Assessment of Supervised Classification Methods for the Analysis of RNA-seq Data

Mustafa Abu El Qumsan

Aix-Marseille Université
Faculté des sciences de Luminy

ECOLE DOCTORALE DES SCIENCES DE LA VIE ET DE LA SANTE
PhD THESIS in BIOINFORMATCS

Assessment of Supervised Classification Methods for the Analysis of RNA-seq Data

Presented by: **Mustafa AbuElQumsan**

Director: Prof. Jacques van Helden
Co-supervisor: Prof. Badih Ghattas

Presented and defended the 20th of December 2018
Revised version – 14 of January 2019

Jury Members :

Rapporteur	Prof. Gaëlle Lelandais, Université Paris-Sud, France
Rapporteur	Dr. Marie-Agnès Dillies, Institut Pasteur, Paris, France
Examinateur	Dr. Christine Brun, TAGC, CNRS
Examinateur	Dr. Denis Puthier, TAGC, Aix-Marseille Université
Examinateur	Dr. Pascal Barbry, CNRS, Université de Nice

TABLE OF CONTENTS

PREFACE

RNA-sequencing (RNA-seq), which is one of the first applications of the next generation sequencing (NGS) technology, has become widely used and replaced the previous microarrays in the analysis of transcriptome. Many statistical methods were developed and applied for the statistical analysis of gene expression with both mentioned technologies. However, microarrays and RNA-seq deserve separate methodological assessment, due to the fundamental difference: array-based technology quantifies levels of expression by continuous distributions, whereas RNA-seq datasets does it by counts of reads. There is currently a need for powerful advanced statistical methods to extract the valuable information from fast developing sequencing technology and limited works have been done into exploit the machine learning methods on expression analysis of RNA-seq data.

Functional supervised and unsupervised methods are important methods for performing sample classification based on their expression profile. In this dissertation, we assessed the effectiveness of supervised and unsupervised approaches to classify biological samples based on RNA-seq data, based on the analysis the seven carefully chosen study cases from genomic experiments with human samples downloaded from recount2 repository.

Due to the complexity and size of genomic data, the choice of machine learning approach and parameters that return optimal classification results with RNA-seq data is a real concern for current biologists, we tested here several commonly used supervised classification approaches and assessed the impact of pre-processing procedures.

Another common issue in the analysis of classification with high-throughput data is exploit feature selection approaches, which requires to study their impact on the accuracy of classifiers in order to improve the methodological choices. Whereas many feature selection methods have been developed for various applications of machine learning, limited works has been done on RNA-seq data. In the current work, we examine some feature selection methods to test their impact on the classifiers and clustering of RNA-seq data.

We developed an R package called *RNAseqMVA*, which is specifically designed to apply various multivariate analysis (MVA) approaches on RNA-seq datasets, and to assess the performance of classifiers with two various pre-processing and feature selection approaches. *RNAseqMVA* is available as a GitHub repository.

ACKNOWLEDGEMENTS

First and foremost, I acknowledge jury members Prof. Gaëlle Lelandais, Dr. Marie-Agnèse Dillies, and Dr. Christine Brun, CNRS, Dr. Denis Puthier, TAGC, and Dr. Pascal Barbry, CNRS who booking their time for carious reading, proof and evaluating my dissertation.

I am very thankful for their valuable input and candid advice that indeed asses to enlarging my vision of science.

I would like to give my unlimited thanks to my supervisor **JACQUES VAN HELDEN** who always guides and tip me along the way. I am too profoundly grateful for the chance that I've had conducted this research with him. The guidance, encouragement and great discussions regularly produced some of the most important ideas and results of such project. His impact on a science will be felt long after I leave his lab and extend far beyond my academic pursuits.

Unlimited thanks for **Badih Ghattas** Co-supervisor for me. He was too resourceful man in recent and novel machine learning science, I exploit such occasion to thanks him a lot for his valuable advice and guidance me.

Thanks to whole members of the **TAGC** for hosting me by creating an environment where I can concerning on doing great science, the job with you have been extremely amazing, stimulating and education. I have been so lucky being welcomed in to the lives with several people who I consider as second family through the last three years. I am much grateful to spend the most wonderful time of my life with you.

Last but not least, I would like to give my special thanks to all my dear family (the passion is Rewaa, my heart Khaled and Ledia) in France for their helps, amity and kindness. All of you have been irreplaceable part of my experience here. The out of work pressure with you helps me to balance my life and enjoying here, my life here was much colorful and meaningful by your sharing with me to stay in France to obtain Doctoral degree.

Above all, I don't know how to thank *my daddy* Khaled and *my compassionate heart mom* Buthina to harness their whole life to affording joyful life and they pray every day to success in my life and to become happy in my life, all thank for my parents and private thanks for my ** lovely brothers** Eng.Mohammed, Mr.Hamza and Eng.Ahmed and ** lovely sisters** Mrs.Lina and Eng.Chaima they didn't save any effort to help and support me, they always puts theirs hope to me, that's inspired me every day. I am proud of them in such occasion and i dedicated my

dissertation for whole my family wherein this is the first work in whole Palestine in Bioinformatics realm.

Finally, heartfelt gratitude go to **Islamic Development Bank (IDB)-Jeddah** for their full sponsorship of my work to obtain my Doctoral degree from France, without them, I wouldn't have been able to pursuit my high education for doctoral in France. Continuous thanks for the IsDB for their generosity and support me especially and for Palestine in whole.

Mustafa ABUELQUMSAN

Marseille, Novembre 2018

RESUME

Depuis une décennie, l'avènement des technologies de séquençage massivement parallèle (Next Generation Sequencing, NGS) a révolutionné la façon de mener les études génomiques. Une application particulièrement importante et largement répandue du NGS est l'étude du transcriptome par séquençage de l'ADNc obtenu à partir de l'ARN d'un échantillon (RNA-seq).

La technologie RNA-seq présente un grand nombre d'avantages par rapport aux précédentes (notamment les biopuces) : élargissement de la plage dynamique de mesure, accroissement de la précision, débit élevé, découverte de nouvelles formes d'épissage, etc. Conséquemment, le RNA-seq a progressivement remplacé les approches de biopuces pour devenir la principale technologie d'analyse du transcriptome. Les études NGS produisent d'énormes quantités de données, qui appellent au développement de méthodes d'analyse multidimensionnelle efficients, qui prennent en compte la nature particulière des données (comptages discrets, étendue dynamique énorme, présence de valeurs aberrantes, …). Dans cette thèse, nous nous focalisons sur l'utilisation de méthodes d'apprentissage automatique pour assigner des échantillons à des classes sur base de leurs profils d'expression RNA-seq.

Tout d'abord, nous dressons une revue de l'état de l'art pour la génomique, et des méthodes statistiques qui ont été appliquées aux méthodes NGS, afin de tirer les leçons des derniers développements méthodologiques et d'évaluer l'apport de notre recherche par rapport aux derniers développements en analyse multidimensionnelle des données NGS.

Nous effectuons ensuite une évaluation comparative des méthodes de classification supervisées sur base de données téléchargées de la base de données recount2, qui contient à peu près 2000 expériences de RNA-seq. Dans cette base de données, nous avons sélectionné 7 cas d'étude représentatifs d'études RNA-seq typiques, avec différents types de catégories (classes) : maladies (types de cancers, leucémies, psoriasis), ou types cellulaires (cellules nerveuses). Nous avons évalue l'impact du pré-traitement des données sur les méthodes de classification supervisée: procédures de filtrage (mise à l'écart de gènes et/ou échantillons non fiables), normalisation, transformation en composantes principales (ACP). Nous avons également étudié l'impact de la sélection de variables afin de réduire la sur-dimensionnalité de l'espace des variables, et d'identifier le sous-ensemble de gènes ou composantes qui optimisent la précision des classifications. Cette sélection repose sur un tri préalable des variables basé soit sur l'analyse différentielle d'expression, soit sur l'importance des variables calculée lors d'un premier cycle de classification avec Random Forest.

Durant toute cette étude, nous avons prêté une attention particulière aux métadonnées, et nous avons exploré la structure des jeux de données, afin d'interpréter le comportement de chaque méthode (Support Vector Machines, Random Forest, K Nearest Neighbouts) à la lumière des spécificités de chaque cas d'étude : nombre d'échantillons, de classes, distribution des comptages bruts, RNA-seq sur échantillons entiers (« bulk ») ou cellules isolées (« single-cell »).

ABSTRACT

In recent years, the advent of next-generation sequencing (NGS) technology has been revolutionizing the way genomic studies are processed. An important and widely used application of NGS technology is the study of transcriptome through sequencing of cDNA obtained from RNA (RNA-seq). Compared with previous technologies like microarrays, RNA-seq data have many advantages, such as dynamic and wider ranges of measurements, increased precision, higher throughput, discovery of novel RNA species and splice forms, etc. Thence, RNA-seq has been became suitable alternative for the microarray approach as the main platform for transcriptome studies. NGS technologies produce huge amounts of data, which urges the development of effective multivariate analysis methods adapted to the particular nature of the data (discrete counts, huge dynamic range, outliers, …). In this dissertation, we focus on the use of machine learning methods to perform supervised classification to assign samples to groups based on their RNA-seq gene expression profiles.

First, we briefly revise the state-of-art for the genomics and the statistical methods to treat NGS data, in order to draw lessons from the latest developments in analysis the NGS data and to evaluate what our research will provide to the latest scientific developments in the scope of multivariate analysis for the NGS data.

We perform a comparative assessment of three supervised classification methods (Support Vector Machines, K nearest neighbors, Random Forests), based on published data downloaded from the recount2 warehouse, which contains around 2000 RNA-seq experiments. From this database, we selected seven study cases that are representative for typical RNA-seq studies with different types of biological categories (classes), including diseases (cancer types, leukemia, psoriasis) or cell types (nervous cells). We assessed the impact of pre-processing on classifiers: filtering procedures (discarding unsuited genes and/or samples), normalization, PCA transformation. We also studied the impact of feature selection to circumvent the problem of over-dimensionality of the feature space by finding out a subset of genes or principal components that optimizes the accuracy of classifiers. The feature selection relied on variable ordering based on either differential expression analysis, or on variable importance returned by a Random Forest classifier.

We pay a particular attention to the metadata and we explore the structure of the datasets, in order to interpret the behavior of each tested classifier in light of the specificities of each study

case (number of samples, number of classes, distribution of the count values, bulk or single-cell RNA-seq, …).

Keywords

Bioinformatics, Biostatistics, Next Generation Sequencing, RNA-seq, Supervised classification

Bioinformatique, Biostatistique, Séquençage Massivement Parallèle, RNA-seq, Classification supervisée

Short abstract for wide audience

Since a decade, Next Generation Sequencing (NGS) technologies enabled to characterize genomic sequences at an unprecedented pace. Many studies focused of human genetic diversity (inter-individual variations in genomic DNA sequences) and on transcriptome (the part of genome transcribed into ribonucleic acid). Indeed, different tissues of our body express different genes at different moments, enabling cell differentiation and functional response to environmental changes. Since many diseases affect gene expression, transcriptome profiles can be used for medical purposes (diagnostic and prognostic). A wide variety of advanced statistical and machine learning methods have been proposed to address the general problem of classifying individuals according to multiple variables (e.g. transcription level of thousands of genes in hundreds of samples). During my thesis, I led a comparative assessment of machine learning methods and their parameters, to optimize the accuracy of sample classification based on RNA-seq transcriptome profiles.

Résumé de thèse vulgarisé pour le grand public en français

Les technologies « *Next Generation Sequencing* » (NGS), qui permettent de caractériser les séquences génomiques à un rythme sans précédent, sont utilisées pour caractériser la diversité génétique humaine et le transcriptome (partie du génome transcrite en acides ribonucléiques). Les variations du niveau d'expression des gènes selon les organes et circonstances, sous-tendent la différentiation cellulaire et la réponse aux changements d'environnement. Comme les maladies affectent souvent l'expression génique, les profils transcriptomiques peuvent servir des fins médicales (diagnostic, pronostic). Différentes méthodes d'apprentissage artificiel ont été proposées pour classer des individus sur base de données multidimensionnelles (par exemple, niveau d'expression de tous les gènes dans des d'échantillons). Pendant ma thèse, j'ai évalué des méthodes de « machine learning » afin d'optimiser la précision de la classification d'échantillons sur base de profils transcriptomiques de type RNA-seq.

ABBREVIATIONS

Acronyms	Meaning
AUC	Area under the ROC curve
bp	Basepair
CAGE	cap analysis gene expression
cDNA	Complementary DNA
DE	differential expressions
EST	expressed sequence tag
GEO	Gene Expression Omnibus
GWAS	genome-wide association study
HGP	Human Genome Project
indels	Small insertions and deletions
Kbp	kilo basepair
KNN	k-near neighbor
lncRNAs	long noncoding RNAs
LRT	likelihood ratio test
Mbp	megabase pair
Med	Median
MER	Misclassification error rate
miRNAs	micro RNAs
mRNA	messenger RNA
MSE	Mean Square Error
NGS	Next-generation sequencing
OOP	Object-Oriented Programming
PCA	Principal component analysis.
PS	Prototype Selection
Q	Quantile
qPCR	quantitative polymerase chain reaction
RBF	the redial basis function
RF	Random Forest
RLE	Relative Log Expression
ROC	Receiver operating curve
RPKM	Read per Kilobase per Million
SAGE	serial analysis of gene expression
SMS	Single Molecule Sequencing
snoRNAs	small nucleolar RNAs
SNPs	single nucleotide polymorphisms
snRNAs	small nuclear RNA
SNVs	single nucleotide variants
SVM	Support vector machine
TC	Total Count
TMM	Trimmed mean of M-Values
UML	Unified Modeling Language
UQ	Upper Quartile
VIMP	variable importance

CHAPTER 1: INTRODUCTION

Genomics and the next-generation sequencing revolution

The genomics era

Since the time DNA was revealed as the code to all biological life, man has sought to disclose its secrets. If the genetic sequence could be sequenced or *read*, the basis of life itself may be discovered. Although this idea might not be fully true, the recent progresses of sequencing technologies have certainly revolutionized the scope of the biological research, and influenced the way biologists try to address the complexity of life by analyzing biological phenomena at the scale of whole genomes.

The original sequencing methodology, known as Sanger chemistry, uses specially labeled nucleotides to read through a DNA mold during DNA synthesis. This sequencing technology needs a certain primer to begin reading at a certain location through the DNA shape, and records the different labels for each nucleotide within the sequence. After a series of technical inventions, the Sanger method has reached the amplitude to read through 1,000–1,200 base pairs (bp); however, it still cannot exceed 2 kilo base pairs (Kbp) behind a certain sequencing primer. To sequence longer sections of DNA, a new method called shotgun sequencing was improved during the international scientific research project that has been aimed determining the sequence of nucleotide base pair that make up human DNA that is called the Human Genome Project (HGP). In this approach, genomic DNA is enzymatically or mechanically cut into smaller fragments and cloned into sequencing vectors in which the cloned DNA fragments can be sequenced individually. The full sequence of a long DNA segment can then be obtained through alignment and reassembly of the sequence fragments based on partial sequence overlaps. Shotgun sequencing was of great benefit to HGP, and it made the sequencing of the entire human genome possible.

In 2007, several companies proposed novel technologies that enable the main principle of massive parallel sequencing to be used in what is called **next-generation sequencing** (NGS), which is adapted from shotgun sequencing (Margulies et al., 2005a); (Venter et al., 2003). Modern NGS methods read the DNA molds technically over the whole genome. This is done by cutting the whole genome into small pieces and then linking these small pieces of DNA to designated adapters for a random read during DNA synthesis (sequencing by synthesis). The DNA

synthesis/reading process is done in parallel for millions of DNA fragments simultaneously; therefore, NGS technology is also called *massively parallel sequencing*.

In the first applications of NGS, the read length (the actual number of contiguous sequenced bases) for NGS was much shorter than that achieved by Sanger sequencing. At the advent of NGS, sequencing provides reads that were typically 26–25 bp, which is why the sequencing results are identified as *short reads*. These short reads were a major limitation at the beginning of NGS technology; however, developing NGS technologies, such as **single-molecule sequencing** (SMS), may exceed Sanger methodology and have the potential to read several continuous Kbps (Zhang et al., 2011). Table 1 shows a summary of the benefits of each method.

As next-generation technologies generate short reads, coverage is a very important issue. Coverage is defined as the number of short reads that overlap one another within a certain genomic area (Zhang et al., 2011). An adequate coverage is critical for the accurate assembly of the genomic sequence; in addition, the coverage is crucial for applications other than genome sequencing. For instance, in ChIP-seq, discriminating peaks from the background (noise) is important. For transcriptomic RNA-seq, increasing the sensitivity (detection of poorly expressed genes) and obtaining a reliable quantification of the RNA concentration for each gene are crucial, and so is minimizing the impact of the fluctuations of small numbers. A sufficient coverage is also important to compensate for the fact that many short read sequences cannot be interpreted or *mapped* to any reference DNA or be carefully assembled.

Short-read sequences can be matched against a reference genome, a process called read mapping (the results are commonly called mapped reads). We could define the sequence coverage as the average number of reads that overlap each nucleotide in a given region (*local coverage*) or on the entire genome (*genome coverage*). The possible ambiguities of read mapping because of repetitive regions still need to be considered, through mapping short reads against a reference genome that is classically the first step of many next-generation sequencing data analysis.

Table 1 Brief comparison between Sanger, NGS, and SMS

Technology	Pros	Cons
Sanger Sequencing	High precision error rate: 0.001%–1% Long reads	Low throughput High cost
Next-Generation Sequencing (NGS)	Low cost High throughput Decent precision error rate: 0.46%–2.4%	Short reads (which will result in low precision when doing sequence assembly)
Single-Molecule Sequencing (SMS)	Long reads High throughput	Low precision error rate: 11%–14%

NGS is a rapidly evolving technology that is changing on an almost daily basis.

Previously, DNA sequencing was performed almost exclusively with the Sanger method, which has excellent precision and a sensible read length but has very low throughput. Sanger sequencing was used to obtain the first draft sequence of the human genome in 2001 (Consortium, 2001) and the first individual human diploid sequence (J. Craig Venter) in 2007 (Levy et al., 2007). Immediately thereafter, the second complete individual genome (James D. Watson) was sequenced using next-generation technology, which marked the first human genome sequenced with new NGS technology (Wheeler et al., 2008). Since then, several additional diploid human genomes have been sequenced with NGS by utilizing a variety of related approaches to quickly sequence genomes with varying degrees of coverage (Wang et al., 2008) and (Metzker, 2010). A common mechanism for NGS is to use DNA synthesis or ligation process to read through many different DNA shapes in parallel (Seo et al., 2017).

Genomics involves the systematic study at a whole genome scale of genetic contributions to different aspects of an organism of interest (e.g., human, bacteria, yeast, drosophila, plants). The progress in our understanding of many essential biological phenomena has accelerated dramatically over the last two decades, driven by the evolution of genomic technologies. Under other applications, these new genomic technologies have revolutionized our knowledge of many genes or genomic areas involved in the pathogenesis of human diseases (Novelli et al., 2010). Developments in high-throughput genomic technologies, such as microarray and NGS technologies, have resulted in massive accomplishments on genetic linkage, association studies, DNA copy number, and gene expression analysis. The progress of genomics will undoubtedly lead to the birth of genetic medicine, which will, in turn, result in significant developments and improvements in human health (Gonzalez-Angulo et al., 2010). Candidate gene methods were initially used in genomic studies, with a focus on the genes known to be included in well-defined molecular pathways for targeted human conditions through linkage and association studies. Through nominate-gene studies, certain genetic variants among numerous genetic loci have been successfully identified for their important contributions to specific human diseases. After the completion of the HGP, a new approach, genome-wide association study (GWAS), has been applied to genomics. GWAS is highly effective in specifying the genetics factors related to disease or other human traits by allowing deduction over the entire length of the genome through acquisition of direct information on a relatively small number of loci. But NGS relies on the direct procuration of information from all adjustable loci (Alonso, 2015). The wider applicability of GWAS over full genome sequencing is based on the equilibrium between the lower costs of GWAS and main goal for the RNA sequencing experiments, allowing for the analyses of larger cohorts.

Because many of the genetic variants that contribute to many human conditions are still unknown, unbiased whole genome sequencing will help identify these genetic variants, involving single nucleotide variants or single nucleotide polymorphisms, small insertions and deletions (indels, 1–1,000 bp), and structural and genomic variants (> 1,000 bp) (Daly, 2010).

The quantity of short-read sequences produced by NGS is increasing at exponential rates as a result of the many NGS approaches recently developed to allow DNA sequencing. In less than a decade, current NGS platforms have increased the throughput of sequencing, and the massive reduction in cost has transformed NGS into a vastly used genomic technology. Different NGS instruments generate different base read lengths, error rates, and error profiles relative to Sanger sequencing data and to one another. NGS technologies have increased the speed and throughput capacities of DNA sequencing and, as a result, dramatically reduced overall sequencing costs (Metzker, 2010)(Tucker et al., 2009); (Ng et al., 2009).

Since the release of the first available system (GS20 from 454 Life Sciences) with a throughput of 20 megabase pairs (Mbp) per run (Margulies et al., 2005b), NGS technology has significantly developed. The current Illumina HiSeq X system can produce 1.8 terabase pairs (Tbp) of sequencing data per run, representing nearly a 100,00-fold increase within a 10-year period. During this relatively brief period, several NGS systems, such as HeliScope from Helicos BioSciences (Thompson and Steinmann, 2010) and 454 GS FLX from Roche, have been turned off. The new NGS systems include single-molecule sequencers (Eid et al., 2009), which can provide high read lengths and facilitate the resolution of DNA modifications. The shares occupied by the industry are sequence-by-synthesis (Bentley et al., 2008) systems from Illumina, which boast a wide range of applications, relative ease of use, multiple levels of throughput, a flexible configuration, and a relatively low sequencing cost.

Application of NGS technologies in genomics

NGS has numerous advantages, such as its cost-effectiveness, unprecedented sequencing speed, and high resolution and accuracy in genomic analysis, which have improved biological biomedical research. The majority of these high-throughput sequencing technologies have been fully implemented in different of ways, such as target sequencing, whole genome sequencing, chromatin immunoprecipitation sequencing, gene expression profiling, and small RNA sequencing. All these revolutions in high-throughput sequencing have led to the massive amount of data generated by NGS, which represents a great challenge for bioinformatics.

