset.

dSetData (data.frame)

Table with transcript-oriented expression data.

Details

Create a new table on 'hmg_annot_express' DB, with the name <dSetID>_hexp and fills it with transcript-oriented expression data.

Value

none

Data Joining

joinRaw

Title

Join unaltered datasets.

Arguments

dSetList (data.frame list)

List of the datasets to join.

Details

Simple join of datasets with the same row identifiers.

Value

data.frame

Table with the datasets concatenated by columns.

joinNorm

Title

Join normalized datasets.

Arguments

dSetList (data.frame list)

List of the datasets to join.

Details

Applies a normalization based on mean annulment and standard deviation correction before join datasets.

Value

data.frame

Table with the datasets concatenated by columns.

joinLinear

Title

Join datasets using linear model.

Arguments

dSetList (data.frame list)

List of the datasets to join.

Details

Finds the reference features with lower inter-class variation and use them to correct batch effect with linear model between datasets before join datasets.

Value

data.frame

Table with the datasets concatenated by columns.

joinLinearLog2

Title

Join log_2 datasets using linear model.

Arguments

dSetList (data.frame list)

List of the datasets to join.

Details

Finds the reference features with lower inter-class variation and use them to correct batch effect with linear model between datasets before join datasets. It is applied directly to log_2 transformed datasets.

Value

data.frame

Table with the datasets concatenated by columns.

joinGradeRef

Title

Join grade referenced datasets.

Arguments

dSetList (data.frame list)

List of the datasets to join.

Details

Uses a class as reference between the datasets. The other class values are divided by the reference class to achieve a comparable scale between datasets. It is only possible to apply this function where all the datasets to integrate have the reference class.

Value

data.frame

Table with the datasets concatenated by columns.

joinGradeRefLog2

Title

Join grade referenced datasets.

Arguments

dSetList (data.frame list)

List of the datasets to join.

Details

Uses a class as reference between the datasets. The other class values are divided by the reference class to achieve a comparable scale between datasets. It is only possible to apply this function where all the datasets to integrate have the reference class. It is applied directly to $\log_2$ transformed datasets.

Value

data.frame

Table with the datasets concatenated by columns.

Feature Selection

getRefs

Title

Get reference features.

Arguments

rDset (data.frame)

A data frame where first column is the dependent variable, with numeric classifications for each sample, and the other columns are input variables for the algorithm.

Details

Use mANOVA to find the features that have the lowest inter class variation when compared with inner class variation and global variation.

Value

character vector

Vector with reference features id.

getBestPredictorLDA

Title

Get best predictor set of features with Linear Discriminant Analysis.

Required packages

MASS

Arguments

data (data.frame)

A data frame where first column is the dependent variable, with numeric classifications for each sample, and the other columns are input variables for the algorithm.

maxNFeat (numeric)

Maximum number of features to use as algorithm inputs. Default to 20.

maxKeepCombs(numeric)

Maximum number of features sets combinations to keep between iterations.

verbose (logical)

Whether or not to show information about algorithm execution.

Details

Wrapper method to test the predictive strength of a set features when used the mentioned machine learning algorithm. The iterations start with sets of 1 feature and goes to a maximum of maxNFeat sets. From one iteration to the next it is only kept the maxKeepCombs sets that presents best results.

Value

character vector

Vector with the features IDs set that present best classification results, when tested with Leave One Out Cross Validation.

getBestPredictorKnn

Title

Get best predictor set of features with K Nearest Neighbors.

Required packages

class

Arguments

data (data.frame)

A data frame where first column is the dependent variable, with numeric classifications for each sample, and the other columns are input variables for the algorithm.

maxNFeat (numeric)

Maximum number of features to use as algorithm inputs. Default to 20.

maxKeepCombs(numeric)

Maximum number of features sets combinations to keep between iterations.

verbose (logical)

Whether or not to show information about algorithm execution.

Details

Wrapper method to test the predictive strength of a set features when used the mentioned machine learning algorithm. The iterations start with sets of 1 feature and goes to a maximum of maxNFeat sets. From one iteration to the next it is only kept the maxKeepCombs sets that presents best results.