The available applications of NGS technologies include whole genome sequencing, *de novo* assembly sequencing, resequencing, and transcriptome sequencing at the DNA or RNA level. For instance, *de novo* assembly sequencing assembles the genome of a particular organism without a reference genome sequence (Li et al., 2010), resulting in a better understanding of the genomic level and may help in predicting genes, protein coding regions, and pathways. Moreover, resequencing the organism with a known genome can assist in understanding the relationship between genotype and phenotype and specify the difference among reference sequences (Vallender, 2011); (Voelkerding et al., 2010). Moreover, NGS technologies have been used to analyze small RNAs (Friedländer et al., 2008) and (Zywicki et al., 2012), including the identification of differentially expressed micro RNAs (miRNAs), prediction of novel miRNAs, and annotation of other small non-coding RNAs. Currently, many companies are implementing different NGS technologies; some of these companies are Illumina (http://www.Illumina.com), Roche (http://www.454.com), ABI Life Technologies (http://www.lifetechnologies.com), Helicons BioSciences (http://www.helicosbio.com), Pacific Bioscience (http://www.pacificbiosciences.com), and Oxford Nanopore (http://www.nanoporetech)

Current technologies for NGS

Short-read coverage must enable the characterization of a complete sequence and assemble it with precision to guarantee the correct specification of genetic variants. Currently, at least 30x coverage is recommended in whole genome scans for rare genetic variants in human genomes, but this is burdensome in terms of computer resources and cost management. Even if the cost of whole genome sequencing has decreased, cost remains a major hurdle. By targeting certain regions of interest, selective DNA enrichment techniques reduce the overall cost and increase the efficiency of NGS by increasing the sequencing depth on the regions of interest (Rehman et al., 2010); (Tyler et al., 2016). However, targeted enrichment must have uniform coverage, high reproducibility, and no allele bias for any genomic area (Flowers et al., 2015). Targeted sequencing generally concentrates on all protein-coding subsequences (the functional exome), only requiring ~5% as much sequencing compared with that required for the whole human genome (Pussegoda, 2010); (Teer and Mullikin, 2010). This strategy currently reduces the overall cost for sequencing a single individual. An important consideration in the cost of such experiments is the depth of sequence coverage required to achieve a desired sensitivity and specificity of at least 25-fold nominal sequence coverage.

The most common techniques for targeted sequence enrichment are either microarray based (Igartua et al., 2010) or solution hybrid based (Bainbridge et al., 2010); (Tewhey et al., 2009). Many targeted selection technologies have been successfully applied in different NGS projects with variable success and may become the tools of choice to lower the burden of time and cost. Clarifying selective DNA enrichment techniques will considerably reduce the overall cost and speed up the discovery of genetic variants that cause rare genetic disorders. Other genetic loci for rare diseases have also been successfully identified through exome sequencing (Walsh et al., 2010); (Rios et al., 2010).

In comparison with the microarray, a recent approach to study gene expression, which was developed at the end of the last century, is RNA-seq technology; it has become a ubiquitous tool to measure a range of expression levels with less noise and high throughput, and it has an additional capability of detecting allele-specific expressions, novel promoters, and isoforms (Wang et al., 2010). For these reasons, RNA-seq is gradually substituting microarray-based approaches as the major platform in gene expression analysis. Meantime, the massive amounts of discrete data generated by NGS technology call for effective methods of statistical analysis.

RNA sequencing

RNA-seq enables digital gene expression measurement; it is a substitute of microarrays. The pattern of gene expression in cells and tissues can largely mirror their functional state. NGS-based expression profiling by RNA-seq (Marioni et al., 2008a); (Mortazavi et al., 2008) allows the comprehensive qualitative and quantitative mapping of all transcripts (Garber et al., 2011). Prior to NGS, transcriptome profiling techniques were limited in scope and accuracy, and they were not quantitatively sensitive.

The principle of RNA-seq is based on high-throughput technology. In general, a population of RNA is converted to a library of cDNA fragments with adaptors attached to one or both ends. Each molecule is sequenced in a high-throughput manner to obtain the sequence of either a single end or both ends per DNA fragment. The reads differ from 30 to 400 bp, which depends on the sequencing technology used. Three sequencing systems have been designed by companies for RNA-seq: Illumina IG, Applied Biosystems SOLiD, and Roche 454 Life Sciences. The results are either aligned to a reference genome or transcripts, or they are assembled de novo without a genomic sequence to produce a genome-scale transcription map that consists of both the transcriptional structure and the level of expression for each gene.

The transcriptome is defined as a complete set of RNA molecules produced by a given cell under given conditions (cell type, developmental stage, environment). It is fundamental for explaining the functional elements of the genome and for understanding the impact of a disease at the cellular level. Indeed, the final expression of genetic information, which depends on the interaction genetic and environmental factors, characterizes the phenotype of an organism. The transcription of a subset of genes into complementary RNA molecules defines a cell's identity and organizes the biological activities within it.

The transcriptome has a high degree of complexity and reveals multiple types of coding and non-coding RNA forms. Genetically, RNA molecules are mostly considered as simple intermediates between genes and proteins. Therefore, messenger RNA (mRNA) molecules are the typically studied RNA type because they encode proteins via the genetic code. Furthermore, in addition to the protein-coding mRNAs, there are many types of noncoding RNA (ncRNA) molecules that are functional. Most of the known ncRNAs achieved requisite cellular functions, such as ribosomal RNAs (rRNA) and transfer RNAs involved in mRNA translation, small nuclear RNAs (snRNAs) included in splicing, and small nucleolar RNAs (snoRNAs) included in the adjustment of rRNAs (Mattick and Makunin, 2006). More recently, new classes of RNA have been found, reinforcing the repertoire of ncRNAs. Another interesting class of ncRNAs is long noncoding RNAs (lncRNAs). As a functional class, LncRNAs were first described in mice during the large-scale sequencing of cDNA libraries (Team*, 2002). Many molecular functions have been revealed for IncRNAs, including chromatin remodeling, transcriptional control, and post-transcriptional processing. Despite these developments, however, most of them are not yet fully characterized (Wilusz et al., 2009) and (Bainbridge et al., 2010).

Initial gene expression studies depended on low-throughput methods. Examples are northern blots and quantitative polymerase chain reaction (qPCR), which are limited to scaling single transcripts. Throughout the last two decades, methods have been developed to enable the genome-wide quantification of gene expression, or best known as transcriptomics. The first transcriptomics studies were done using hybridization-based microarray technologies, which provide a high-throughput option at a relatively low cost (Schena et al., 1995). These methods have many limitations, such as the requirement for a priori knowledge of the sequences being investigated, the presence of problematic cross-hybridization artifacts in the analysis of highly similar sequences, and the limited ability to accurately quantify lowly expressed and very highly expressed genes (Shendure, 2008). Similar to hybridization-based methods, NGS-based methods have been

improved to clarify the transcriptome by directly identifying the transcript sequence. At first, the generation of expressed sequence tag libraries by Sanger sequencing of complementary DNA (cDNA) was used in gene expression studies, but this method was relatively low in throughput and not optimal for quantifying transcripts (Kouichi et al., 1994) and (Adams et al., 1995). To overcome these technical restrictions, tag-based methods, such as serial analysis of gene expression (SAGE) and cap analysis gene expression (CAGE), were developed for a higher throughput and a more precise quantification of expression levels. By quantifying the number of tagged sequences, which directly corresponded to the number of mRNA transcripts, these tag-based methods show a distinct advantage over the measurement of analog-style severities, such as in array-based methods (Shiraki et al., 2003). By contrast, these assays do not enable to measure the expression levels of splice isoforms and cannot be used to discover new genes. Furthermore, the overwork resulting from the cloning of sequence tags, the high cost of automated Sanger sequencing, and the requirement for a large amount of input RNA have limited its use.

The evolution of high-throughput NGS has developed transcriptomics by enabling RNA analysis through the sequencing of cDNA (Wang et al., 2009). This method, called RNA sequencing (RNA-Seq), has distinguishing traits over prior approaches and has revolutionized our understanding of the complex and dynamic nature of the transcriptome. RNA-seq provides a more detailed and quantitative view of gene expression (as is exemplified in **Figure 1**), substitutional splicing, and allele-specific expression. The development of RNA-seq workflows, from sample preparation to sequencing platforms to bioinformatic data analysis, has enabled deep profiling of the transcriptome and the opportunity to clarify different physiological and pathological conditions.

RNA-seq count data consist of tables indicating the number of sequenced fragments for each transcript. These data are modeled as emerging from random sampling events for a large number of sequences (the library size). Individual gene probabilities are small, as counts are measured for tens of thousands of reads. However, the multi-nomial model is too simple to reflect biological complexities. Indeed, it has been repeatedly shown that RNA-seq data are over-dispersed (Robinson and Oshlack, 2010a). Any careful analysis of the data, especially for differential expression analysis, should account for this over-dispersion. Additional factors, such as the length of the transcript and potential sequencing bias, are important in making an inference on the absolute expression levels.

The limitations in the use of RNA-seq are as follows:

- **Large number of genes**: The huge dimensions of RNA-seq datasets also require heavy computation in the analysis, which necessitates high computing power from both machine hardware and algorithm design.
- **Discreteness of the raw data**: RNA-seq data use counts of reads to quantify gene expressions. This is quite different from continuous data that are typically modeled by Gaussian distributions. Computing log-transformed counts can be used to partly normalize the measures of gene expressions, but methods that keep the discrete nature of count data are still preferred for differential expression. Various discrete probabilities have been proposed to model the counts, such as Poisson (Sultan et al., 2008), hypergeometric (Marioni et al., 2008a), and negative binomial (NB) distributions (Robinson and Oshlack, 2010a); (Anders and Huber, 2010a). A fundamental difficulty for the analysis of count-based data is to identify suitable theoretical distributions to model the data and take into account the properties of their distribution (in particular the variance between samples).

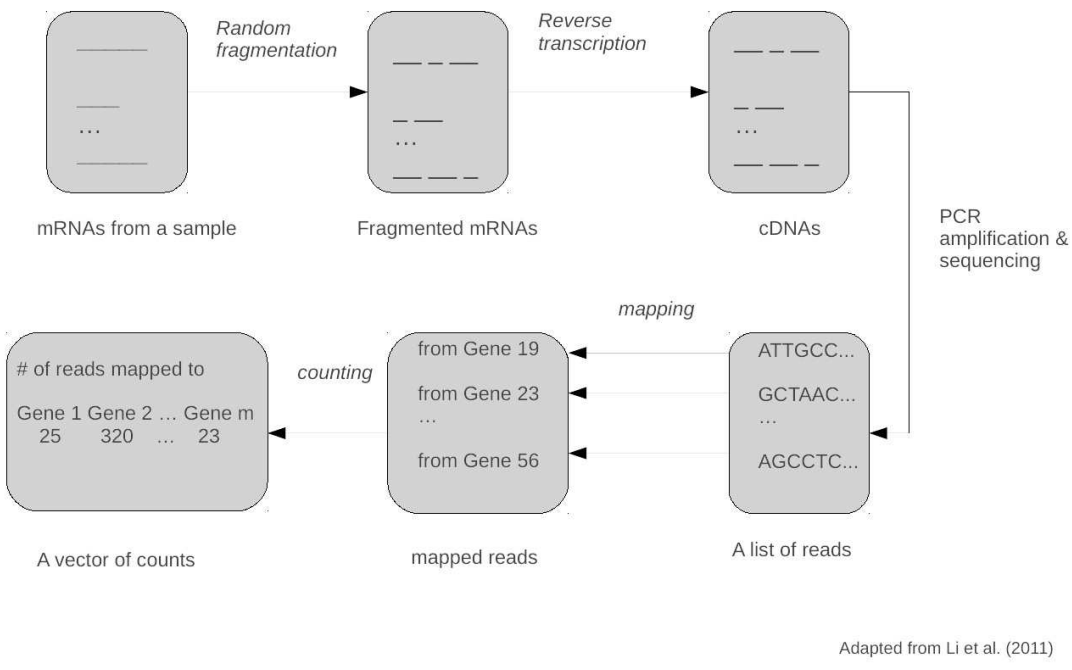


Figure 1 Pre-step of RNA sequencing
Source: Julie Aubert [Statistical Challenges in RNA-seq Data Analysis] [2012] [https://www.cnrs.fr/inee/recherche/fichiers/EPEGE/Communications/Julie_AUBERT.pdf].

Transcriptome sequencing

High-throughput NGS technologies were rapidly adopted for transcriptomics. This development addressed the many difficulties posed by hybridization-based microarrays and Sanger sequencing-based methods that were formerly utilized for scaling gene expression. An ideal RNA-seq experiment depends on isolating RNA, transforming it to cDNA, and performing the sequencing library, as shown in Figure 2. Prior to the conduct of any RNA-seq experiments, many factors should be carefully taken into account in the design of the experiment in order to ensure a balance between the quality of the results and the time and monetary investments made; some of these factors are biological and technical replicates, the depth of sequencing, and a desirable coverage across the transcriptome.

Isolation of RNA

The isolation of RNA from a biological sample is the initial procedure in transcriptome sequencing. To ensure a successful RNA-seq experiment, the RNA material should have sufficient quality in order to produce a library for sequencing. The Agilent Bioanalyzer enables the measurement of the quality of RNA by producing an RNA integrity number (RIN) between 1 and 10, where a score of 10 indicates the highest quality of samples and shows the least degradation. Notably, RIN measures are based on mammalian organisms, and certain samples with abnormal ribosomal ratios may improperly generate degraded RIN numbers. Low-quality RNA (RIN < 6) can essentially affect the sequencing results (e.g., uneven gene coverage, $3^{/}$-$5^{/}$ transcript bias). In turn, this might lead to improper biological conclusions. Consequently, high-quality RNA is fundamental for successful RNA-seq experiments. The effect of degraded RNA on the sequencing results should be accurately determined (Rudloff et al., 2010) and (Thompson et al., 2007)

Methods for library preparation

The creation of the RNA-seq library is the subsequent step in transcriptome sequencing, which occurs after RNA isolation; this process differs according to the chosen RNA species and among NGS. The creation of sequencing libraries includes isolating the desirable RNA molecules, reverse-transcribing the RNA to cDNA, fragmenting the cDNA, amplifying it by PCR, and linking sequencing adaptors. Through these main processes, as shown in **Figure 3**, there are multiple options in library preparation and experimental design that should be cautiously taken into

consideration according to specific scientific needs. The precision of the detection for certain types of RNAs is fully dependent on the quality of the library built.

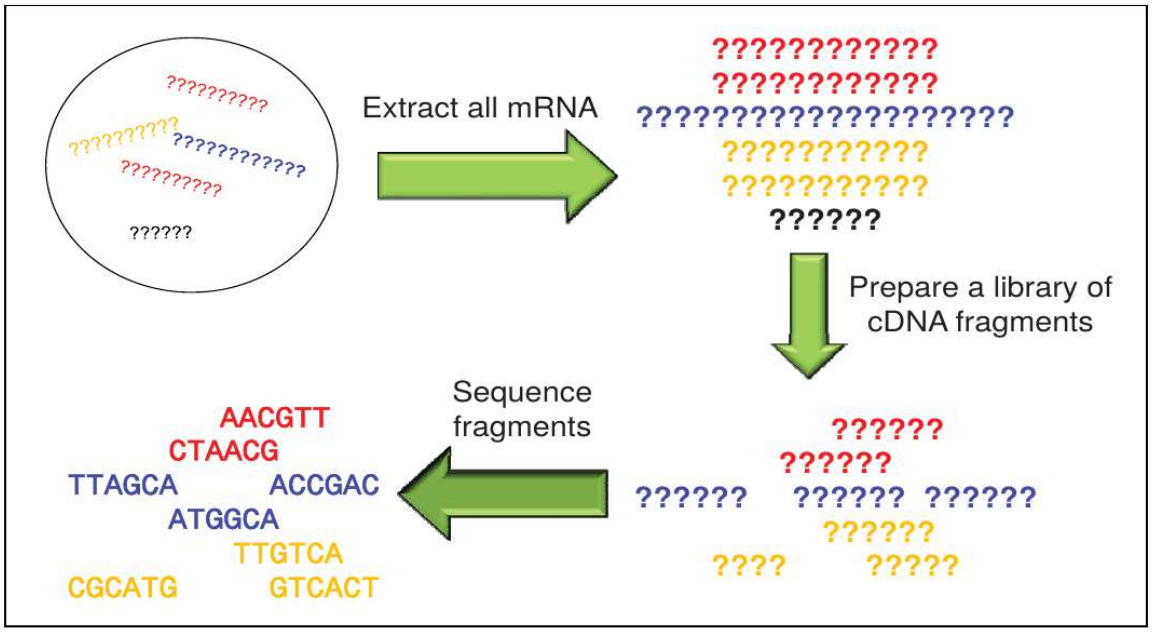


Figure 2 Sequencing steps
A sample of mRNA is transformed to a library of cDNA fragments and then sequenced with high-throughput sequencing. Source: Lecture notes from Dr. Penge Liu's Stat 416 class.

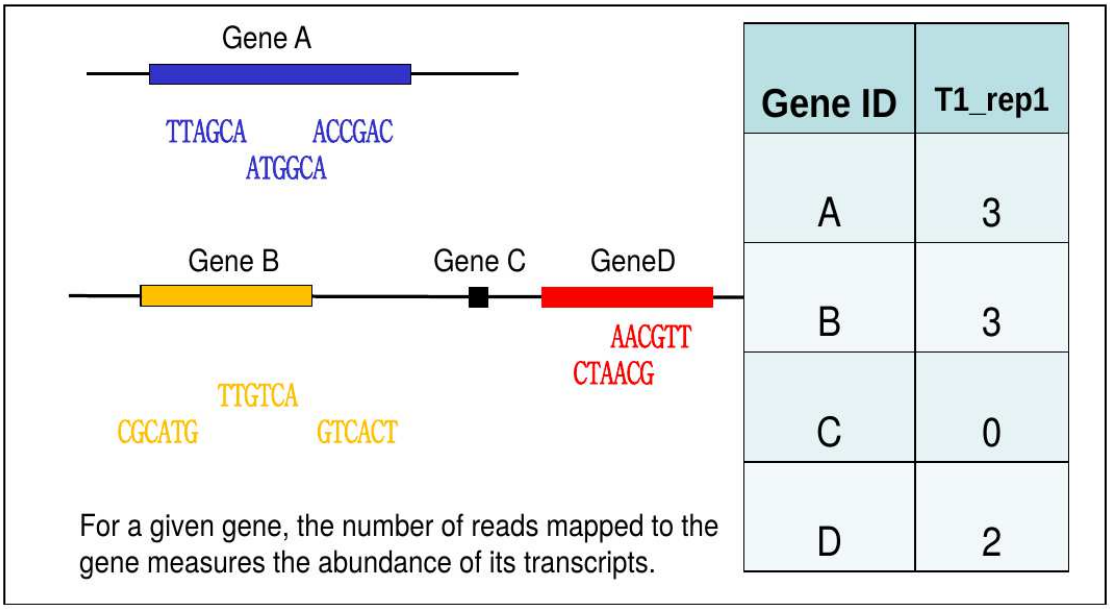


Figure 3 Mapping steps
The reads are mapped to the reference genome, and the mapped reads are counted for each gene to measure its expression level. Source: Lecture notes from Dr. Penge Liu's Stat 416 class.

Detecting Differentially Expressed Genes

Since the advent of NGS, the detection of differentially expressed genes (DEGs) has been a very important motivation for characterizing the transcriptome with RNA-seq; different specialized tools have also been developed to detect DEGs from RNA-seq counts.

Pepke et al. (Pepke et al., 2009) reported on the power of counting-based measurements (RNA-seq and ChIP-seq), resulting in the treatment of tens to hundreds of millions of reads. Utilizing deep DNA sequencing methods has led to the measurement of genome-wide protein-DNA interactions and transcriptomes. A new generation of more sophisticated algorithms and software programs emerged to assist in the analysis of the first RNA-seq and ChIP-seq datasets.

Trapnell et al. (Trapnell et al., 2012) illustrated in the first stages of development of high-throughput mRNA sequencing that TopHat and Cufflinks are free, open-source software tools for gene discovery and the comprehensive expression analysis of high-throughput mRNA sequencing (RNA-seq) data. These allowed biologists to identify new genes and new splice variants of known ones. They also facilitate the comparison of genes and transcript expression under two or more conditions. Where Trapnell et al. (Trapnell et al., 2010) has tested Cufflinks to sequenced and analyzed more than 430 million paired 75-bp RNA-seq reads from a mouse myoblast cell line over a differentiation time series. They concluded that Cufflinks can illuminate the substantial regulatory flexibility and complexity in even muscle development and that it can improve transcriptome-based genome annotation.

Following this stage, Robinson et al. (Robinson et al., 2010) invented edgeR, a Bioconductor software package for examining the differential expression of replicated count data. An over dispersed Poisson model is used to account for both biological and technical variabilities. Empirical Bayes methods are implemented in this package to moderate the degree of over dispersion across transcripts, improving the reliability of inference. edgeR can be used even with the most minimal levels of replication, provided that at least one phenotype or experimental condition is replicated.

Moreover, (Anders and Huber, 2010d) used the DESeq R/Bioconductor package, which depends on negative binomial distribution, with the variance and mean linked by local regression. They proposed methods to infer the differential signal in count data correctly and with good statistical power, as well as to estimate data variability throughout the dynamic range and a suitable representative model.

Basing on the sequence of the historical development of detecting differential expression genes, in 2012, Dillies et al. (Dillies et al., 2013) reported that there has been no clear consensus on the appropriate normalization methods to use nor the impact of a chosen method on downstream

analysis. Dillies (Dillies et al., 2013) therefore addressed a comprehensive comparison of seven commonly used normalization methods (TC, UQ, Med, DESeq, TMM, Q, and RPKM) for the differential analysis of RNA-seq data. She then provided her recommendations on RPKM and TC, both of which were widely used (Liu et al., 2011); (Young et al., 2010); these were ineffective and should be abandoned in the context of differential analysis. The reason is that scaling counts by gene length with RPKM is not sufficient for removing bias (Bullard et al., 2010a); (Oshlack and Wakefield, 2009). Dillies explained that only DESeq and TMM can maintain a reasonable false positive rate without any loss of power. Consequently, Dillies said that these two methods performed much better than others for data with differences in library composition.

Schurch et al. (Schurch et al., 2016) concluded from a large-scale analysis of the required biological replicates in RNA-seq data that at least six replicates per condition are required for RNA-seq experiments. In addition, there should be at least 12 replicates per condition for the experiments, in which identifying the majority of all DE genes is important. Moreover, Schurch explained when researchers can utilize edgeR or DESeq2. In case an experiment has less than 12 replicates per condition, Schurch encouraged the use of edgeR or DESeq2; otherwise, he recommended the use of DESeq2.

In the evaluation of methods for differential expression analysis of multi-group RNA-seq count data, Tang et al. (Tang et al., 2015) conducted two pipelines based on the TCC package, which implemented a multi-step normalization strategy called DEGES; this approach uses the same principle as edgeR, DESeq2, and so on to identify DEGs (18.5%–45.7% of all genes). Based on the TCC package, which is best used for a three-group comparison, as well as a two-group comparison, Tang et al. recommended using a DEGES-based pipeline that internally uses edgeR for count data with replicates (especially for a small sample size). For data without replicates, Tang et al. recommended the use of DESeq2.

Jaskowiak et al. (Jaskowiak et al., 2018) systematically evaluated different choices that emerge naturally during the clustering process and their effects on quality. They provided an evaluation of the computational steps relevant for clustering cancer samples via an empirical analysis of 15 mRNA-seq datasets; the authors assessed the performance of four clustering algorithms (K-medoids and hierarchical clustering algorithms, namely, single-linkage, average-linkage, and complete-linkage clustering) and 12 distance measures. The authors found that the data should be log-transformed initially in cluster analysis. Regarding the choice of clustering algorithms, average-linkage and K-medoids provided sound results. In general, Jaskowiak et al. recommended the careful selection of a distance measure and then showed that symmetric rank-magnitude correction provides consistent and sound results in different scenarios.

The key idea in RNA-seq experiments is focused on the comparison of gene expression levels over multiple conditions, tissues, disease types, and phenotypes, among others. Most RNA-seq studies are designed to detect DEGs, which are genes whose expression levels differ between two or more conditions. Detecting DEGs can be considered a critical step for some subsequent objectives, such as clustering the gene expression profile and testing the functional enrichment of DEG sets.

Many methods have been used to detect DE genes based on RNA-seq data. Examples are Fisher's exact test (Bloom et al., 2009), the x^2 goodness-of-fit test (Marioni et al., 2008b), and the likelihood ratio test (Bullard et al., 2010b). Because the first technical evaluation of RNA-seq technology relied on technical replicates, the use of Poisson models whose variance is equal to the expected value $E[Y] = var[Y]$ for count data was initially proposed. However, when count tables contain biological replicates, RNA-seq data show more variability (the means for these RNA-seq data would have variance that is greater than the expected value they represent in a simple equation by $var[Y] > E[Y]$); **NB** distribution has been proposed as an alternative to Poisson to model counts with biological replicates. Depending on the NB models, many tests have been developed and implemented in R packages, such as edgeR (Robinson and Smyth, 2007), DESeq2 (Anders and Huber, 2010b), and baySeq (Hardcastle and Kelly, 2010).

Despite the widespread use of the above-mentioned approaches for detecting DE genes, there is no consensus nor theoretical justifications for which methods are optimal nor how the optimal test can be identified. In addition, the principle of most of those methods relied on the mean expression levels which are rigorously the same or not throughout all conditions, whereas sometimes, biologists are interested in detecting genes with expression changes that are larger than a certain threshold.

Another important issue with the above-mentioned methods is that supervised classification methods have not been previously assessed for their reliability with RNA-seq count data.

We therefore utilize these methods to generate the p-values for each gene, and then we arrange in ascending order all genes based on their p-values to test the impact of the number of variables on the effectiveness of the classifier. In other words, we determine the most significant features (variables) that affect the accuracy of the classifier.

Statistical methods for classification

Machine learning concepts

Supervised learning is considered when there are input variables (X) and an output variable (Y), and there are requisites to use an algorithm in order to learn the mapping function from the input to the output. $Y = f(X)$, where the essential goal is to approximate the mapping function well, and then when there are new input data (X) consequently someone can predict the output variables (Y) for these data.

This is called supervised learning; the process of algorithm learning from the training dataset can be envisaged as a teacher supervising the learning process because the correct answers are well known. The algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

But unsupervised learning is considered when there are only input data (X) and no corresponding output variables. The main goal of unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

This is called unsupervised learning because unlike supervised learning, there are no correct answers, and there is no teacher. Algorithms are left to their own devices to discover and present the interesting structure in the data.

To train the learning algorithm, a set of labeled training individuals is used in supervised learning. The main goal of a supervised learning approach is to predict an output variable for a set of individuals (the test set) based on knowledge gained from another set of individuals for which the output value is provided (the training set). Through the supervised learning family of methods, we can further differentiate between ***classification methods,*** which focus on the prediction of discrete (categorical) outputs, and ***regression methods***, which predict continuous outputs. Regression methods are beyond the scope of our study.

Motivation

Recent approaches, both biological and statistical, are increasingly needed to take advantage of recent advances in machine learning and the HGP for disease diagnosis and prognosis. Supervised and unsupervised classifications hold great promise for making classifications in the whole genome massive datasets recently generated by biologists. Various machine learning algorithms have been utilized to perform classification.

Despite their popularity in many fields of application, supervised classification methods have been seldom used for the analysis of NGS data, such as RNA-seq count data. Considering our biological motivation, we strive to predict classes (multi-class) for new individuals based on their transcriptome profile by using the supervised classification approach to train each supervised classifier in assigning individuals to predefined classes based on their expression profile. A priori, there is no obvious choice regarding the best method and parameters to classify NGS data. To elucidate our methodological objectives, we are concerned with the evaluation of different classifier methods according to various indicators, such as accuracy, generalization power (ability to correctly classify new individuals), and robustness to sampling variations and the variables.

Supervised classification algorithms

This study focuses on supervised classification, which implies that we dispose of an outcome variable; in this process, we train the classifier to recognize per-established classes and then use them to predict the class for new individuals.

Among the numerous classification methods reported in the literature, in our study, we focused on support vector machines (SVMs), random forests (RFs), and K-nearest neighbors (KNN). In the subsequent sections, we will briefly introduce these three supervised classification methods.

K-Nearest Neighbor (KNN)

KNN is a famous algorithm and the simplest one among all machine learning algorithms. It is quite simple and easy to implement, in which Cover and hart (Cover and Hart, 1967) shows that the error of the nearest neighbor rule is bounded above by twice the Bayes error under certain reasonable assumptions. Also, the error of the general KNN method asymptotically approaches that of the Bayes error and can be used to approximate it, and its idea is to memorize the training set and then predict the label of any new instance on the basis of the labels of its closest neighbors in the training set. The rationale behind this method is based on the assumption that the features used to describe the domain points are relevant to their labelling in a way that makes close-by points likely to have the same label. Furthermore, in some situations, even when the training set is immense, finding a nearest neighbor can be done extremely fast (Shalev-Shwartz and Ben-David, 2014).

Mathematically, $\rho: X * X \rightarrow \mathbb{R}$, where Ψ is a function that returns the distance between the two points of X (x_i, x_i'). The Euclidean distance between two points can be calculated by the following formula:

$$\rho(x, x') = |x - x'| = \sqrt{\sum_{i=1}^{d}(x_i - x_i')^2}, \quad (1.1)$$

where k in the KNN is the number of data points closest to the individual that has to be assigned to a class. For example, if k = 1, then the algorithm will assign a new individual to the class of the nearest one; if k = 4, then the algorithm will choose the closest four neighbor individuals and will classify them accordingly. The idea can be better exemplified with **Figure 4**. The KNN algorithm is implemented using the class package (R: a language and environment for statistical computing) in R.

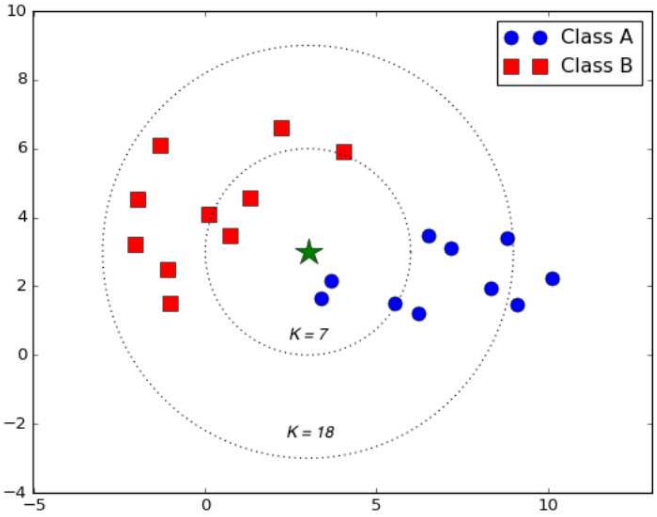


Figure 4 Illustration of the principle of k-nearest neighbors
The color red indicates class B, and the color blue indicates class A. With k = 7 chosen, this means that the majority vote will be favorable to the red class; the new sample will therefore be classified to the red class based on the four votes for class B against the three votes for the blue class. Modified from the K nearest neighbor and dynamic time wrapping (2016) (Time-Series Analysis: Wearable Devices using DTW and kNN).

Our motivation for utilizing supervised classification to classify the samples based on their respective classes is that we noticed the behavior of each classifier with RNA-seq count data that are downloaded from recount2 repository. We conclude that after the training process, the classifier can be used to assign an individual to existing classes. From our predictions, in the case of breast cancer, the status "classes" of the cancer samples (e.g. with cancer class, prognostic of success for a treatment) indicates the effectiveness of the supervised classification methods in performing the classification of the sample based on their respective class labels.

Some of KNN variations, such as weighted KNN and assigning weights to objects, are relatively well known, some of the more advanced techniques for KNN are much less known, in which it is typically possible to eliminate many of the stored data objects, but still retain the classification accuracy of KNN classifier. This is known as "condensing" and can greatly speed up the classification of new objects (Hart, 1968). In addition, data objects can be removed to improve classification accuracy, a process known as "editing" (Wilson, 1972) . There has also been a considerable amount of work on the application of proximity graphs (nearest neighbour graphs, minimum spanning trees, relative neighborhood graphs, Delaunay triangulations, and Gabriel graphs) to the KNN problem.

Decision Trees

Decision trees are powerful classifiers because of their high execution speed. The main traditional methods for growing trees cannot extend to high-complexity data sets because they are sensitive to over-fitting, and they loss their generalization power (the capability to correctly classify unseen data). The essence of the method is to build multiple trees in randomly selected subspaces of the features space. Trees in different subspaces generalize their classification in complementary ways, and their combined classification can be monotonically improved.

One advantage of decision trees is balancing the error for instances in which the class population is an unbalanced dataset; the generated forests can then be saved for future use on other data.

The common weakness of decision trees is that they are extremely sensitive to small perturbations in the data. A slight change can result in a drastically different tree. In addition, they can easily overfit, and even though this can be addressed by validation methods and pruning, the avoidance of overfitting is still non-trivial.

Random Forest (RF)

The RF algorithm (Breiman, 2001b) is a an ensemble approach that aggregates the results of many randomly constructed classification trees. Two components of randomness are presented into the building of individual trees. First, each tree is built using a random bootstrapped sample of the training data. Second, a random subset of variables is tested at each split in each tree, rather than all features being tested for the best split. The baseline principle from the concept of randomness into the construction of trees and the averaging of the result over many tress is that

the final outcome will be less subject to any random fluctuations in the training parts from datasets and will have an increased capacity for generalizing patterns. The prediction is made for unobserved data by taking the majority vote of individual trees. The samples that are not a part of the bootstrapped sample for each tree, referred to as out-of-bag (OOB) samples, are used to create a cross-validated prediction error for the forest. Furthermore, as a part of the construction of RF, the OOB samples are used to mold a measure of feature importance. This is done by randomly shuffling the values of each input feature in turn and observing how much the prediction error of the OOB samples has grown. The randomForest package in R was used in our study; we tested the majority of its parameters to produce the optimal results with RF in the task classification of RNA-seq data.

Support Vector Machine (SVM)

SVM is the most widely used method of supervised machine learning. The technique was introduced by Vapnik (Vapnik, 2000) and Giveki et al. (Giveki et al., 2012). SVM aims to identify the ideal boundary separating classes in feature space. This decision boundary is called the ideal separation hyper-plane. The classification of new samples from data is based on which side of the decision boundary the sample point falls. The ideal hyper-plane is chosen based on the maximum margin principle by choosing the boundary that maximizes the distance between classes. SVM can be used for treating problems in which classes are not linearly segregated by transforming the samples using a non-linear kernel function, such as the radial basis function (RBF) kernel. The RBF kernel is a common choice for classification tasks (Meyer) (Luts et al., 2010). We tested with our *RNAseqMVA* the following kernels: linear, polynomial, and sigmoid for the main kernel value. The SVM model tries to find the space in the matrix of data where different data classes can be widely differentiated, and draws a hyper-plane, as illustrated in **Figure 5**.

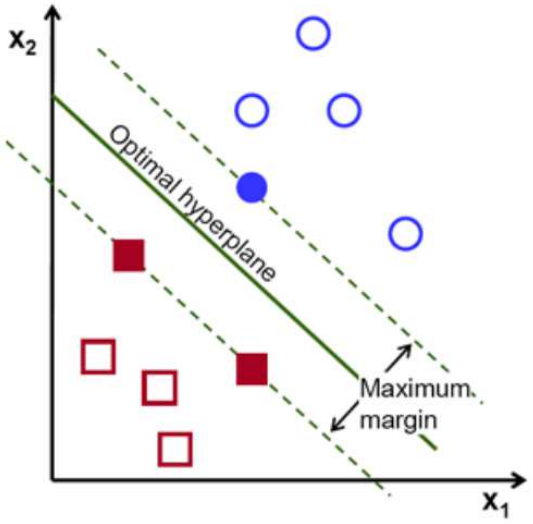


Figure 5 Schema illustrating the principle of the support vector machine
Source: Introduction to SVM (Introduction to Support Vector Machines — OpenCV 2.4.13.7 documentation)

In **Figure 5**, the colors red and blue depict the classes of labelled training data points. To classify these linearly, a hyper-plane can be drawn, but the issue is that there is more than one way to draw a hyper-plane, so which one is optimal? An optimal hyper-plane is chosen in a way that maximizes the margin between classes. It does not necessarily need to be linear. A hyper-plane in SVM can also work as a non-linear classifier by using the parameter kernel, as stated previously. We tested most of these parameters to obtain the best results that lead to the ideal use of the SVM with the optimal parameters for the classification of RNA-seq datasets. The SVM was implemented using e1071 R package (Meyer). However, from a practical point of view, the weakness of SVM is its high algorithm complexity and the extensive memory requirements of the required quadratic programming in large-scale tasks (Suykens, 2009).

Unsupervised learning

Unsupervised learning does not rely on a prior definition of the class labels of individuals (Aggarwal and Reddy, 2013). Clustering is an unsupervised technique that tries to group individuals in order to optimize the criterion reporting that the distance among individuals in the same cluster is minimized and the distance among individuals in different clusters is maximized (Tan et al.). A main issue in clustering is the choice of a relevant measure of the distance between a pair of individuals. Various similarity measures have been used for such a target, such as Euclidean, cosine, and city block distances. In traditional clustering, all features are used to compute the distance

between a pair of individuals. Alternatively, a subset of features can be selected prior to clustering on the basis of different criteria (e.g., choose some features that have similar properties to either discard redundant features or group them together).

A cluster is a group of individuals that are close to one another with respect to their mutual distance. In another meaning, these individuals are similar in nature over the whole set of features. However, in the case of a number of individuals available in a huge dataset, we aimed to find the groups of samples (individuals) that are similar over a subset of the available individuals. This type of clustering is called biclustering, in which each bicluster is associated with a subset of individuals. Clustering and biclustering analyze 2D data, in which each feature corresponds to an attribute of individuals. However, this is outside the scope of our study.

Clustering algorithms

A large number of clustering methods have been developed in the domain of machine learning. These clustering methods are basically classified into partitional clustering, hierarchical clustering, density-based clustering, graph theoretic clustering, soft computing-based clustering, and matrix operation-based clustering (Aggarwal and Reddy, 2013).

Hierarchical clustering methods can be classified into agglomerative and divisive methods (Berkhin, 2006). Agglomerative approaches operate in the bottom-up direction on the tree and start with nodes with individuals. These nodes are iteratively merged to reach the root of the tree. BIRCH (Zhang et al., 1996) is a popular agglomerative hierarchical clustering method that builds the clustering feature (CF) tree first, which operates in a bottom-up way to extract the clusters. But in the divisive technique, the root with all the nodes is iteratively split to finally reach the leaf nodes. DIANA (Kaufman and Rousseeuw, 2009) is a divisive hierarchical clustering method that splits the largest cluster iteratively to find splinter groups.

Evaluation of the performance of classifiers

Validation procedures

Validation procedures are commonly used to assess how well the classification algorithms can build accurate models for the data. Usually its results are affected by three main factors: (1) the accuracy of the underlying classification algorithm, (2) the correctness of the implementation of the algorithm, and (3) the characteristic of the training dataset.

It is often implicitly assumed that the implementation of the algorithm is correct. As KNN, RF, and SVM are extensively used classification algorithms, their predictive power is expected to

be reliable. For instance, given a reasonable training set, they should perform well in cross-validation. Therefore, in our experiments, we assumed that the implementation was correct for these three classifiers, and cross-validation was used to assess the performances of the classifiers on different datasets.

Resampling methods have become an integral tool in novel statistics; the essence of these mechanisms is based on repeatedly drawing samples from a training set of individuals and refitting a model on each sample in order to obtain additional envisages into that model. For instance, to examine the misclassification error rate (MER) and variability of classifiers, the key idea is to fit the classifier into each new sample and test the difference in the results. The underlying objective for this process is to better estimate how the classifier will perform on out-of-sample, real-life data.

Two methods could be applied for resampling: cross-validation and Bootstrap.

Cross-validation

Cross-validation (CV) is primarily used to validate the appropriateness of the classification algorithm to the given problem. It was proposed by Kurtz (1948 sample cross-validation) and extended by Mosier (1951 double cross-validation) and by Krus and Fuller (1982 multi-cross-validation). The main goal is to verify the replicability of results; similar to hypothesis testing, the objective is to determine if the results are replicable or just random.

Subsets of the data are held out for use as validating sets; a model is fit to the remaining data (a training set) and used to predict for the validation set. Averaging the quality of the predictions across the validation sets yields an overall measure of prediction accuracy. Cross-validation is employed repeatedly in building decision trees. The underplaying role of cross-validation for assessing how the results of classification will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and to estimate how accurately a predictive model will perform in practice.

Cross-validation is generally used to estimate the error associated with a given learning classifier by splitting the individuals into two sets, one for training and the other for testing. The training part is for fitting the classifier, and the testing part is for testing the computation of the prediction error, which is known as cross-validation. The results from cross-validation are evaluated by measuring testing and training errors. The testing error is the average error from the use of the statistical learning method to predict the response on a new individual, one that was not used in training the classifier. By contrast, the training error can be easily calculated by applying the statistical learning method to the individuals used in its training.

In our experiments, we conducted k-fold cross-validation, which is a typical cross-validation method. In k-fold cross-validation, the original sample set is randomly partitioned into k subsets ($k > 1$). Among k subsets, a single subset is retained as the validation data for testing the classifier model. The remaining subsets are used as training data. The cross-validation process is then repeated k times. The k results from k folds can then be averaged or summarized (or otherwise combined) to produce a single estimation (McLachlan et al., 2005). In cross-validation, a classifier is simply evaluated in terms of its respective fraction of misclassified instances, noted as the error rate. A lower error rate means better classifier performance.

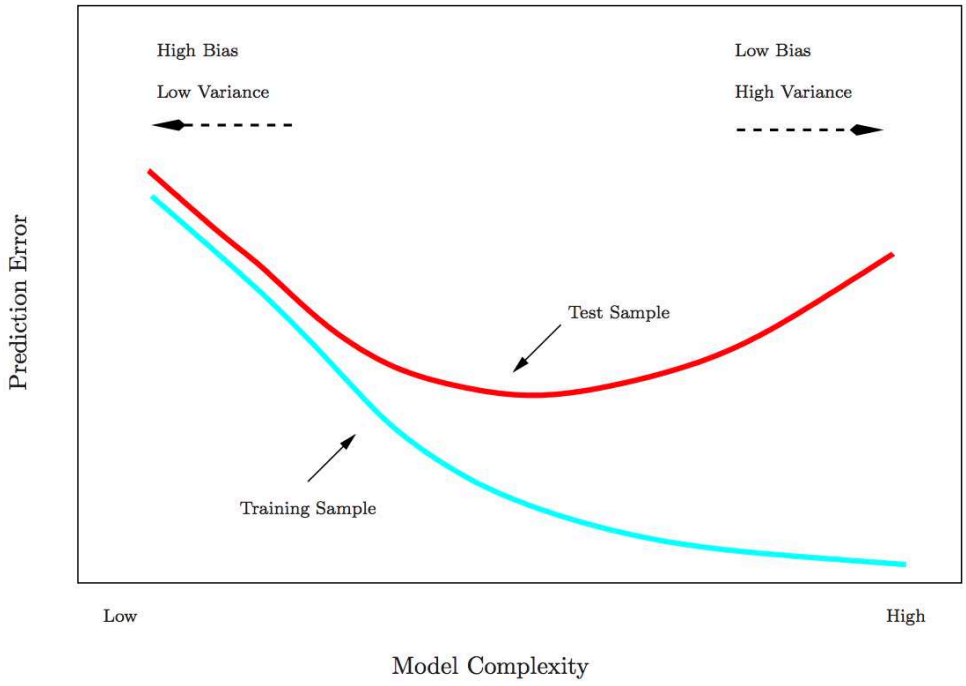


Figure 6. Curve illustrating the impact of model complexity on training and testing errors.
Source: "An Introduction to Statistical Learning, with Applications in R" (Springer, 2013), Model Complexity.

In a prediction problem, a model is usually given a dataset of known data on which training is run (training dataset), and a dataset of unknown data (first seen data) against which the model is tested (called the validation set or testing set). The goal is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias and to give an insight on how the model will generalize to an independent dataset.

Types of discriminator metrics to evaluate classifiers

After the testing phase, we dispose of two vectors of class labels: predicted and actual class labels for each individual. The question, then, is as follows: how do we measure the correspondence between them?

The first step is to build a confusion matrix and then derive some statistics from this matrix. Shown in Figure 7 are the confusion matrix and the derived statistics for multi-class problems.

In our study cases, we have a typical "two-class problem" with the psoriasis dataset (lesional psoriatic skin *versus* normal skin); this problem is easily handled, with one class being positive and the other being negative (as in Table 2). For the other datasets, however, we have multi-group class problems, (because the data sets contain more than two classes) or two classes without any reason to qualify one of them as positive and the other one as negative.

Table 3 shows the different statistics that can be used to assess classifiers. Considering our biological motivation, we aim at predicting classes (multi-class) for the new individuals based on their transcriptome profile by using the supervised classification approaches. To this purpose, we need to evaluate the accuracy of different classifier methods. We likewise seek to estimate the generalization power of the classifier (ability to correctly classify new individuals) and determine the robustness of each classifier to sampling variations and to variables.

The assessment metric for the classification problem has been utilized in two phases, which are the training phase (learning process) and the testing phase. In the training phase, the assessment metric was used to optimize the classification algorithm. This means that the assessment metric was utilized as a discriminator to distinguish and select the optimal classifier, which can produce a more accurate prediction of the future assessment of a particular classifier. By contrast, in the testing phase, the assessment metric was used as the evaluator to scale the effectiveness of the produced classifier when tested with the unseen data. Hossin and Suliman (2015) published an excellent review on how to study and construct metrics that are particularly designed to discriminate optimal classifiers during the training process.

Metrics for binary classification with one positive and one negative class

Some metrics are dedicated to binary classification. These can be defined based on the confusion matrix, as shown in Table 2. The rows of the table represent the predicted class, whereas the columns represent the actual class. From this confusion matrix, TP and TN denote the number of positive and negative instances that are correctly classified. Meantime, FP and FN denote the number of misclassified negative and positive instances, respectively. From Table 2, many commonly used metrics can be summarized.

Among them, we focus hereafter on the **MER** and its complement, accuracy (**Acc**). The **MER** is defined as the proportion of misclassification errors among all individuals:

$$\boldsymbol{MER = \frac{errors}{errors + correct}}.$$

Accuracy is then defined as the complement of the MER; it is the proportion of correct classifications among all the cases:

$$\boldsymbol{Acc = 1 - MER = \frac{correct}{errors + correct}}.$$

In binary classification, MER and accuracy are defined as follows:

$$MER = \frac{FP + FN}{FP + FN + TP + TN},$$

$$Acc = \frac{TP + TN}{totFP + FN + TP + TNal}.$$

In multi-class problems, accuracy is the ratio between the diagonal element of the confusion matrix and its total sum:

$$Acc = \frac{\sum_{i=1}^{c} n_{i,i}}{\sum_{i=1}^{c}\sum_{j=1}^{c} n_{i,i}},$$

$$MER = \frac{\sum_{i\neq j}^{c}\sum_{j=1}^{c} n_{i,j}}{\sum_{i=1}^{c}\sum_{j=1}^{c} n_{i,j}}.$$

where c is the number of classes and $n_{i,j}$ is the number of elements from class i assigned to class j.