Value

character vector

Vector with the features IDs set that present best classification results, when tested with Leave One Out Cross Validation.

getBestPredictorSVM

Title

Get best predictor set of features with Support Vector Machines.

Required packages

e1071

Arguments

data (data.frame)

A data frame where first column is the dependent variable, with numeric classifications for each sample, and the other columns are input variables for the algorithm.

maxNFeat (numeric)

Maximum number of features to use as algorithm inputs. Default to 20.

maxKeepCombs(numeric)

Maximum number of features sets combinations to keep between iterations.

verbose (logical)

Whether or not to show information about algorithm execution.

Details

Wrapper method to test the predictive strength of a set features when used the mentioned machine learning algorithm. The iterations start with sets of 1 feature and goes to a maximum of maxNFeat sets. From one iteration to the next it is only kept the maxKeepCombs sets that presents best results.

Value

character vector

Vector with the features IDs set that present best classification results, when tested with Leave One Out Cross Validation.

getBestPredictorOrdLogReg

Title

Get best predictor set of features with Ordered Logistic Regression.

Required packages

MASS

Arguments

data (data.frame)

A data frame where first column is the dependent variable, with numeric classifications for each sample, and the other columns are input variables for the algorithm.

maxNFeat (numeric)

Maximum number of features to use as algorithm inputs. Default to 20.

maxKeepCombs(numeric)

Maximum number of features sets combinations to keep between iterations.

verbose (logical)

Whether or not to show information about algorithm execution.

Details

Wrapper method to test the predictive strength of a set features when used the mentioned machine learning algorithm. The iterations start with sets of 1 feature and go to a maximum of maxNFeat sets. From one iteration to the next it is only kept the maxKeepCombs sets that presents best results.

Value

character vector

Vector with the features IDs set that present best classification results, when tested with Leave One Out Cross Validation.

getNonCollinear

Title

Extract a subset of non-collinear variables.

Arguments

data (numeric matrix or data.frame)

A table with the variables, in columns, to be compared for collinearity.

corThreshold (numeric)

Maximum threshold for correlation.

Details

The variables on data are compared ones to each other. If a variable is found that presents a correlation with other variable, greater then corThreshold, is discarded.

Value

character vector

Vector with non-collinear variables id.

eBayesFeatureFilter

Title

Get features with the best differential expression results.

Required packages

Biobase, limma

Arguments

dVar (numeric vector)

Reference variable with classifications.

data (numeric matrix or data.frame)

A table with the variables, in columns, to be contrasted by classes on reference.

maxPVal (numeric)

Maximum threshold for eBays result p-value. Default to NULL.

maxFeatures (numeric)

Maximum number of informative variables to return. Default to NULL.

Details

Calculate eBayes for features over all possible contrasts of the classes present on dVar. Return features ordered by crescent p-value for differential expression.

Value

numeric vector

Vector with p-values as values and feature names as indexes.

getAbsCorrelated

Title

Get variables correlated to a given reference.

Arguments

dVar (numeric vector)

Reference variable with classifications.

rData (numeric matrix or data.frame)

A table with the variables, in columns, to be compared with the reference.

minCor (numeric)

Minimum threshold for correlation. Default to NULL.

maxFeatures (numeric)

Maximum number of correlated variables to return. Default to NULL.

Details

The variables on rData are compared to dVar and are ordered by absolute Pearson correlation coefficient. If a minCor is given, only returns the variables with correlation greater than the defined value. If a maxFeatures is given, only return that number of the most correlated features.

Value

numeric vector

Vector with correlation coefficient as values and feature names as indexes.

getDfBestAnova

Title

Get features with best ANOVA results.

Arguments

classifications (numeric vector)

Reference variable with classifications.

dataset (numeric matrix or data.frame)

A table with the variables, in columns, to be tested over reference.

nFeatures (numeric)

Maximum number of features to return. Default to NULL.

Details

The variables on the dataset are ANOVA tested for the classes on 'classifications' and are ordered by crescent p-value. If nFeatures is given, only returns that number of features from the ones that present best p-value.

Value

numeric vector

Vector with p-value for F-test ANOVA as values and feature names as indexes.