As shown in previous studies (Chawla et al., 2004), (García and Herrera, 2008), (Hossin and Sulaiman, 2015),(Ranawana and Palade, 2006), and (Wilson, 2001), accuracy is the most commonly used evaluation metric in practice either for binary or multi-class classification problems. Through accuracy, the quality of the produced classifier is evaluated based on the percentage of correct predictions over the total instances. The complement metric of accuracy is the MER, which evaluates the produced classifier by its percentage of incorrect predictions. Both of these metrics are commonly used by researchers to discriminate and select the optimal predicted classes.

The advantages of accuracy or error rate are that they are easy to compute with less complexity, they are applicable for multi-class and multi-label problems (beyond the scope of this study), they facilitate easy-to-use sorting, and they can be easily understood by humans. As pointed out in many studies, the accuracy metric has limitations in evaluation and discrimination processes. One of the main limitations of MER is that it produces less distinctive and less discriminable values (Goksuluk

et al.). Consequently, it leads to less discriminating power for accuracy when selecting and determining the optimal classifier. In addition, accuracy is powerless in terms of informativeness.

Assessment of the performance of the classifier with different foci of evaluations. Because of multi-class problems, few of the metrics listed in **Table 3** have been used for multi-class classification evaluations.

Table 2 Confusion Matrix for Binary and Multi-class Classification

	Actual Positive Class	Actual Negative Class
Predicted Positive Class	True positive (TP)	False negative (FN)
Predicted Negative Class	False Positive (FP)	True negative (TN)

As shown in previous studies (Chawla et al., 2004); (Hossin et al., 2011); (Ranawana and Palade, 2006), accuracy is the most commonly used evaluation metric in practice either for binary or multi-class classification problems. Through accuracy, the quality of the produced classifier is evaluated based on the percentage of correct predictions over the total instances. The complement metric of accuracy is the error rate, which evaluates the produced classifier through its percentage of incorrect predictions. Both of these metrics are used commonly by researchers to discriminate and select the optimal predicted classes.

Table 3 Threshold Metrics for Classifier Evaluations.
Adapted from: Hossin and Suliman (2015). The last column indicates whether each metric is suitable for multi-class problems.

Metrics	Formula	Evaluation Focus	Multi-class?
Accuracy (acc)	$\frac{TP + TN}{TP + FP + TN + FN}$	In general, the accuracy metric measures the ratio of correct predictions to the total number of instances evaluated.	No
Error Rate (err)	$\frac{FP + FN}{TP + FP + TN + FN}$	The misclassification error measures the ratio of incorrect predictions to the total number of instances evaluated.	No
Sensitivity (sn)	$\frac{TP}{TP + FN}$	This metric is used to measure the fraction of positive patterns that are correctly classified.	No
Specificity (sp)	$\frac{TN}{TN + FP}$	This metric is used to measure the fraction of negative patterns that are correctly classified.	No
Precision (p)	$\frac{TP}{TP + FP}$	Precision is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class.	No
Recall (r)	$\frac{TP}{TP + TN}$	Recall is used to measure the fraction of positive patterns that are correctly classified.	No
F-Measure (FM)	$\frac{2 * p * r}{p + r}$	This metric represents the harmonic mean between recall and precision values.	No
Geometric Mean (GM)	$\sqrt{TP * TN}$	This metric is used to maximize the TP rate and the TN rate, simultaneously keeping both rates relatively balanced.	No
Average Accuracy	$\frac{\sum_{i=1}^{i} \frac{TP_i + TN_i}{TP_i + FN_i + FP_i + \text{The average effectiveness of all classes.}}}{l}$		No
Average Error Rate	$\frac{\sum_{i=1}^{i} \frac{FP_i + FN_i}{TP_i + FN_i + FP_i + the\ average\ erroe\ rate\ of\ all\ classes}}{l}$		No
Average Precision	$\frac{\sum_{i=1}^{i} \frac{TP_i}{TP_i + FP_i}}{l}$	The average of per-class precision	No
Average Recall	$\frac{\sum_{i=1}^{i} \frac{TP_i}{TP_i + FN_i}}{l}$	The average of pre-class recall	No
Average F-measure	$\frac{2 * p_M + r_M}{p_M + r_M}$	The average of the per-class F-measure	No

Note: C_i; i[th] class of data; TP_i: true positive for C_i; FP_i: false positive for C_i; FN_i: false negative for C_i; TN_i: true negative for C_i; and M macro-averaging.

We report here for the seven study cases tested in our study that the only dataset that had two classes was the psoriasis dataset (SRP035988). Correspondingly, with the rest of the datasets, we could not apply any metric from the above **Table 3** because those metrics are related to two classes of problem classifications. But in our case, most of our study cases belonged to the multi-class classification problem.

Instead of accuracy, FM and GM were also reported as good discriminators; it performed better than accuracy in optimizing the classifier for binary classification problems (Joshi, 2002).

For discriminating and selecting the optimal classifier during classification training, the significance tradeoff between classes is essential to ensure that every class is represented by its representative prototype. The tradeoff between classes becomes more crucial when imbalanced class data are used. The selection of the best classifier is useless if none of the minority class instances are correctly predicted by the chosen representative prototype or selected as the representative.

Receiver Operating Curve (ROC)

Aside from the above-mentioned different types of metrics, which are used to assess the accuracy of the classifier based on classification methods and to estimate the classifier, there are graphical-based metrics, which are better than accuracy, and have been presumed to evaluate the performance of classifiers. As stated by (Prati et al., 2011), these metrics can depict the tradeoffs between different evaluation perspectives, therefore allowing a richer analysis of the results. Although these metrics are better than accuracy or error rate, their graphical-based output limits them; examples are the receiver operating curve (ROC) (Fawcett, 2006), Bayesian receiver operating characteristics (Davis and Goadrich, 2006), the precision-recall curve (Davis and Goadrich, 2006), the cost curve (Drummond and Holte, 2006), and the lift and chart calibration plot (Vuk and Curk), which can be used as discriminators.

Area under the ROC curve (AUC)

AUC is one of the popular ranking-type metrics. In the works of (Hand and Till, 2001), (Huang and Ling, 2005), and (Rosset, 2004), AUC was used to construct an optimized learning classifier and compare learning algorithms (Provost and Domingos, 2003). Unlike the threshold and probability metrics, the AUC value can be calculated as follows:

$$AUC = \frac{S_p - n_p(n_n+1)/2}{n_p n_n}, \qquad (5.2)$$

where S_p is the sum of the all positive examples ranked, whereas n_p and n_n denote the number of positive and negative examples, respectively. AUC was proven to be theoretically and empirically

better than the accuracy metric (Huang and Ling, 2005) for evaluating classifier performance and discriminating optimal predicted classes during the training classifier.

Although the performance of AUC was excellent for evaluation and discrimination processes, its computational cost is high, especially for discriminating a volume of generated predicted classes of multi-class problems. To calculate the AUC for multi-class problems, the time complexity is $O(C|n\log n)$ for Provost and Domingos' AUC model (Provost and Domingos, 2003) and $O(C|^2\ n\log n)$ for Hand and Till's AUC model (Hand and Till, 2001).

Another weakness of ROC and AUC is that they are not relevant when strongly unbalanced classes are dealt with. In this case, the best way to obtain a very high AUC is to assign all objects to the majority class (but this is absolutely not interesting from the point of view of the user).

Choice of the validation metric for our study

(A) Confusion table

		Actual class				
		A	B	C	D	Predicted class sizes
Predicted class	A	124	4	1	1	**130**
	B	3	25	0	1	**29**
	C	4	2	10	0	**16**
	D	2	1	0	2	**5**
	Actual class sizes	**133**	**32**	**11**	**4**	**180**

Derived statistics		
Hits	= sum(diagonal)	161
Misclassified	= total - hits	19
Hit rate (accuracy)	= hits / total	0.89
Misclassification rate	= 1 - hit rate	0.11

(B) Opportunistic classifier "the majority takes it all"

		Actual class				
		A	B	C	D	Predicted class sizes
Predicted class	A	133	32	11	4	**180**
	B	0	0	0	0	**0**
	C	0	0	0	0	**0**
	D	0	0	0	0	**0**
	Actual class sizes	**133**	**32**	**11**	**4**	**180**

Derived statistics		
Hits	= sum(diagonal)	133
Misclassified	= total - hits	47
Hit rate (accuracy)	= hits / total	0.74
Misclassification rate	= 1 - hit rate	0.26

(C) Confusion table - strongly unbalanced data

		Actual class				
		A	B	C	D	Predicted class sizes
Predicted class	A	**1240**	4	1	1	**1246**
	B	**30**	25	0	1	**56**
	C	**40**	2	10	0	**52**
	D	**20**	1	0	2	**23**
	Actual class sizes	**1330**	**32**	**11**	**4**	**1377**

Derived statistics		
Hits	= sum(diagonal)	1277
Misclassified	= total - hits	100
Hit rate (accuracy)	= hits / total	0.93
Misclassification rate	= 1 - hit rate	0.07

(D) Opportunistic classifier "the majority takes it all" with strongly unbalanced data

		Actual class				
		A	B	C	D	Predicted class sizes
Predicted class	A	**1330**	32	11	4	**1377**
	B	0	0	0	0	**0**
	C	0	0	0	0	**0**
	D	0	0	0	0	**0**
	Actual class sizes	**1330**	**32**	**11**	**4**	**1377**

Derived statistics		
Hits	= sum(diagonal)	1330
Misclassified	= total - hits	47
Hit rate (accuracy)	= hits / total	0.97
Misclassification rate	= 1 - hit rate	0.03

Figure 7 Confusion matrix and the derived statistics for multi-class problems
Green: correct assignment (hits); red: incorrect assignment (misclassifications). (A) Confusion matrix and the derived performance metrics: hit rate and misclassification error rate. (B) Performance of an *opportunistic* classifier, which would assign all elements to the majority class. (C) Confusion matrix with strongly imbalanced classes. Observe the strong weight of the majority class on the other class predictions (bold red). (D) The opportunistic classifier with strongly imbalanced classes. Note that in this case, the opportunistic classifier achieves better results than the *honest* classifier (panel C).

Given our specific purposes for analyzing RNA-seq data, these metrics are unsuitable for discriminating and identifying the optimal classifier (Hand and Till, 2001); in addition, our work focuses on RNA-seq data, so it is related to other specificities of our study cases (multi-class problems; some classifiers do not produce an output score). Our work exploited the confusion matrices to discriminate between our targeted classifiers (SVM, KNN, and RF) during the classifier training because some metrics are appropriate for some classifiers but are not for others; therefore, we utilized the confusion matrix because it is the best one for performing a comparison between some classifiers that do not rely on the same concept of *mechanism* in classifying. And then we interpreted our results to take a decision about the choice of classifiers and their parameters. We illustrated our comparisons for assessing each classifier and then compared the classifiers to identify the optimal one for the analysis of RNA-seq data.

Goal of the thesis: evaluation of classifiers with RNA-seq data

Despite the wide adoption of RNA-seq technology to monitor transcriptome, and the innumerable publications relying on the detection of differentially expressed genes from RNA-seq, very few studies have been dedicated to the use of this technology for classification purposes. This contrasts with the important work that had been done in the beginning of the years 2000 on supervised classification and on clustering with microarray data.

The goal of my thesis is to assess the accuracy of supervised classifiers to assign samples to classes based on their RNA-seq transcriptome profiles. The evaluation covers three classifiers (support vector machines, k nearest neighbors, and random forest) representative of different families of algorithmic approaches to classification. I also evaluate the impact of standardization of data preprocessing (library size scaling, logarithmic transformation of the counts, conversion to principal components), and of the choice of classifier parameters (number of neighbors for KNN, kernel for SVM), as well as feature selection based on gene ranking with three alternative criteria (principal components, differential expression p-value, and variable importance after a first pass of random forest) in order to identify the optimal conditions to use classifier for predictive purposes.

CHAPTER 2: MATERIALS AND METHODS

Statistical analysis

The methodology used here compares three supervised machine learning methods: support vector machine (SVM), random forest (RF) and K-nearest neighbour (KNN). The evaluation relies on seven RNA-seq datasets (studies) downloaded from recount2 repository (Collado-Torres et al., 2017a),

We also studied the impact of normalisation methods on the performances of the classifiers by comparing four normalisation techniques:

1. Third Quartile (Q3),
2. Trimmed mean of M-Values (TMM),
3. Relative Log Expression (RLE),
4. Median Ratio (MA) method implemented in DESeq2 (Anders and Huber, 2010c)

The general pipeline for this study is recapitulated in Figure 8.

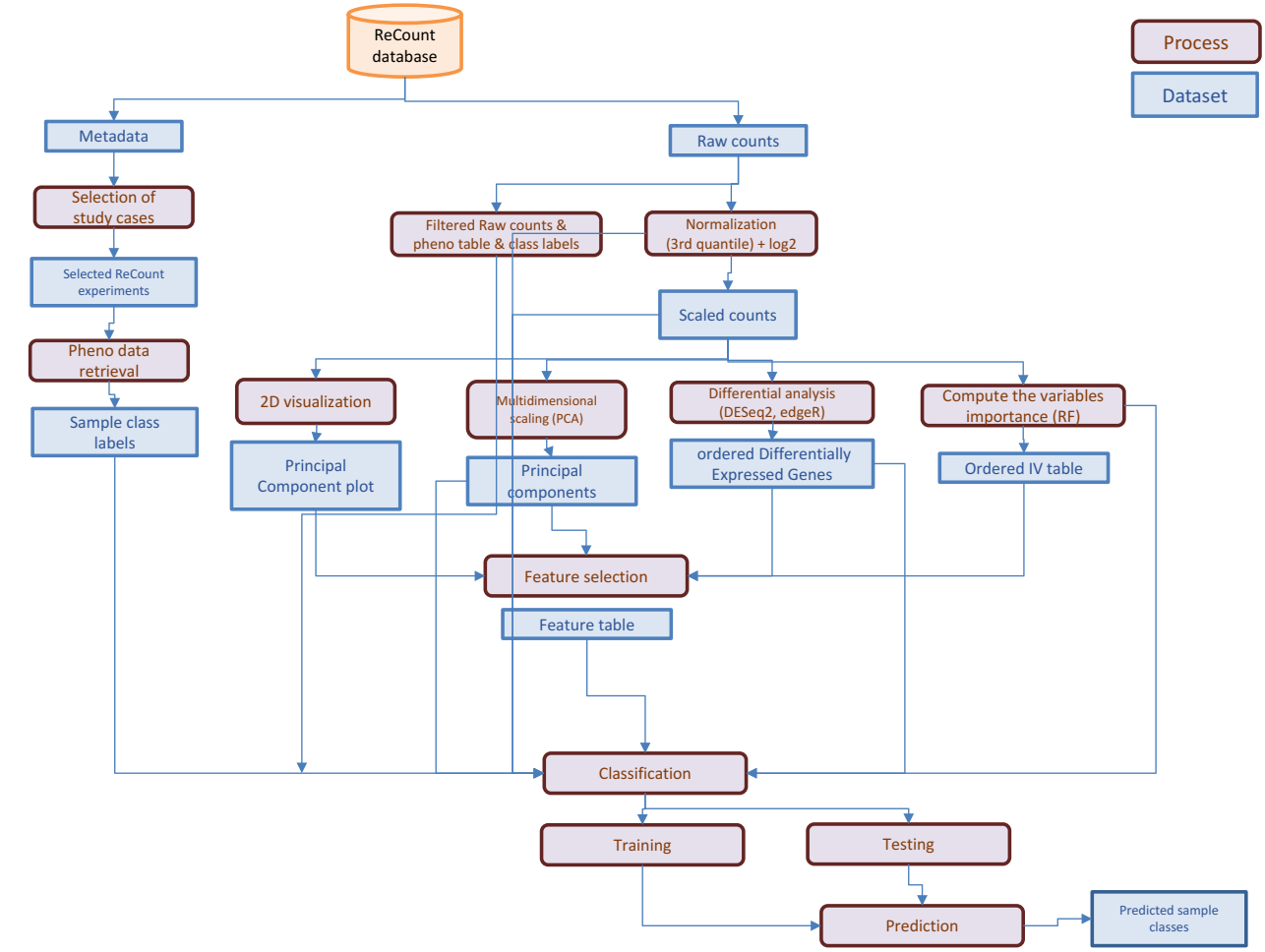


Figure 8 general flowchart for supervised classification methods to analysis RNA-Seq datasets.

We investigated RNA-seq data to assess the efficiency of supervised classification methods to perform supervised classification for samples based on the information corresponding to class descriptions that we could find in the phenotype table, in particular in the field "characterisation of the samples". This information was parsed to associate each sample with a class label, defined on the basis of one or more fields of the phenotype. The class labels were then used to train the classifiers and test their predictive accuracy. Our approach attempts to systematically evaluate the classifiers that rely on class labels for the samples, with the goal of providing a comprehensive analysis. We use the interesting features of RNA-seq to study the effectiveness of supervised classification methods to make classifications for all samples in the RNA-Seq datasets, which are stated in Table 4.

Table 4 summary of the seven datasets downloaded from recount2 repository as studies

Study	Description	Nb. Classes	Nb. Samples	Nb. genes
SRP042620	Breast cancer	6	167	58037
SRP057196	Adult and foetal human brain	15	461	58037
SRP056295	Human leukaemia	4	263	58037
SRP035988	Psoriasis	2	173	58037
SRP061240	Cancer disease types	4	192	58037
SRP062966	Lupus	3	117	58037
SRP066834	Cerebral organoids and foetal neocortex	3	729	58037

Several difficulties should be expected when classifying RNA-seq data, due to some particularities of this data type.

- Huge range of counts varying from gene to gene have boosted the unprecedented progress of multivariate statistical analysis of the RNA-seq data. Thereby, most trends go forward to machine learning science to overcome such a huge range of RNA-seq count data.
- In the presence of these raw count data, which contain a high resolution and broad dynamic range, this leads to the presence of several outliers as one of the drawbacks of these raw count data, i.e. a few genes associated with millions of reads, e.g. ribosomal RNAs.
- The prior consequence led to biases in statistical estimations that are induced by many undetected genes: zero or very low counts.
- Over-dimensionality in the RNA-seq count data which have too many variables, for example mean counts per gene for 2 tissue types is 0 to 2^{20} ~ 1M reads. Such issues stimulate researchers in machine learning to find solutions for such features of the RNA-seq count data.

Here we highlight the fact that most of the literature is focused on the classification of the two groups, whereas we are interested in multi-group classification problems.

Software environment and list of the most commonly used bioconductor packages

All of the statistical analyses used the R, RStudio, and Bioconductor packages to perform the differential expression analysis listed below:

- The DESeq2 package provides methods to test for the differential expression by using negative binomial generalised linear models; the estimate of dispersion and logarithmic fold changes incorporate data-driven prior distribution (Anders and Huber, 2010a) based on the hypothesis that most genes are not DE. A DESeq scaling factor for a given lane is computed as the median of the ratio, for each gene, of its read count over its geometric mean across all lanes.
- The edgeR package also provides methods to test for differential expression by negative binomial generalised linear models to estimate the dispersion and logarithmic fold changes base on Trimmed mean M-Values (TMM). This normalisation method is based on the hypothesis that most genes are nor differentially expressed (DE), while the TMM factor is computed as the weighted mean of log ratios between that test and the reference after the exclusion of the most commonly expressed genes and those with the largest log ratios.
- We developed the same environment using an R statistical package named *RNAseqMVA* which has been available in the GitHub repository since the turn of this year.

  The main target of this package is to employ machine learning methods to perform a comparative assessment of supervised classification algorithm efficiency to assign samples to classes based on the gene expression profile, and to identify the relevant procedures of data pre-processing and the optimal parameters of the classifiers.

A full list of R packages and versions is available in Appendix C.

Availability of the *RNAseqMVA* package

The *RNAseqMVA* package developed for this thesis is available at (https://github.com/elqumsan/RNAseqMVA). Each function if documented using the roxygen2 format. *RNAseqMVA* can be downloaded and installed easily with the devtools package[1].

[1] https://devtools.r-lib.org/

Machine learning methods

Machine learning is the field of computer science that includes efforts in the development of various computational methods that learn from training data.

Figure 9 schematises the categorisation of machine learning. The field is divided into two approaches: shallow learning and deep learning (Jabeen et al., 2018). Shallow learning consists of neural networks with a single hidden layer or SVM. They are simply supervised and unsupervised learning methods (Bhattacharyya et al., 2013). The supervised learning methods rely on classifiers whereas unsupervised learning implements a clustering algorithm. A supervised learning model learns from a set of predefined individuals with a class label (training set). The knowledge inferred from this is used to classify the unknown individuals (test individuals) accordingly, whereas unsupervised learning does not rely on the availability of prior knowledge (training data sets), meaning that it is beyond the scope of our study.

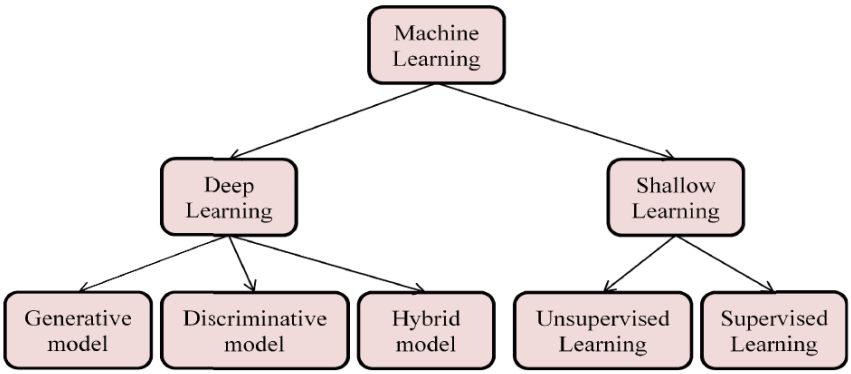


Figure 9 categorisation of machine learning technique.
Source: reference (Machine Learning-Based State-Of-The-Art Methods For The Classification Of RNA-Seq Data | bioRxiv)

The main goal of applying MLM into RNA-seq is the accurate prediction of the class labels of the sample based on their expression profile. Thus, RNA-seq is an excellent field for applying machine learning. In this section, we outline the support vector machine (SVM), random forest (RF), and K-nearest neighbour (KNN) being tested in such a study.

Misclassification error rate boxplots

To test the effectiveness of the supervised classification to handle the multi-class problem, we started by splitting the data sets downloaded from recount2 into two: a **training set** (2/3 of the samples from the whole data set) and a **testing set** (1/3 of the samples).

The performances were measured by the **misclassification errors**, defined as observations for which the class predicted by the trained classifier differed from the known class.

Parameters used for the classifiers

For the sake of traceability, all of the parameters are defined in a YAML-file included in the RNAseqMVA package. For our analyses, we used the following parameter values.

KNN: we used the function knn () from the class package and tested the following values for k (number of neighbours): 3, 5, 7, 10 , 15. Based on the results, we set the default value to k=10 for the main analyses.

SVM: we used the svm () function from the package e1071. We tested 4 kernels: linear, polynomial, radial and sigmoid. Based on this analysis, we set a kernel parameter as linear for all of the other analyses.

RF: we used the randomForest () function of the package randomForest with default parameters.

Data sources

Recount2

Collado-Torres and co-workers (Collado-Torres et al., 2017a) downloaded raw data (unaligned reads) for 2041 human RNA-seq publications available in GEO, and processed each of them with a custom analysis workflow to ensure the homogeneous treatment of all these datasets (indeed, all of the original studies were from thousands of publications where the authors had treated them in different ways depending on their preferences. The results of this huge compilation are available via Recount2, an online resource consisting of RNA-Seq count of reads per gene and exon wherein the tables contain one row per feature (gene or exon) and one column per sequencing library (run or sample). Recount2 also provides coverage profiles (bigwig files) for 2041 different studies. It is the second generation of the ReCount project. The raw sequencing data were processed with Rail-

RNA which is a cloud-enabled spliced aligner that analyses many samples at once. It also eliminates redundant work across samples, making it more efficient as samples are added (Nellore et al., 2017), as described in the recount2 article (Collado-Torres et al., 2017b) and Nellore et al. (2016), which created the coverage bigwig files. For ease of statistical analysis, each study contains count tables at the gene and exon levels and extracted phenotype data, in raw formats as well as in the form of RangedSummarizedExperiment R objects, whose structure can be loaded easily. These contain all of the information about one experiment, in addition to the data matrix, sample descriptions (pheno table) etc. (for details, see the SummarizedExperiment Bioconductor package) (SummarizedExperiment-class function | R Documentation).

The ready to use count tables, RangedSummarizedExperiment objects, phenotype tables, sample bigWigs, and file information tables are and freely available in the recount2 repository. The R package also allows the user to search and download the data for a specific study. That makes analysing RNA-seq data considerably more straightforward.

The phenotype information (sample metadata) is also included in RangedSummarizedExperiment object to facilitate the download of pheno tables from the recount2 repository.

Object-Oriented programming

The main ideas of object-oriented programming (OOP) are also quite simple and intuitive for the following reasons:

I. Everything we compute is an object, and objects should be structured to suit the goals of our computation.

II. For this, the key programming tool is a class definition stating that objects belonging to this class have a structure defined by the properties that they share, with the properties eventually being the objects of a specified class.

III. A class can inherit from (contain) a simpler superclass, such that an object of this class is also an object of the superclass.

IV. In order to compute with objects, we define methods that are only used when objects are of certain classes.

Many programming languages reflect these ideas, either from or by adding some or all of the ideas to an existing language. R was not an OOP language from its inception, but it has incorporated important software features reflecting the main ideas. In fact, it has done so in at least three separate forms, giving rise to some confusion.

In R, the natural role of methods correspond to the intuitive meaning of "method" - a technique for computing the desired results of a function call. In functional OOP, the particular computational technique is chosen because one or more arguments are objects from recognised classes.

Methods in this situation belong to functions, not to classes; the functions are generic. In the simplest and most common case, referred to as a standard generic function in R, the function in R language defines the formal arguments but otherwise consists of nothing but a table of the corresponding methods plus a command to select the method in the table that matches the classes of the arguments. The selected method is a function; the call to the generic is then evaluated as a call to the selected method. For the implementation of the *RNAseqMVA* package, we used this form of object-oriented programming as a functional OOP, as it works with principles in a form in which methods are part of the class definition.

We summarised hereafter the motivations for using OOP in the *RNAseqMVA* package.

- All of the parameters, variables, and results that belong together are put together under one title, i.e. the class name in the code.
- All of the functions that are used to manipulate the class are included.

- Using the OOP also facilitates the inheritance process, wherein by defining any object as belonging to a given class, it inherits all of the attributes of the ancestor classes, which speeds up the development of methods to compute, store and retrieve results.

In Appendix C, we provide further detail about the OOP implementation of the *RNAseqMVA* package, as well as an UML diagram showing the architecture of its classes.

CHAPTER 3: DESCRIPTION OF THE STUDIES

Recount2 is an online resource consisting of counts of reads per gene and exon, as well as coverage profiles (bigWig files) for 2041 different RNA-seq studies in humans, each study has a certain number of samples based on its experiment, but the overall number of the samples is more than 70,000. The raw sequencing data were processed with Rail-RNA read mapper. For the ease of statistical analysis, for each study, Collado-Torres et al. (Collado-Torres, 2017) created count tables at the gene and exon levels and extracted the metadata associated with the publication (description of the technical and biological characteristics of each sequencing run associated to each sample).

Large-scale RNA-seq datasets have been produced by studies such as the GTEx (Genotype-Tissue Expression) consortium (Lonsdale et al., 2013), which comprises 9,662 samples from 551 individuals and 54 body sites, and the Cancer Genome Atlas (TCGA) study (The Cancer Genome Atlas Research Network et al., 2013), which comprises 11,350 samples from 10,340 individuals and 33 cancer types; furthermore, public data repositories such as the sequence Read Archive (SRA) host tens of thousands of human RNA-seq samples (Leinonen et al., 2011). These data collectively provide a rich resource which researchers can use for discovery validation, replication, or method development.

We began our study by analysing the metadata of these datasets in order to identify suitable studies to assess the performance of supervised classifications.

We considered relevant criteria to retain a recount2 dataset in a suitable study.

The most important criteria are:

- Number of classes: for most of them we attempted to have more than 2 classes. Note, however, that after the filtering procedure (see below), some of our studies were restricted to 2 classes.
- Number of samples: studies covering several tens of samples.
- Clear identification of relevant biological classes from the metadata fields (the so-called "pheno table").

Table 5 provides a short description for each one of the selected datasets.

Table 5 summary description of the seven datasets from recount2 that we selected as studies. Gene-wise count tables cover 58,037 genes for each dataset. The last column indicates whether sequencing was performed at the level of single-cells (SC) or whole samples (bulk).

ID	Title	Summary of the experiment	Nb. Classes	Nb. Samples	Bulk / sc
SRP042620	Cancer type	This study aimed to determine fusion transcripts in breast cancer, by performing paired-end RNA-seq of 168 breast samples, including 28 breast cancer cell lines, 42 triple negative breast cancer primary tumours, 42 oestrogen receptor positive (ER+) breast cancer primary tumours, and 56 non-malignant breast tissue samples. PMID: 24929677	6	167	Bulk
SRP057196	Cellular complexity of the adult & foetal human brain	This study used single cell RNA-seq sequencing on foetal human cortical neurons to identify genes that are differentially expressed between foetal and adult neurons and those genes that display an expression gradient reflecting the transition between replicating and quiescent foetal neuronal populations. PMID: 26060301	15	461	SC
SRP056295	Human Leukaemia	Using next-generation sequencing of primary acute myeloid leukaemia (AML) specimens to identify the knowledge the first unifying genetic network common to the two subgroups of KMT2A (MLL)-rearranged leukaemia. PMID: 26237430	4	263	Bulk
SRP035988	Psoriasis	Using high-throughput complementary DNA sequencing (RNA-seq) to assay the transcriptomes of lesional psoriatic and normal skin. Polyadenylated RNA-derived complementary DNAs from 92 psoriatic and 82 normal punch biopsies were sequenced. PMID: 24441097	2	173	Bulk
SRP061240	Cancer disease types	In such experiments, RNA-seq analysis is performed on plasma extracellular vesicles derived from 50 healthy individuals and 142 cancer patients, to identify significant associations of these exRNAs with age, sex and different types of cancers. PMID: 26786760	4	192	Bulk
SRP062966	Lupus	Autoantibodies target the RNA binding protein Ro60 in systemic lupus erythematosus (SLE) and Sjogren's syndrome. It is not clear whether Ro60 and its associated RNAs contribute to disease pathogenesis. The goal for this experiment was to catalogue the Ro60-associated RNAs in human cell lines among other RNAs. PMID: 26382853	3	117	SC
SRP066834	Cerebral organoids and foetal neocortex	Utilising single-cell RNA sequencing (scRNA-seq) to dissect and compare cell composition and progenitor-to-neuron lineage relationships in human cerebral organoids and the foetal neocortex. PMID: 26644564	3	729	SC

CHAPTER 4. DATA PRE-PROCESSING

RNA sequencing (RNA-seq) is a great approach that exploits the advantages of next-generation sequencing technologies for the gene-expression profiling of organisms, but required specific methodological developments in order to fully exploit its potential. With the former data produced by transcriptome microarrays, advanced interdisciplinary research led to the emergence of robust methods for gene-expression-based classification of biological samples (Mooney et al., 2013) and (Natsoulis et al., 2005). However, the vast majority of the statistical methods proposed for the classification of gene-expression data are either based on a continuous scale (e.g. microarray data) or rely on a normal distribution assumption. Thus, with these classical methods for differential expression analysis, unsupervised or supervised classification cannot be directly applied to RNA-seq data since these have a discrete nature and do not fulfil the distributional assumptions. It is therefore recommended to perform pre-processing before applying these algorithms.

For our analysis, we applied the following pre-processing steps:

1. **Class filtering.** Discard classes with an insufficient number of samples to be suitable for classification. We set the minimum number of samples per class to 10.
2. **Gene filtering.** Discard genes that are not suitable, for different reasons (zero values, zero variance, outliers with huge number of counts, etc…).
3. **Library size scaling.** Correct for differences in library sizes (library size correction, scaling). methods are Trimmed mean of M-values (TMM) which is implemented by edgeR Bioconductor package, quantile (Q, many different methods have been proposed, and this really affects the results as that have been of certain from A comprehensive evaluation of normalization methods that are performed by (Dillies et al., 2013).
4. **Log2 transformation.** Attenuate the huge differences in the dynamic range of the counts (some genes have a few counts, others have hundreds of thousands counts in each sample) methods: log2transformation.
5. **Principal Component transformation.** Reduce the number of dimensions (can be useful for some purposes): Principal component analysis (PCA) methods.

In the next chapter we will study the impact of pre-processing on the performances of supervised classifiers. In this chapter, we will describe some of the pre-processing steps and use descriptive statistics to explore the structure of the data before and after pre-processing. We summarise here the main results; detailed results are provided in Dillies et al. (2013).

Filtering procedures

Filtering classes based on sample number

Along with the seven selected studies, we filtered out the classes that contain fewer than 15 samples, since the scarcity of individuals in this class would be misleading for classifiers.

That means that in cases with insufficient samples, the classifier is poorly trained and there is a risk of overfitting to the particular individuals used for training, which would result in a poor generalisation power.

Figure 10 shows the distribution of the number of samples per class, and highlights in white those classes which were discarded because they did not reach the threshold of 15 samples. This is the first step of the pre-processing of RNA-seq data, i.e. class filtering. In particular, for the brain cell type dataset, almost all of the classes corresponding to hippocampal neurons are filtered out (only one hippocampal class is kept), which means that the evaluation will be mainly based on cortex cells. This is quite important, because it means that the evaluation will be based on only one of the two main types of neurons included in the original study.

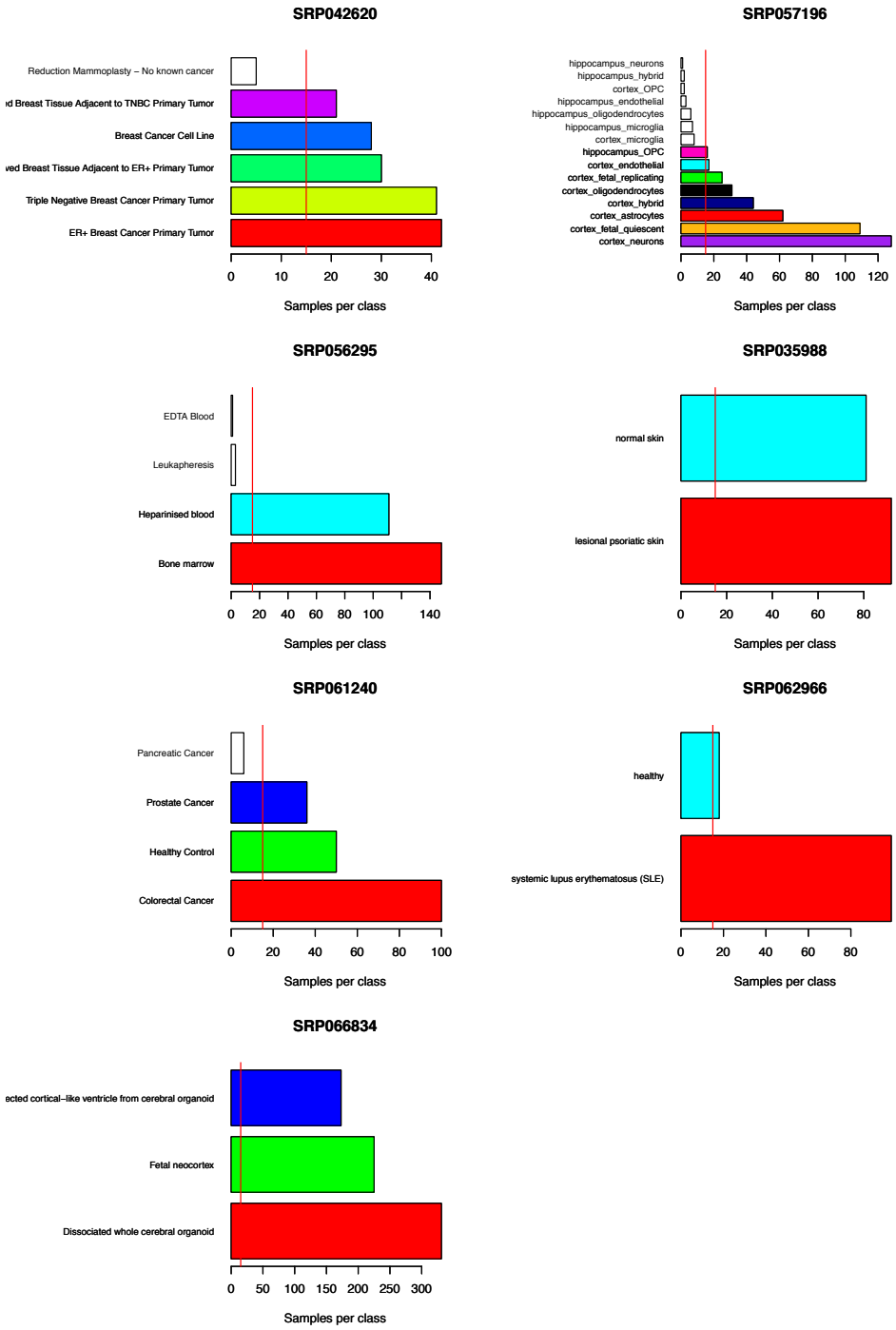


Figure 10 Filtering classes based on the sample number.
The bar plots show the number of samples (abscissa) per class (ordinate) in the 7 studies. The vertical bars indicate the lower threshold of samples per class. We retain only those classes containing at least 15 samples. The classes discarded because they failed to pass this threshold are highlighted in white.

Gene filtering

The second step of pre-processing process is feature filtering (gene filtering), which consists of selecting a subset of genes that will be kept for use with the supervised classifier and – optionally – to apply some transformation of the raw data. Gene filtering includes the following steps: discard all genes that have zero variance and genes and NA values. Thereafter, we applied a so-called "near-zero variance" filter to exclude all genes that have near-zero variance also from the raw table.

The necessity for these pre-processing steps comes from the fact that if these suppressed genes are retained in the analyses, it will influence the reliability of the classifier in some cases, while, in other cases, the classifier will simply crash with unfiltered features. Supplementary figures in Appendix A1 provide further illustrations of the necessity for pre-processing to tackle the RNA-seq data.

Single-cell RNA-seq (scRNA-seq) is a powerful high-throughput technique enabling genome-wide transcription levels to be measured at the resolution of a single cell. Given the low sequencing depth per cell in single-cell RNA-seq (a few thousands reads per cell), some genes may fail to be detected even though they are expressed. These missed genes are typically referred to as *dropouts*. Risso et al. (2018) suggested using the general and flexible zero-inflated negative binomial model (ZINB-WaVE) that accounts for zero inflation (dropouts), over-dispersion, and the count nature of the data.

Risso et al. (2018) defined **"**Zero inflation" as a data set that includes an extreme number of zeros. Indeed, zero inflation leads to the invalidation of the underlying distributional assumptions of standard parametric analysis and thus undermines the validity of the scientific inference (Lambert, 1992). The zeros could also strictly aggravate the numerical conditions of the data and cause computational hardness (Tu and Zhou, 1999) and (Li et al., 1999). However, much of the existing literature on zero inflation has highlighted count data (Gurmu et al., 1999; Cameron and Trivedi, 1986). Data need follow a specific count data distribution to be zero inflated (Hall, 2000; Vieira et al., 2000).

A breast cancer case study (SRP042620) is a simple example for a bulk RNA-seq (**Figure 11**) where we can quickly notice that there is diversity in the variance of genes in such a dataset. The grey histogram (top panel) indicates the distribution of log2 (variance) for all genes with a non-null variance. The majority of the genes have variances varying from 0 to 33,554,432 (their log2 spans range from -10 to 25 on the histogram, where genes with null variance are not represented); in such

case studies, some genes have much higher variance of around $4.39 \bullet 10^{12}$ (log2 ~42). The orange histogram (second panel) shows those genes that are discarded by the near-zero variance filter. Consistently, these genes are concentrated in the low ranges of variance. The green histogram (third panel) shows the genes kept after filtering, i.e. all of the genes remaining after we subtract those with zero or near-zero variance. The last panel (bottom) indicates the distribution of zero values (abscissa) per gene (the ordinate shows the number of genes having a given number of zero values). With the *Breast* cancer study, we observe that the vast majority of the genes are on the extreme left of the histogram, indicating that they have a null value in zero samples (in other terms, they are detected in all the samples, which reflects a very good genomic coverage).

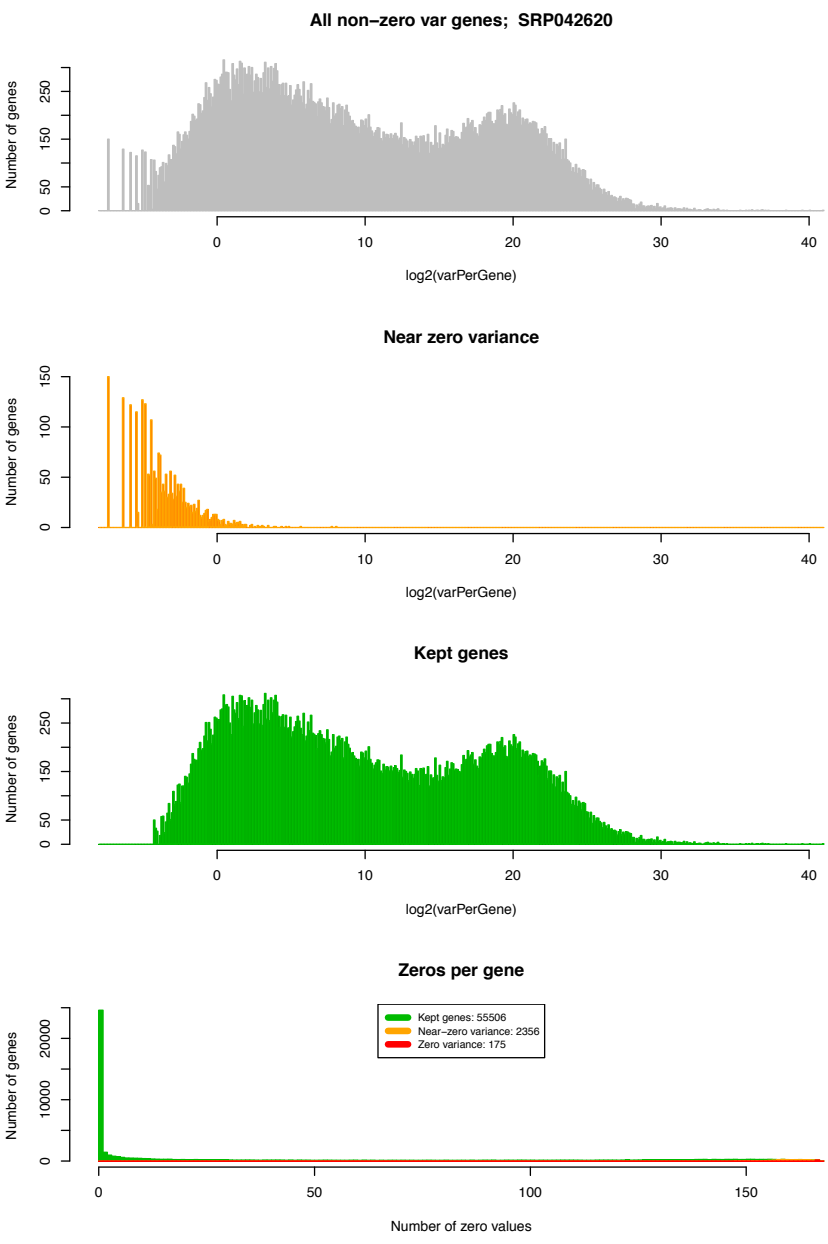


Figure 11 Impact of variance-based gene filtering on the study *Breast Cancer* (SRP042620). The histograms indicate the distribution of variances for a Bulk RNA-seq (abscissa) per gene (ordinates) in the raw data (**top panel, grey**), in the genes discarded by the near-zero filter (**second histogram, orange**), and in the genes kept after filtering (**third histogram, green**). The **bottom** histogram shows the number of genes (ordinate) as a function of the number of samples (abscissa) with a zero value for these genes.

Figure 12 **Impact of variance-based gene filtering on the case study *Cellular complexity of the adult & foetal human Brain type* (SRP057196).** The histograms indicate the distribution of variances (abscissa) per gene (ordinates) in the raw data (**top panel, grey**), the genes discarded by the near-zero filter (**second histogram, orange**), and the genes kept after filtering (**third histogram, green**). The **bottom histogram** shows the number of genes (ordinated as a function of the number of samples (abscissa) having a zero value for these genes.

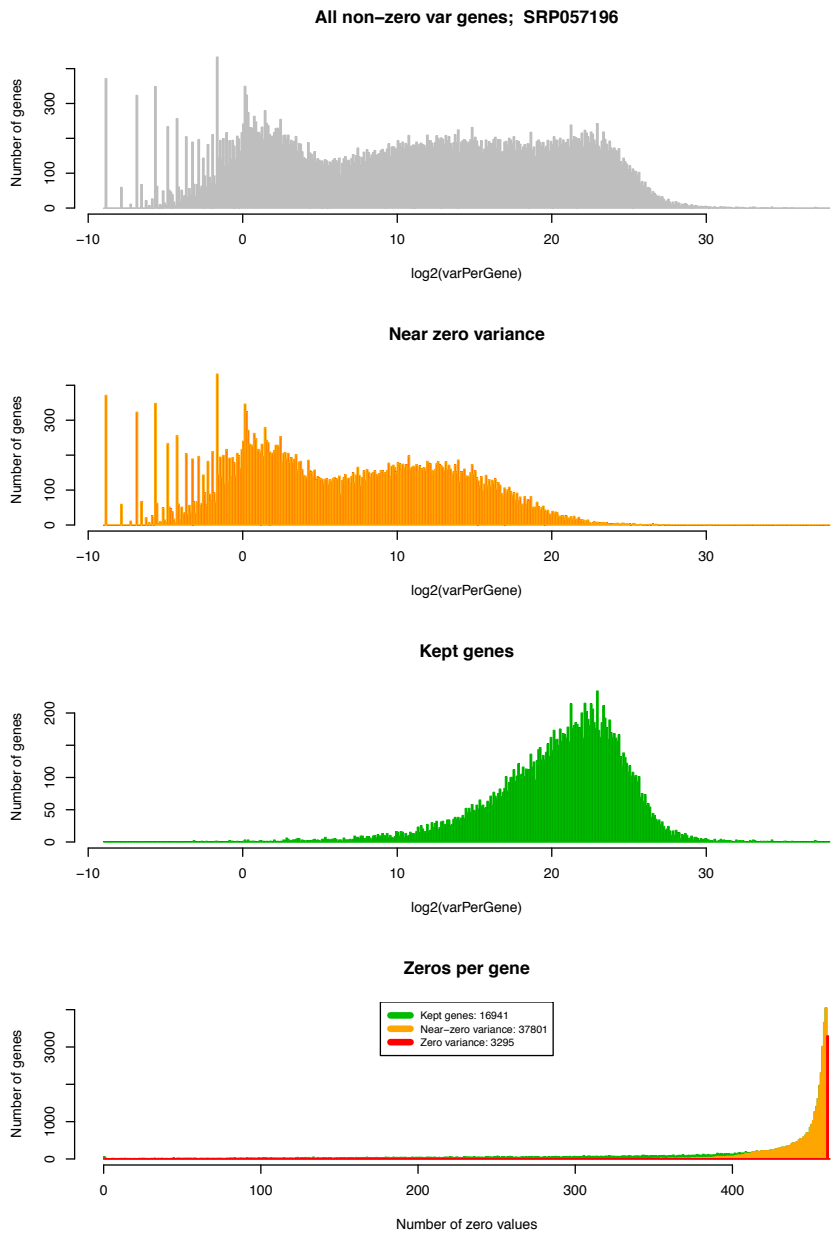


Figure 12 highlights a grey histogram for all of the non-zero variance genes, wherein that histogram contains some genes that have very high variance ~ 10^6. The surprising thing is the orange histogram, which indicates that near-zero variance also includes some genes that have a very high variance (2^20 = 10^6, which is far from "near-zero"), which is very different from the function of near-zero variance!!! In our analysis, the near-zero variance filter is an optional parameter in the pre-processing step, meaning that if the analyst requires that parameter to be

activated, this will lead to some genes being discarded whose variance is very high, and which actually affect the thematic accuracy of the whole result for the classifier.

The concept of "near-zero variance" (NZV), as defined in the R caret package, is to filter out the genes that are likely to be poor predictors for classification for either of two reasons: (i) genes that have a unique value across all samples (i.e. are zero variance predictors) or predictors that are have both of the following characteristics: they have very few unique values relative to the number of samples and the ratio of the frequency of the most common value to the frequency of the second most common value is large. If we used this NZV that means we will suppress many of variables from the analysis and then that will lead to an inaccurate evaluation for classifier that is because many of variables are excluded from analysis.

The common value here is 2^20, which is strongly represented because it has very high variance; the second one also has unique high variance. From the NZV side, this means that it will represent them as the genes have NZV. Consequently, to illustrate how these genes are dropped-down in the case of NZV parameters, within the analysis, we implemented an option enabling to discard then genes tagged as NZV, in order to feed the supervised classifier with supposedly better predictor variables. We tested both approaches (with and without NZV filtering) and compared the performances of the classifiers.

Distribution of counts per reads

From the distribution of counts per reads, we explored the behaviour of each dataset from the seven selected raw datasets, as shown in **Figure 11**. On this figure, we display the log2-transformed variances in order to better emphasize the distribution over its full range, from very low value (left) to very high values (right). For a better explanation, see Appendix A1, Supplementary figures, which provides the vision of the distribution for each one from the selected dataset.

Zero-inflated distributions

RNA-seq has become an attractive alternative to microarrays for specific differentially-expressed genes between several conditions or tissues, as it allows for the high coverage of the genome and enables the detection of weakly expressed genes (Marguerat and Bähler, 2010). This is based on the underlying characteristics of the Bulk RNA-seq, which are specialised to measure genome-wide transcription levels in bulk RNA-seq. Because of the low amount of RNA sequencing in cells, some genes may fail to be detected even though they are expressed. Risso et al. (2018) coined the new term "dropouts" to denote those genes which are undetected due to the weak sequencing depth of single-cell sequencing. According to that concept, they handled the corresponding distributions of counts by using zero-inflated negative binomial model (ZINB-WaVE), which leads to low-dimensional representations of the data that account for zero inflation (dropouts), over-dispersion, and the count nature of the data. Here in this work, we relied on such a study which confirms the necessity of the filtering process in order to make up the supervised classification for these RNA-seq data. Our finding leads us to realise the necessity to filter raw data before making any supervised classification, as we filtered the raw data from genes that have Na values, zero variance, or near-zero variance; for the sake of fairer comparison, we also suppressed duplicates in runs as there are some from the RNA-seq dataset which have two-fold levels. To ensure a fair comparison, we got rid of all duplicate samples. After all pre-processing (filtering), the raw datasets are ready for supervised classification to be applied.

Summary of the impact of zero-variance and near-zero variance filters

Figure 13 summarises the proportion of genes discarded by the zero-variance filter (red) or near-zero variance filter (orange) or kept after filtering (green). **Table 6** shows the corresponding numbers of genes. We can conclude that the proportion of kept genes strongly differs between datasets, depending on the type and kind of experiment (single-cell or bulk DNA) and the quality of the raw data (sequencing depth, number of zero values). The most striking case is SRP061240 (Cancer disease type), where the level of retained genes was 12%.

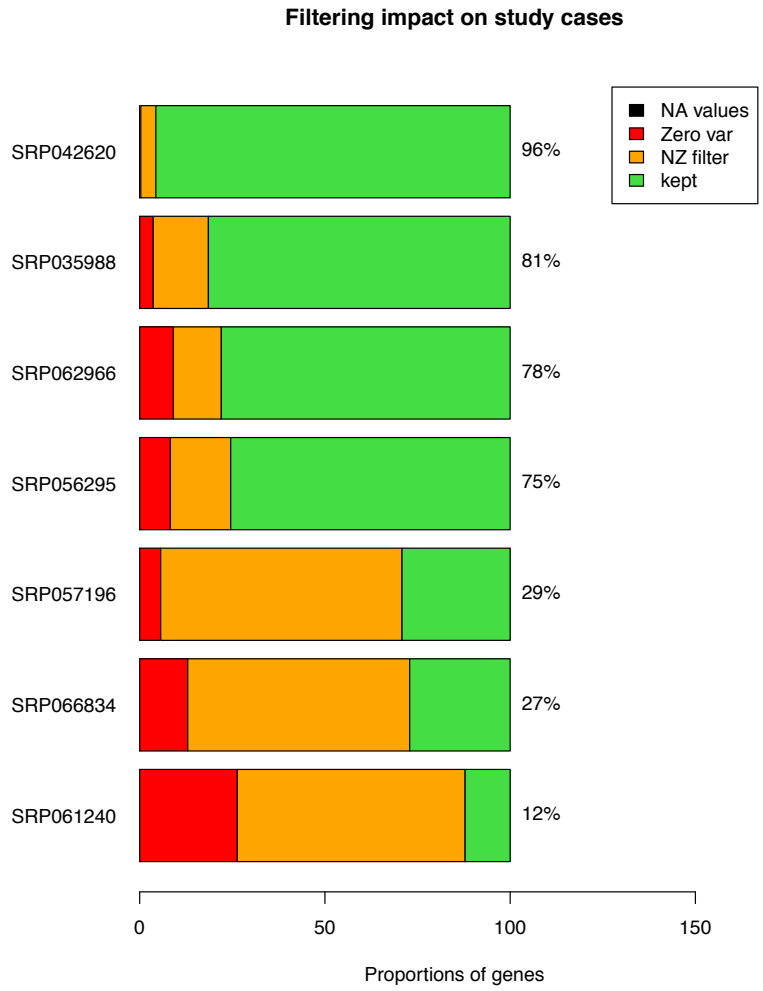


Figure 13 proportions of kept, near-zero variance, zero variance and NA values in each selected dataset.
The bar plot shows the proportion of genes discarded by the zero-variance filter (red) or near-zero variance filter (orange), and those kept after filtering, which have been used to feed the supervised classifier. Note that the near-zero variance filter suppresses the large majority of the genes in single-cell RNA-seq experiments (SRP057196: *Adult and foetal human brain*; SRP062966: *Lupus*), but also in a bulk experiment that contains an exceptionally high number of zero values (SRP066834: *Cerebral organoids and foetal neocortex*). For the assessment of classifiers, we disabled the NZ filter in order to dispose of as many genes as possible for the training, and let the classifiers evaluate whether they consider each gene to be relevant or not.

Table 6. Number of classes, samples (individuals) and genes (features) for the 7 studies before and after filtering.
Abbreviations: ZVF: zero variance filter; NZVF: near-zero variance filter; SC: single-cell RNA-seq sequencing. Libraries are sorted by decreasing size of retained genes after NZVF.

Study description			Before filtering			After filtering			
ID	Name	Type	Classes	Samples	Genes	Classes	Samples	Genes kept after ZVF	Genes kept after NZVF
SRP042620	Breast cancer	Bulk	6	167	58,037	5	162	57,862	55,506
SRP035988	Psoriasis	Bulk	2	173	58,037	2	173	55,915	47,270
SRP062966	Lupus	SC	3	117	58,037	2	117	52,746	45,247
SRP056295	Human leukaemia	Bulk	4	263	58,037	2	259	53,224	43,780
SRP057196	Adult and foetal human brain cells	SC	15	461	58,037	8	432	54,742	16,941
SRP066834	Cerebral organoids and foetal neocortex	SC	3	729	58,037	3	729	50,504	15,741
SRP061240	Cancer disease types	Bulk	4	192	58,037	3	186	42,736	7,056

For the case study *Cellular complexity of the adult and foetal human brain* (SRP057196), we note that the proportion of kept genes is around 29%, and the fraction of genes discarded by the NZ variance filter is large. This is consistent with the fact that this case study belongs to single-cell sequencing. The same effect is observed for the two other studies based on single-cell sequencing (SRP062966 and SRP066834). We can conclude that when the study corresponds to a single cell, the proportion of kept genes will be relatively low, because the near-zero variance filter suppresses a large fraction of the whole set of genes. The proportion of genes with null variance is also generally larger in single-cell experiment compared with the other datasets.

In contrast, with bulk RNA-sequencing, such as in the study *Human leukaemia* (SRP056295), the level of retained genes is larger (around 75%) and the near-zero variance and zero variance filters remove a smaller fraction of genes. The same is observed with the study *Lupus* (SRP062966), where a high level of kept genes was observed (78%), as well as in the studies *Psoriasis* (SRP035988) and *Breast cancer* (SRP042620).

In brief we can conclude that, for 3 of the 7 case studies, the large majority of genes are discarded by the near-zero variance filter. However, contrary to that suggested by the name of this filter ("near-zero variance"), the discarded genes cover a very wide range of variances. Their filtering out results in the presence of many zeros. This effect is not systematically associated with the dropout effect of single-cell sequencing. Indeed, it is observed in one bulk dataset (cancer disease types) (SRP061240), and in only two of the three single-cell studies.

Strikingly, genes with high variance but many zero observations might *a priori* be excellent discriminators between classes, especially if their large variance is due to the fact that they are highly expressed in some classes, and completely absent from others. The suppression of these genes might thus affect the effectiveness of classifiers and in some cases will lead to an erroneous estimation of their performances. We therefore chose to avoid the near-zero filter for the evaluation of classifier performances (Chapter 5).

However, from the issue in the above, pertaining to visualising the retained genes and the number of zero values, we have an open-ended question: what is the difference between the zero variance genes and near-zero variance genes which is the key function the CARET package? The authors Jin X (Jin and Bie) have a main target from the nearZeroVar function which diagnoses not only the predictors that have one unique value (namely that have zero variance), but also those with both of the following characteristics:

1- They have very few unique values relative to the number of samples
2- The ratio of the frequency of the most common value to the frequency of the second most common value is large.

PCA transformation

Principal component analysis (PCA) is an unsupervised approach which can be used to explore multivariate datasets, and can provide some hints about their internal structure. **Figure 14** to **Figure 20** show the results of the PC-transformation of read counts for the 7 selected case studies.

We first focus on the *Breath cancer* study (SRP042620) displayed in **Figure 14**. The variance bar plot (top-left panel) shows the distribution of the variance over the 9 first components. The variance is steadily decreasing from PC 1 to 5, and then shows a lower slope.

Given the fact that PCA is unsupervised, depending on the datasets, the first components sometimes reveal some clustering of objects according to their classes. This is partly true for the breast cancer sample SRP042620, as shown in panels 2 and 3 of Figure 14.

We further analysed the informativeness of PCs 1 to 6 to decouple the classes of the RNA-seq of 168 breast samples of this dataset, which included 28 breast cancer cell lines, 42 triple negative breast cancer primary tumours, 42 oestrogen receptor-positive (ER+) breast cancer primary tumours, and 56 non-malignant breast tissue samples.

The top-right panel shows that PC1 segregates very well with the samples belonging to the class "*Breast cancer cell line*" (red dots). Indeed, we could virtually draw a vertical line around position -150 that would almost perfectly separate these cell line samples from all the other classes, which come from primary tumours. In addition, a vertical line around position 100 on PC1 may segregate the two classes of "*uninvolved breast tissue*" from the three other classes. However, the two classes of uninvolved breast cancer tissue (resp. "adjacent to TNBC primary tumour" and "adjacent to ER+ primary Tumour") are intermingled on the first component.

In contrast with PC1, the second component (PC2, vertical axis on top-right panel) fails to segregate any of the 5 cancer types. For example, there is no single horizontal line that could clearly separate the samples from cell lines (red dots) and those from primary tumours (all other colours). However, the combination of PC1 and PC2 (top right panel of Figure 14) clearly improves the segregation: we could easily draw a diagonal line that would completely decouple cell lines from primary tumours.

The bottom-left panel shows that PC3 against PC4 is more effective than PC2 to segregate the samples of different classes. PC3 is able to decouple the "*Triple negative breast cancer primary tumour*" (green dots) from the two classes of "Uninvolved beast tissues" (pink and blue dots), Moreover, PC4 perfectly separates the green dots from the yellow dots, although we could remark that these dots were completely intermingled on the PC1-PC2 plot.

It is thus completely incoherent to claim that "the combination of PC3 and PC4 did not render any improvement".

For the sake of further exploration, we also plotted PC5 and PC6 (bottom-right). For this study, the combination of these PCs would not enable us to segregate any of the five classes.

Interestingly in this study, a small number of first PCs (from 1 to 4) can capture a substantial proportion of the NGS variance (top-left panel).

We could also probably catch the most relevant information by retaining only a few PCs as features for a classifier, as will be evaluated in Chapter 6 (feature selection), consequently decreasing

the power of PCs in detecting variance associated with these data so that we can discard these lower-variance PCs.

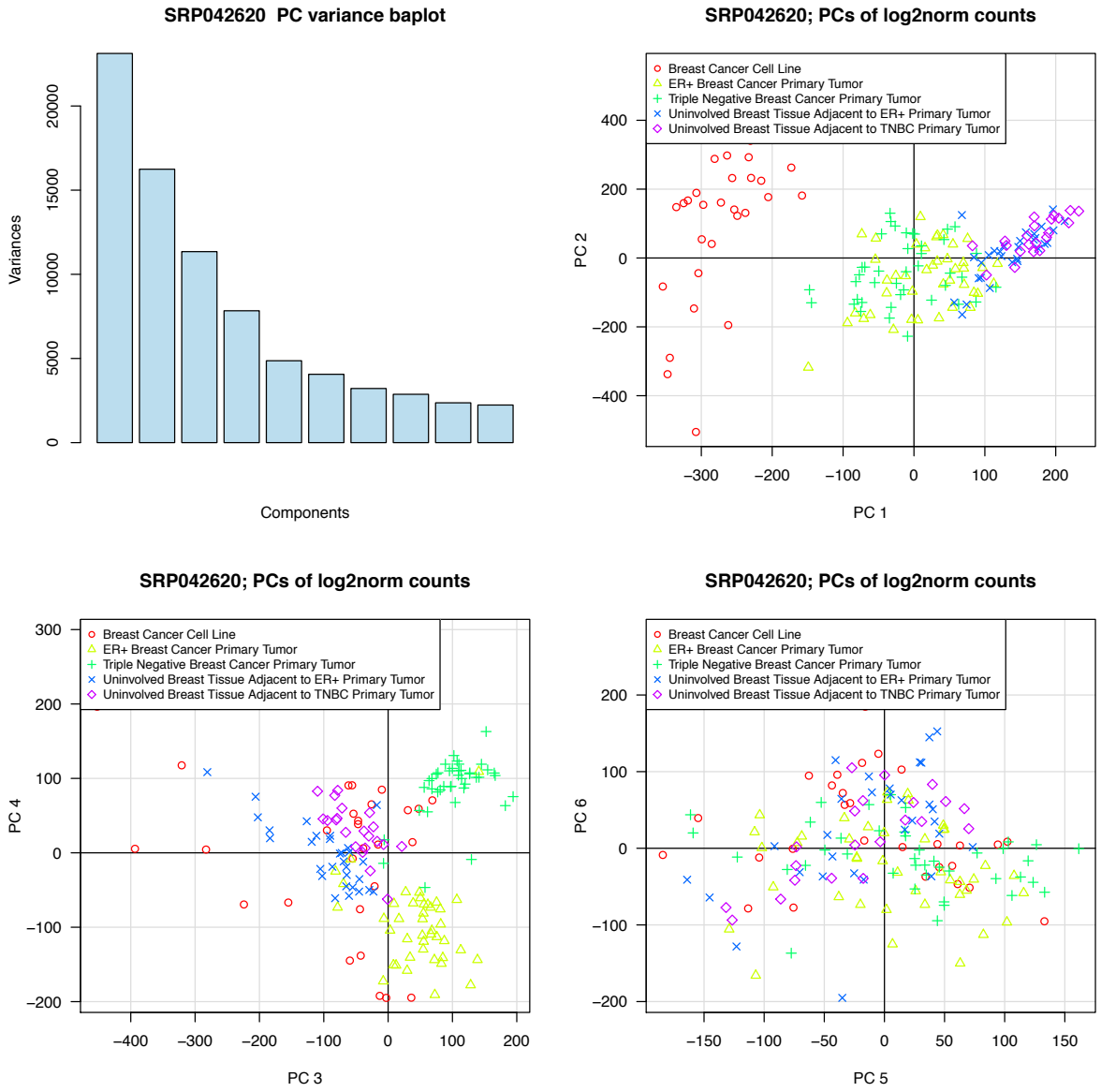


Figure 14. PCs plots of the *breast cancer* study (SRP042620).
Top-left: variance of the 9 first components. **Top-right:** first and second components. **Bottom-left:** third and fourth components. **Bottom-right:** fifth and sixth components.

PC plots of the Psoriasis study (**Figure 15**) clearly show that the variance of the first 9 components is mainly captured by the first 2 PCs. This suggests that these PCs from 1 to 6 are not able to separate the classes of PCs. The top-right panels visibly confirm into the given fact PC1 alone or PC2 alone are unable to decouple and classes from two, but that the combination of both PC1 and PC2 completely segregates between the *lesional psoriatic skin* and *normal skin* samples. In contrast, all of the subsequent PCs (PC3, PC4, PC5, and PC6) are unable to decouple the two classes.

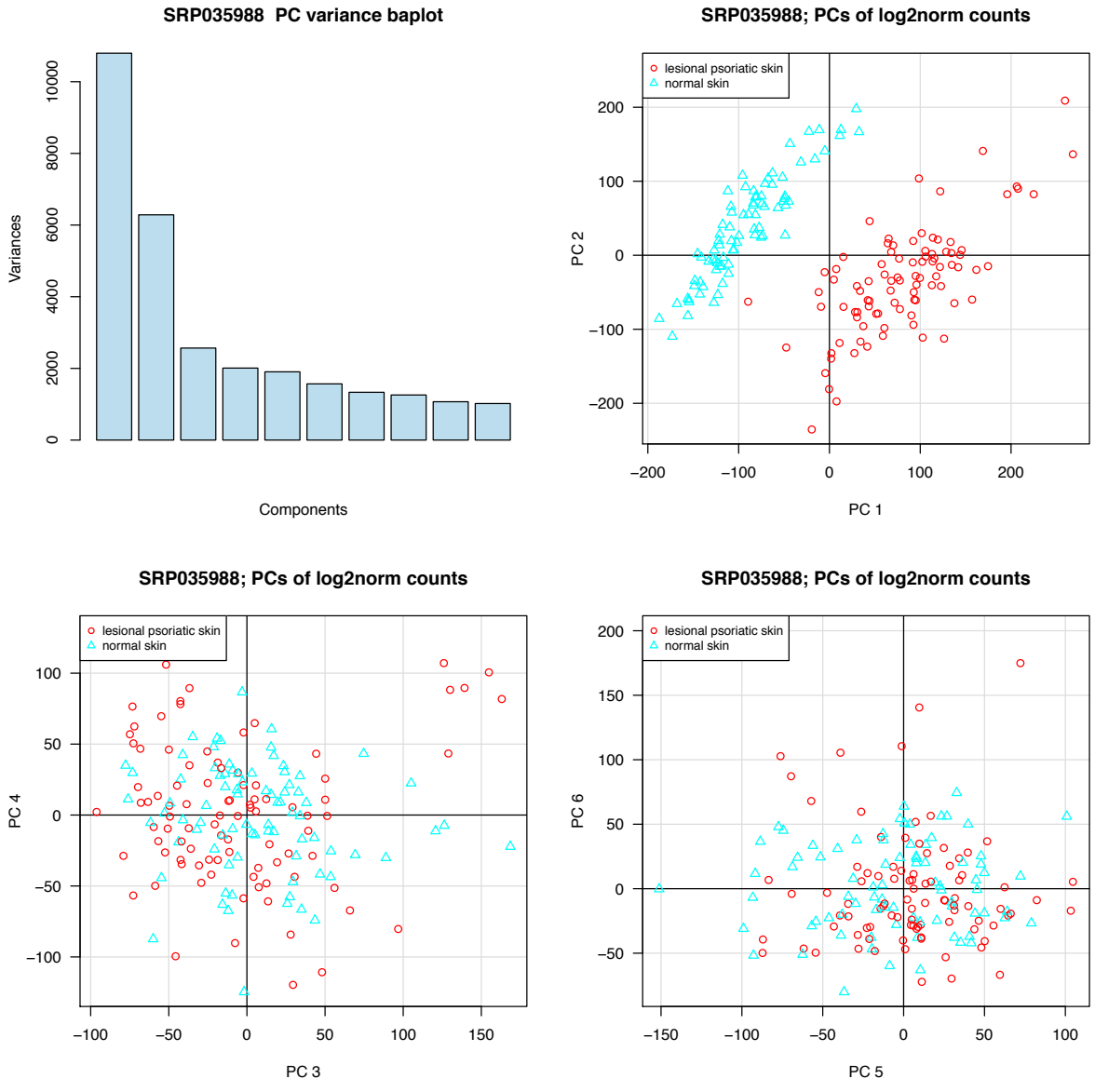


Figure 15. PC plots of the *Psoriasis* study.
Legend as in **Figure 14**.

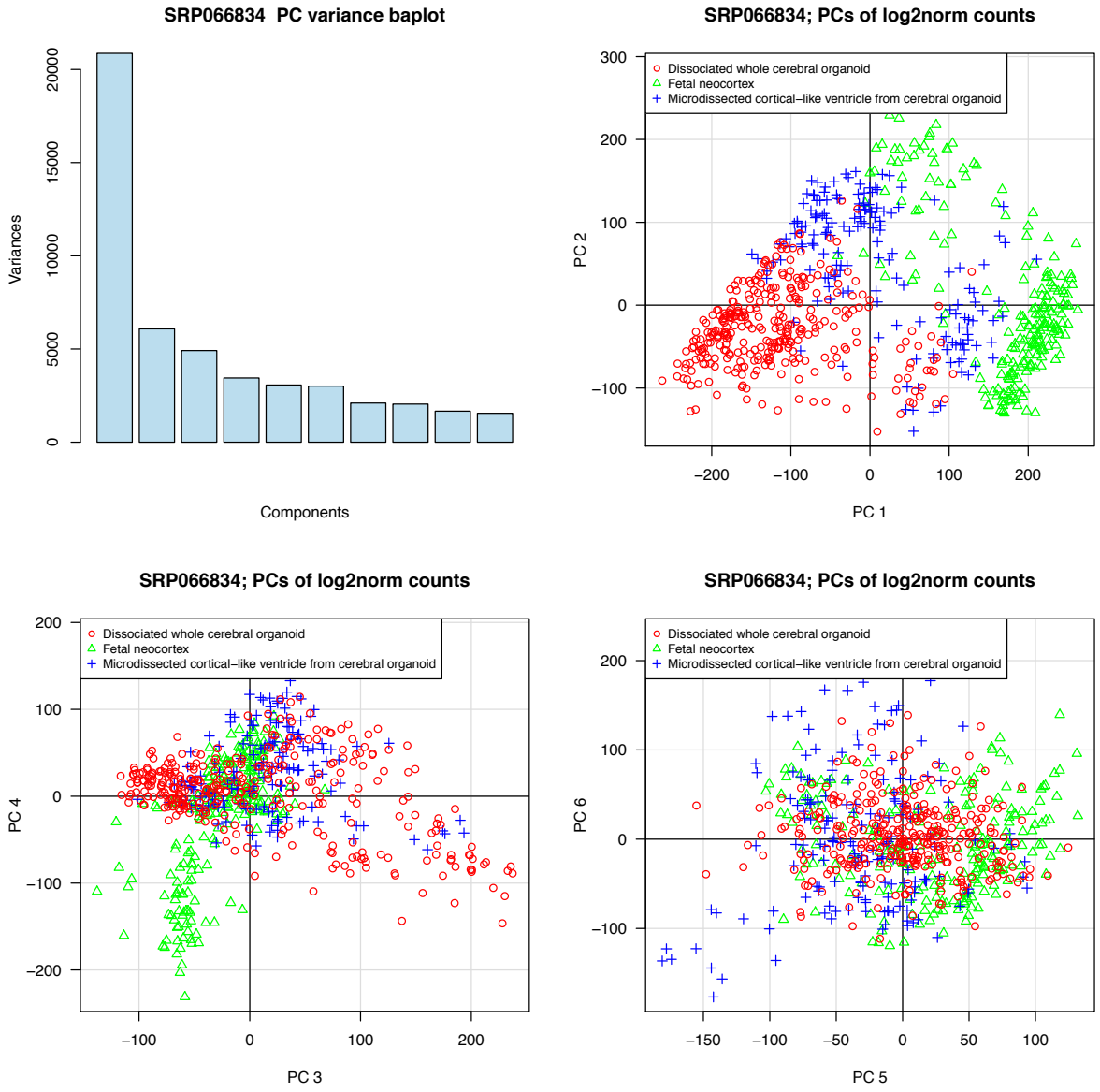


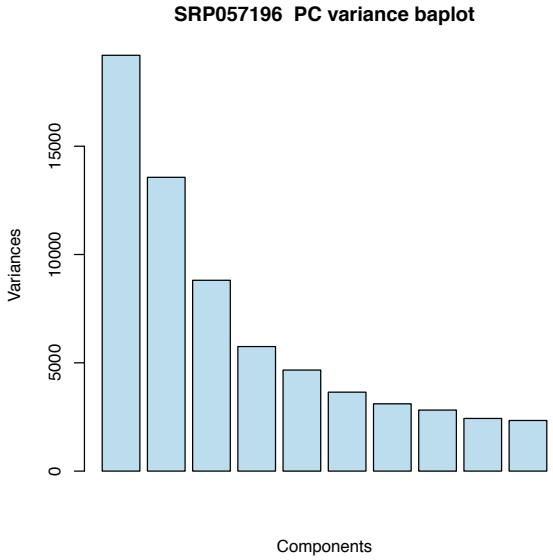
Figure 16. PC plot of the cerebral organoids and foetal neocortex study (single-cell).
Panel legends: see **Figure 14**.

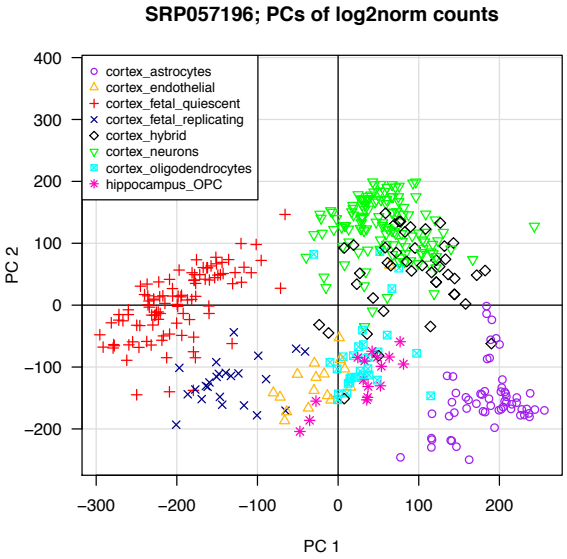
The single-cell RNA-seq for the *Cerebral organoids and foetal neocortex* study is shown in **Figure 16**. We observed a drop between PC1 and the other PCs, showing that it already captures an important part of information. The PCs do not successfully segregate all of the cells, but some separation is noted, in particular:

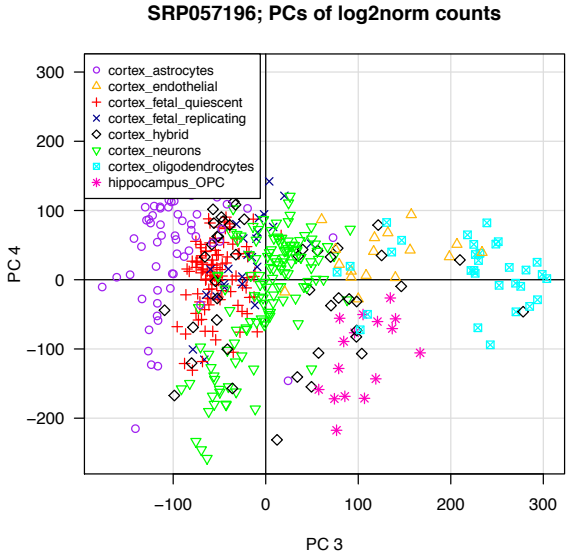
- PC1 + PC2 separate quite well – but imperfectly – the green, red and blue dots
- PC4 highlights a subset of green dots, which might represent an ignored subclass of the foetal neocortex (interesting)

- PC5 and PC6 do not segregate much; everything seems intermingled, although there are more green dots on the right side (thus, PC5 partly distinguishes green dots).

None of the PCs from 1 to 6, nor the pairwise combinations (PC1+PC2; PC3+PC4; PC5+PC6), succeed in perfectly segregating the classes. Some pairwise PC plots, however, do show a pretty good separation of some particular classes from others.







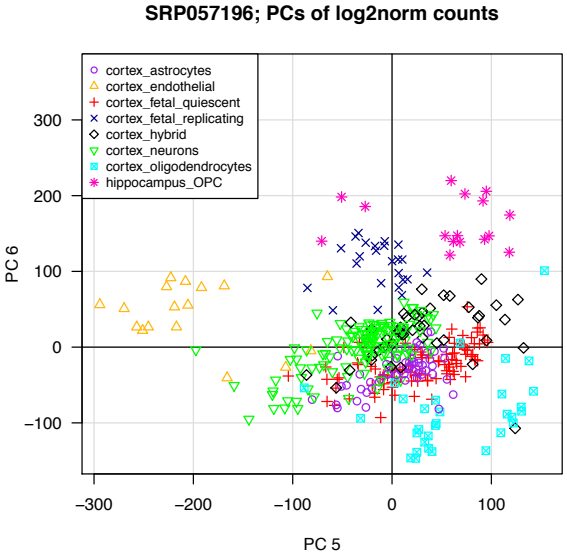


Figure 17. PC plots for cell types in the *healthy human brain* study.
Legend as in **Figure 14**.

Through **Figure 17**, we can easily note there is partial segregation of the 8 classes from the healthy human brain, wherein:

- PC1-PC2 show clear colonies for each class, but it is not completely segregated; we can easily notice aggregations for each class in the cellular human brain. Moreover, the combination of PC1-PC2 may somewhat segregate for a number of classes; for instance, decouple red balls from the remainder (green, black, purple, yellow, indigo, turquoise).
- PC3-PC4 highlights a subset of hippocampus dots, which might represent an ignored subclass of the human brain (interesting).
- PC5 and PC6 do not segregate much; everything seems intermingled, although there are more yellow dots on the left side (thus, PC5 partly distinguishes yellow dots).

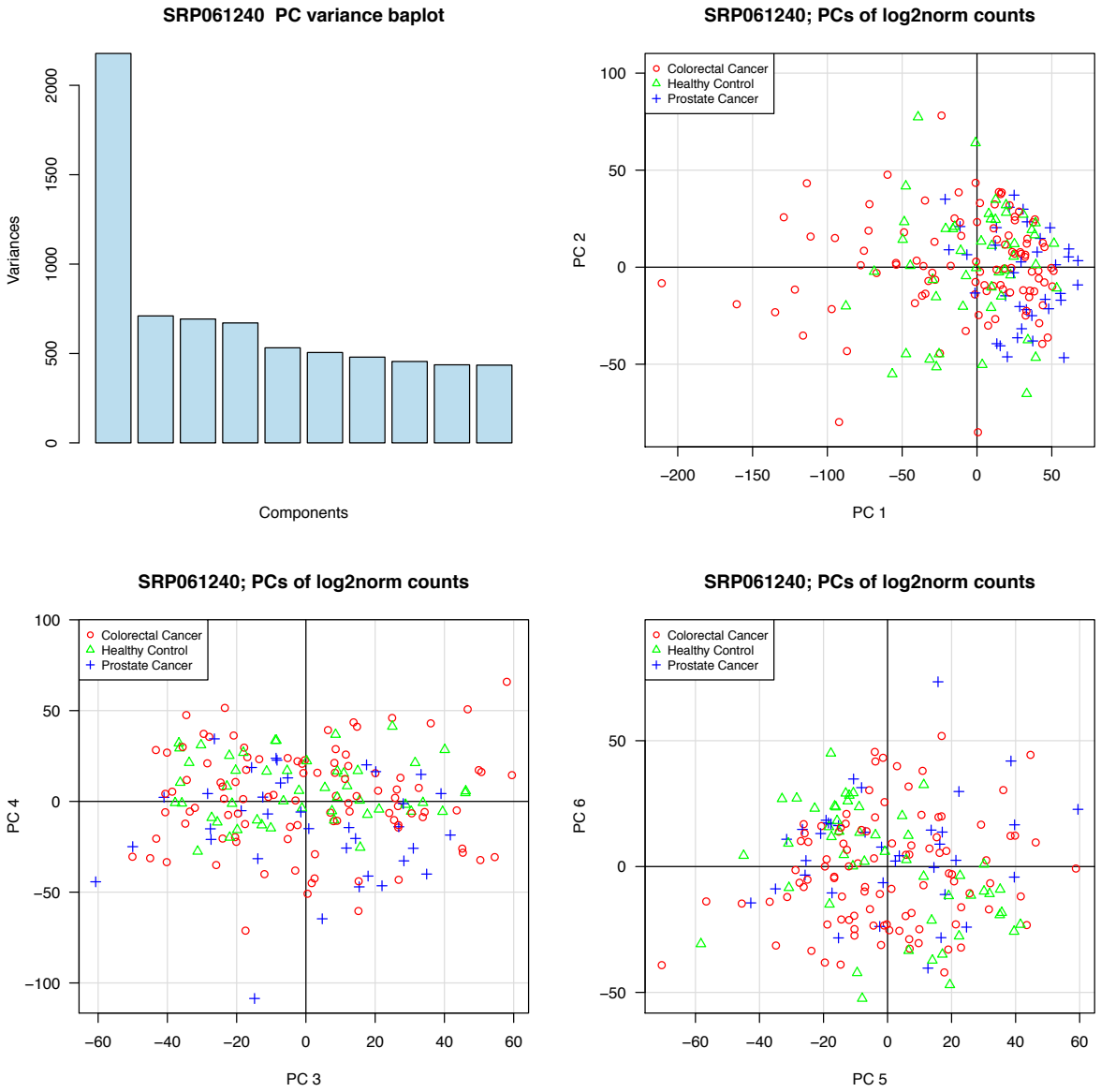


Figure 18. PC plot for the *Plasma extracellular vesicles for the Cancer disease types* study (SRP061240). Legend as in **Figure 14**.

Figure 18 is concerned with the three plasma extracellular vesicles classes.

- It is clear that all classes are mixed up for PC1-PC2. Moreover, PC1-PC2 is not able to segregate any classes; in addition, the combination also could be capable of decoupling any class from 3.
- PC3-PC4, and PC5-PC6 have the same behaviour as PC1-PC2; furthermore, their respective combination does not have any ability to obviously separate any class. This

led us to quickly notice that all samples may have the same properties, which made it difficult to decouple any classes.

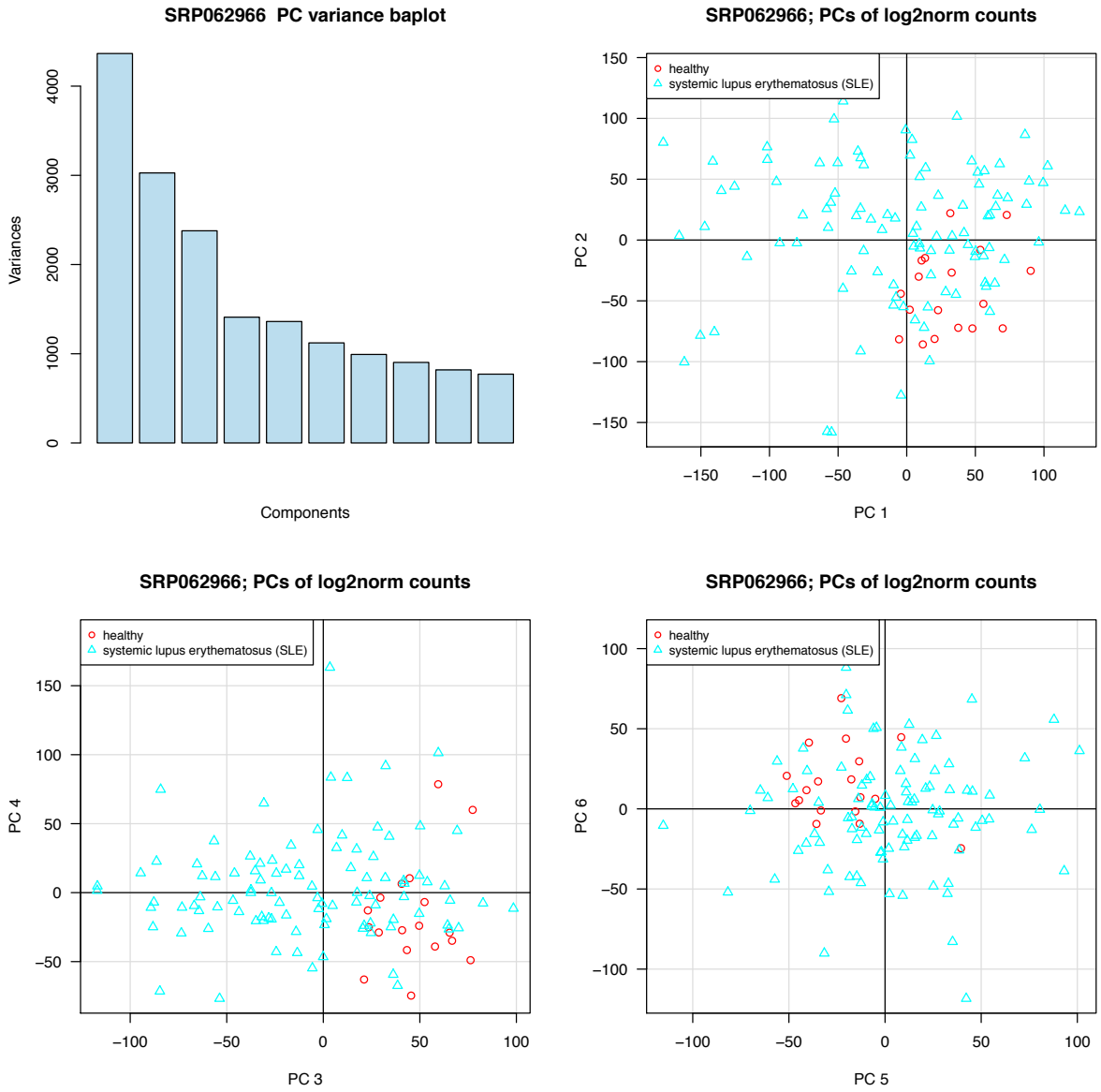


Figure 19. PC plot for the *Systemic lupus erythematosus* (SRP062966) study.
See **Figure 14** for the legend.

Figure 19 gives first impressions about the easy task of separating two classes.

- PC1-PC2 partly segregates the healthy class from systemic lupus erythematosus cases.
- PC3-PC4-PC5-PC6 has the same effect of decoupling the two classes.

- Besides that, the combination was partially capable of segregating two classes.

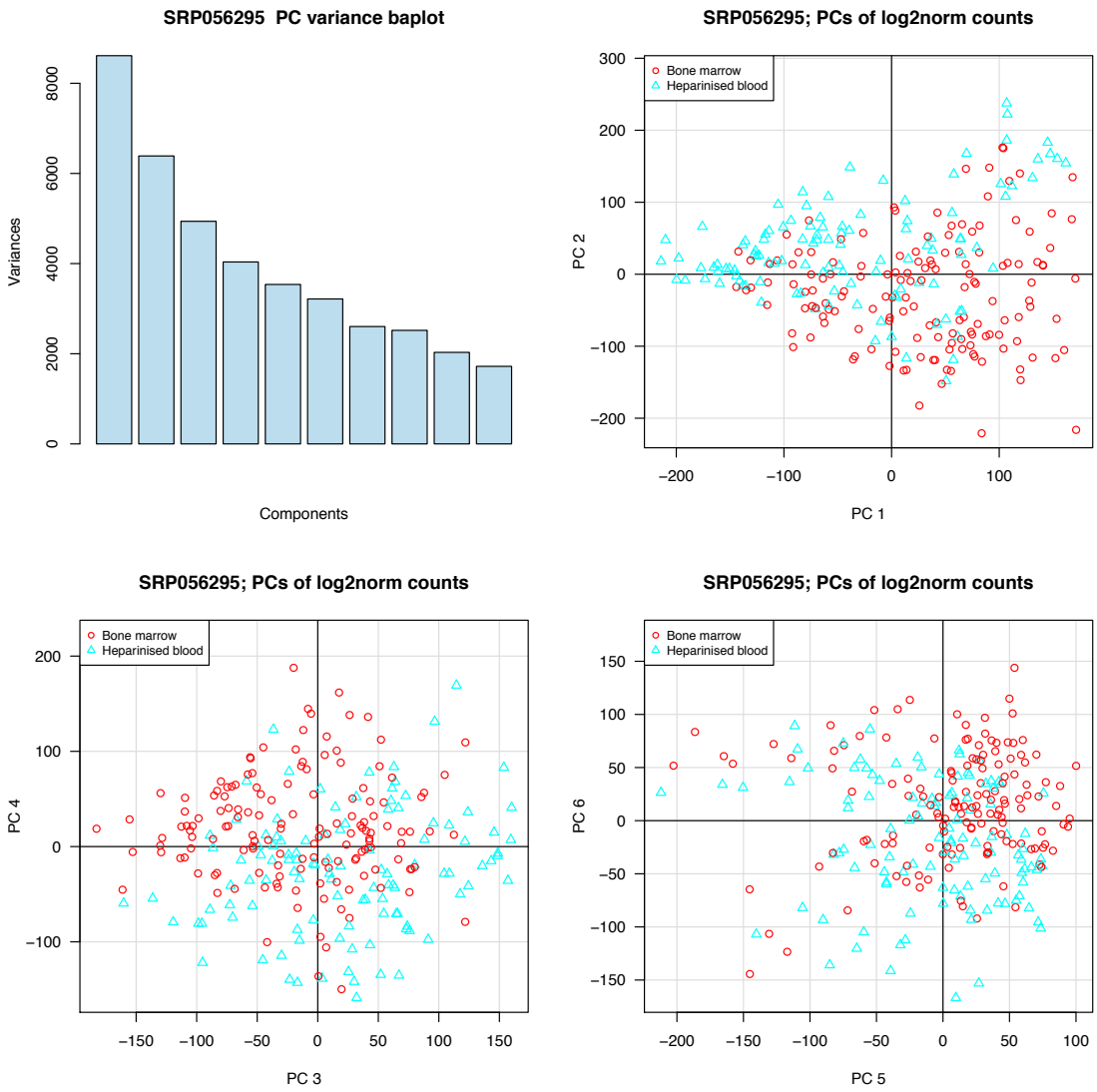


Figure 20. PC plot for the *Human acute myeloid leukaemia (AML)* study.
Legend is the same as for **Figure 14**.

Figure 20 presents the PCA to separate the two classes of human acute myeloid leukaemia. Wherein:

- PC1-PC2 has the same effect as that of the PCs in **Figure 19**. The combination line is important here.
- The PC3-PC4-PC5-PC6 combination has no ability to decouple any classes; moreover, the combination does not have the capacity to segregate any classes.

CHAPTER 5: ASSESSMENT OF CLASSIFIER PERFORMANCES ON RNA-SEQ DATA

Data classification can be divided into binary (each sample belongs to one among two classes), multiclass (each sample belongs to one among more than 2 classes) and multi-labelled classification (each sample can belong to one or more classes) (Ranawana and Palade, 2006). This study focuses on the multiclass problem and relies on the *Misclassification error rate* (*MER*) as a metric for evaluating the effectiveness of the supervised classifiers.

We will perform a comparative assessment of 3 classification methods (KNN, RF and SVM), and study the impact of pre-processing and library size normalisation on their performances.

Principles of the evaluation procedure

Sampling procedure

For each analysis we ran a repeated sub-sampling validation with 10 iterations of the training/testing procedure. We applied a stratified sub-sampling mechanism in order to ensure fair training and testing (each class should be represented in similar proportions in the training and testing sets).

The main idea here was to ensure that each class is composed of a sufficient number of samples for each class category, leading to fairer evaluation for the development of an effective supervised classifier with a selected dataset. The underlying idea for such sampling procedures is to ensure that not all samples come from the same class. If that happened, the evaluation procedures would not be fair; moreover, the assessment would be biased to the class with the majority of samples.

Evaluation metrics for classifier performances

Evaluation metrics play a critical role in achieving the optimal classifier during the classification training. Thus, the selection of a suitable metric is an important key for discriminating and obtaining the optimal classifier for the analysis of RNA-seq data.

From the literature, the evaluation metrics can be categorised into three types: threshold, probability and ranking metrics (Caruana and Niculescu-Mizil, 2004). Each of these types of

metrics evaluates a classifier with different aims. In practice, the threshold and ranking metrics were the most common metrics used by scientists to measure the performance of classifiers.

In most cases, these types of metrics can be employed in three different evaluation applications (Lavesson and Davidsson, 2008): first, the generalisation ability of the trained classifier, when tested with the unseen data; second, using the evaluation metrics as an evaluator for model selection, wherein the evaluation metric task is to determine the best classifier among different types of trained classifiers which focus on the best future performance when tested with unseen data; and third, using evaluation metrics to discriminate between and select the optimal classifier from all of the generated classifiers during the classification training. In other words, only the best classifier which is believed to indicate the optimal model will be tested with the unseen data.

For the first and second application of evaluation metrics, almost all types of threshold, probability and ranking metrics could be implemented to evaluate the performance and effectiveness of classifiers. It should be noted that most of the existing metrics are defined in the context of 2-group classification, and are thus irrelevant to our problem, which is to evaluate classifiers on multi-group datasets. Only a few types of metrics could be utilised to discriminate and select optimal classifiers during the classification training.

Accuracy (or its complement named "**misclassification error rate**", abbreviated **MER,** or "error rate") is one of the most common metrics used to evaluate the generalisation ability of a classifier. Accuracy is defined as the proportion of cases that are correctly predicted by the trained classifier when tested with unseen data. It is thus the complement of the misclassification error rate, which is defined as the proportion of misclassified cases (see Materials and Methods for the formulae).

It has to be noted that using the accuracy or MER as a benchmark measurement has a number of limitations. The accuracy has a poor value to establish an informed decision in order to build an appropriate classifier (Huang and Ling, 2005). Indeed, all misclassification errors are considered equivalent, irrespective of the class on which they occur. In particular, for binary classification problems, MER makes no distinction between false positive and false negatives.

Ranawana and Palade (2006) and Wilson (2001) demonstrated that the simplicity of accuracy could lead to suboptimal decisions, especially when dealing with imbalanced class distribution. For our evaluation of RNA-seq-based classification, most studies have a fair balance between class sizes (**Figure 10**), but this is not the case for two of them: *Cellular complexity of adult and foetal human brain* (SRP057196) and *Lupus* (SRP062966).

Since MER is the most widely used metric in the domain, we adopted it for this study, but we took care to systematically compare the observed MER with that expected at random, *E(MER)*, which can be computed theoretically from class frequencies. We also compared the actual MER to that obtained with permutated class labels, which corresponds to an untrained classifier, and can be considered, to some extent, as an empirical estimation of the expected MER.

Library size normalisation

Given the strong variations in library sizes, as exemplified in **Figure 20** (i.e. sequencing depth), the simplest way to standardise this is to standardise raw read counts by multiplying them using a scale reflecting library sizes.

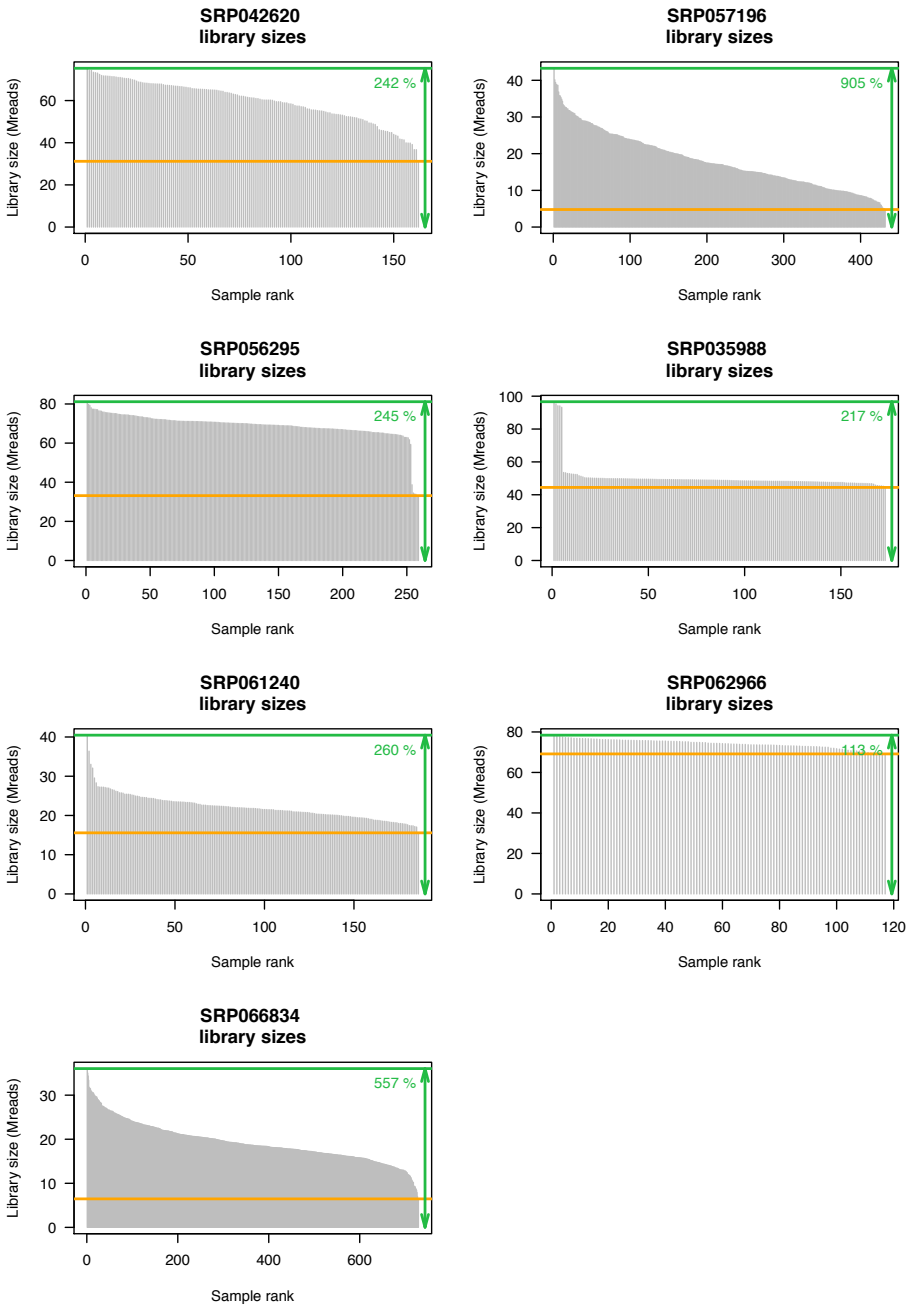


Figure 21. Variation of library sizes for the 7 studies.
On each plot, the ordinate indicates library sizes (in million reads) of the samples sorted by decreasing sizes (abscissa). The horizontal bars show the maximal (green) and minimal (orange) library sizes. The percentage indicates the ratio between maximal and minimal library sizes for the considered study.

We consider four different methods for calculating these scaling factors, as described below.

1) **Filtered raw counts** are non-standardised raw count after we have removed all the genes that have a zero variance, as well as the NA-containing rows (genes).

2) In the **third quartile** method, gene counts of each sample are divided by the third quartile of the counts for all the genes in this sample and multiplied by third quartile of counts different from 0 in the computation of the normalisation factors (Bullard et al., 2010b)
3) **Trimmed mean of M-Values** (**TMM**) is used by (Robinson and Oshlack, 2010b) and is implemented in the edgeR Bioconductor package. It is based on the hypothesis that most genes are not differentially expressed (DE). Wherein its value provides an estimate of the correction factor that must be applied to the library size (and not the raw counts) in order to bring to a comparable level all the counts that a given genes have in different samples The calcNormFactors() function in the edgeR Bioconductor package implements these scaling factors. To obtain normalised read counts, these normalisation factors are re-scaled by mean of the normalised library sizes. Some normalised read counts are obtained by dividing raw read counts by the above-mentioned re-scaled normalisation factors.
4) **DESeq2** it is the normalisation method (Anders and Huber, 2010c) implicated in the DESeq2 Bioconductor package (Anders and Huber, 2010c), which relies on the hypothesis that most genes are not differentially expressed (DE). A DESeq2 scaling factor for a given specimen is computed as the median of the ratio for each gene of its read count over its geometric mean across all specimens. The key idea is that non-DE genes should have similar read counts across samples, leading to a ratio of 1. Assuming that most genes are not differential expressed (DE), the median of this ratio for the specimen provides an estimate of the correction factor that should be applied to all read counts of such specimen to fulfil the hypothesis. We applied DESeq2 normalisation utilising the estimateSizeFactors() and sizeFactors() functions in the DESeq.
5) **Relative Log Expression** (**RLE**) implemented in the DESeq2 package (Anders et al., 2013), (Anders and Huber, 2010c) and (Love et al., 2014). We actually computed RLE normalisation Factors with edgeR, since their calcNormFactors() function supports this method.

Feature types used to assess classifier performances

Our work concentrated on testing different normalisation methods which are third quartile, TMM, RLE and DESeq2. Moreover, we tested log2 transformation and PC reduction to analyse the effectiveness of the normalisation methods to refine the efficiency of the supervised classifiers. We focused on the effect of different normalisation methods besides the log2 transformation and PC reduction to refine the accuracy of the supervised classifier.

With the different combinations of normalisation options, we obtained 12 types of features:

1. **Filtered_counts**: raw counts remaining after class and gene filtering.
2. **third quartile**: library-size standardization based on the third quartile as scaling factor.
3. **q_0.75_log2**: log-2 transformation of the third-quartile (= quantile 0.75) scaled counts.
4. **q_0.75_log2_pc**: principal components derived from q_0.75_log2
5. **TMM**: library-size standardization based on trimmed mean values.
6. **TMM_log2**: log2-transformed TMM values
7. **TMM_log2_pc**: principal components of the TMM_log2
8. **RLE**: relative log expression
9. **RLE-pc**: principal component transformation of RLE values
10. **DESeq2**: normalised counts computed by the DESeq2
11. **DESeq2_log2**: log2-transformation of DESeq2-normalised counts
12. **DESeq_log2-pc**: Principal components from DESeq2_log2

We also analysed these 12 feature types with the classifiers trained with the original class labels or permutated class labels, respectively. We focused on studying the impact of each normalisation method on classifier accuracy.

Evaluation of classifier performances

In this section, we separately analysed each classifier (KNN, SVM and Random Forest) to identify the most suited pre-processing procedure. We chose SVM, KNN and Random Forest because these three methods rely on very different principles and assumptions for their classification: KNN relies on distances in Euclidean space, Random Forest on decision trees with separate decisions on the different features, and SVM on hyper margins established with different types of kernels.

We started with a detailed description of some representative studies. We then performed a comparison of the performances between classifiers and generalised these to the 7 studies.

Breast cancer study (SRP042620)

We started by exploring whether or not the performance of targeted classifiers is affected by the type of features resulting from our different data normalisation methods.

Figure 22 summarises the performances of the three targeted classifiers (SVM, KNN and RF) with the 12 feature types defined above, with either correct or permutated class labels.

SVM. The top-left panel of **Figure 22** shows the boxplots of misclassification error rates (MER) for 10 iterations of training/testing evaluation for the SVM classifier. The first observation is that all standardization methods (q0.75, TMM, RLE and DESeq2) give more or less the same results as the non-standardized raw counts (filtered_counts), with a MER near 12%. However, after we performed log2 transformation, the MER drops to ~5%. A PCA transformation of the log2-transformed counts does not bring any further improvement over the simple log2-transformed data. For all the 12 feature types, the SVM trained with permutated class labels (grey boxes) returns a very high MER (~80%) corresponding to the theoretical random expectation (blue dotted line).

KNN. The top-right panel in Figure 22 summarizes the performance of the KNN with four targeted normalisation methods. With this classifier, the performances are rather poor on the raw and normalised counts (~45% MER). The log2 transformation brings some gain in accuracy (the MER decreases to ~20%), and the PC transformation gives similar results as the log2 counts. The training with permutated labels gives the same MER as for SVM, and it fits the random expectation. The poor classification rate and the relative improvement brought by log2 transformation are consistent with the fact that KNN relies on Euclidian distances computed in the feature space, which are sensitive to outliers (whose effect is reduced by log2 transformation).

RF. In contrast with SVM and KNN, Random Forest classifier achieves rather good results with raw and normalised counts, and the log2 transformation does not bring any improvement. This is consistent with the fact that tree-based methods are insensitive to monotonic transformations of the data. Interestingly, the PC transformation clearly decreases the performances of RF classifiers. We interpret this as an effect of the interaction between the concentration of the relevant information in the first components, and the random sampling of features underlying the RF algorithm: at each RF iteration, a tree is trained with a random subset of the features, and if those are PCs, the top-ranking (and supposedly most informative) features may be included or not in the random selection, thereby leading to fluctuations in the efficiency of the training.

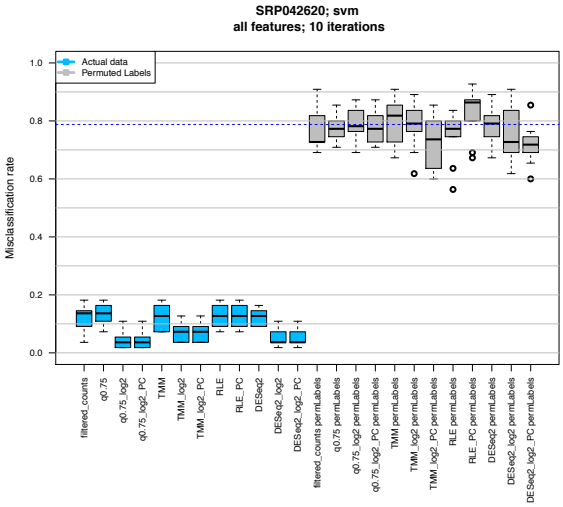
In summary, we confirm here the key principle that some authors adopted it for analysis microarray data (Anders and Huber, 2010a; Robinson and Oshlack, 2010a): they shown the ability of the log2 transformation (and PCA) to improve the visualisation of the structure of RNA-seq data. Our study further shows the influence of log2 transformation to improve from the efficiency of classification process. It turns out there is a tangible effect to log2 transformation to improve the efficiency of classifiers in classification the samples based on their RNA-seq transcription

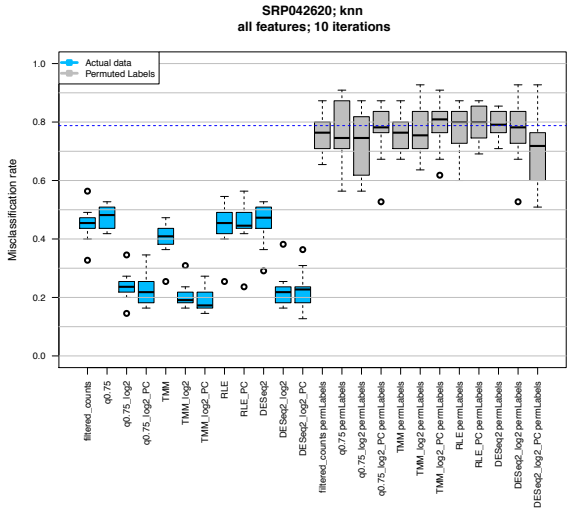
profile, that is main goal from the preliminary investigation about the optimal procedure of pre-processing.

Where the optimal normalisation method depends on a principle classifier:

- For the Breast Cancer study, all classifiers performed well with the trained sets and were much better than those of the training classifier with permutated class labels; among them, RF performs better than SVM, which performs better than KNN, which performs rather poorly.

- We can briefly notice that the TMM standardisation method is better than the others.
- It always better to log2-transform the data before utilising the supervised classification.
- Using the PC transformation and log2 data provide the same advantage of improving the efficiency of classifiers.
- The performance of RF is different to that of other classifiers, as it relies on the ensemble approach; with this classifier, log2 transformation does not bring any improvement. Furthermore, with this classifier, PCA decrease the performance (whereas for other ones the PC-log2 has the same performances as the simple log2 data).





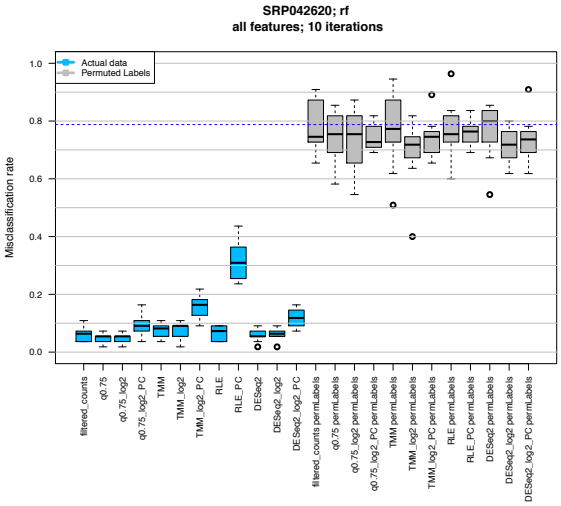


Figure 22 Performance of classifiers measured by misclassification error rate for the Breast cancer study (SRP042620).

The ordinate indicates the misclassification error rates. Each boxplot corresponds to one testing-training experiment (10 iterations) and one particular pre-processing method (from left to right: filtered counts, third quartile(q0.75), q0.75_log2, q0.75_log2_PC, TMM, TMM_log2, TMM_log2_PC, RLE, RLE_PC, DEseq2, DESeq2_log2, DESeq2_log2_PC). **Blue boxes:** results of the analysis of the actual datasets. **Grey boxes:** random expectation estimated by permuting class labels during the training and testing. **Dotted line:** theoretical value of the random expectation for the MER, based on class sizes. **Top left:** Support Vector Machines (SVM); **top right:** K-nearest neighbours (KNN); **bottom left:** Random Forest (RF).

Human Acute Myeloid Leukaemia study (SRP056295)

For the sake of fairer evaluation of the classifiers, analysis of the human acute myeloid leukaemia (AML) study data set (SRP056295) was performed, which contains 4 classes (Bone marrow, Heparinised blood, EDTA blood and Leukapheresis), with an overall number of samples of 263 and the number of features (genes) of 58,037 before filtering.

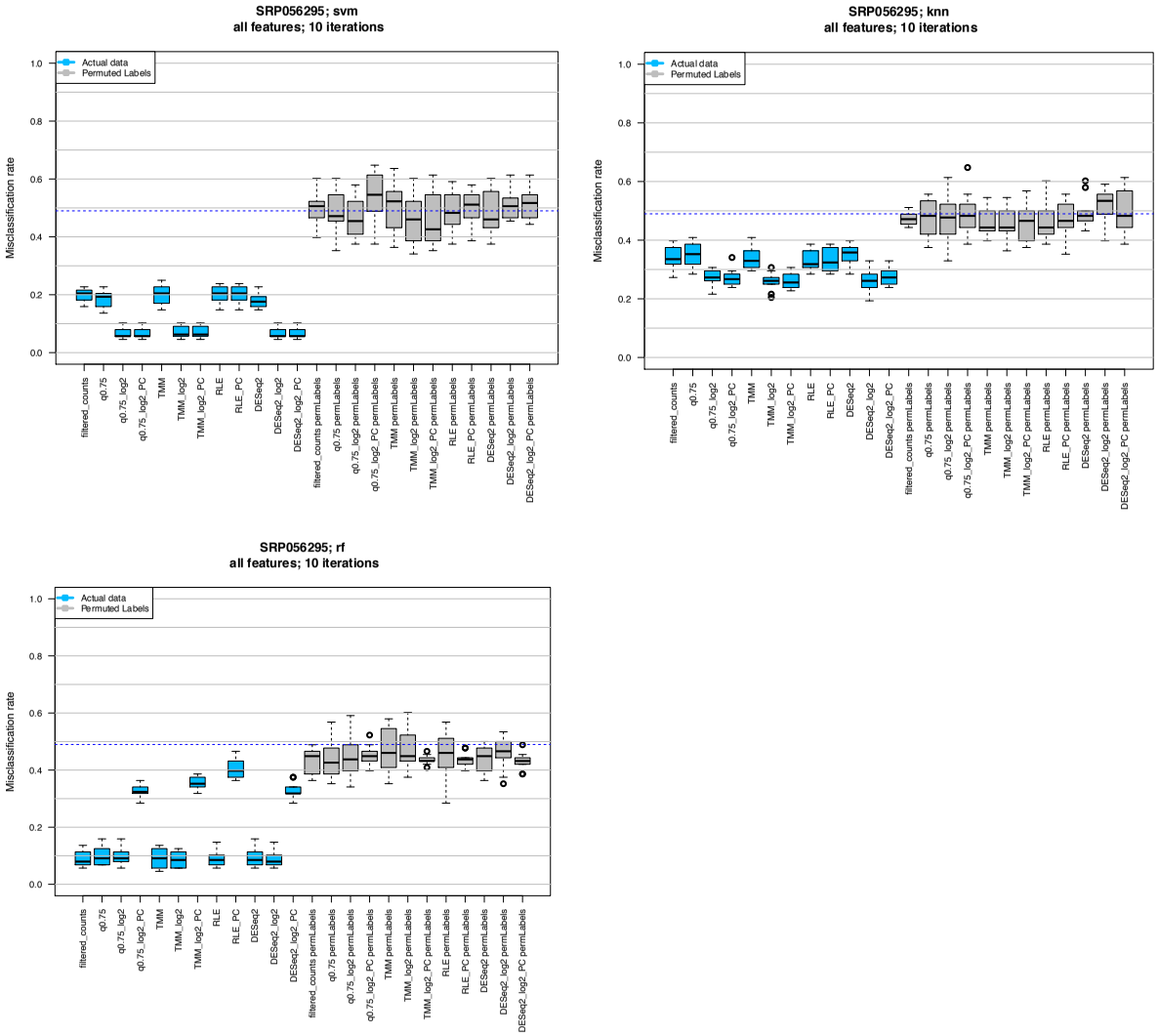


Figure 23. Performance of classifiers measured by misclassification error rate for the *human leukaemia* study (SRP056295).
Legends as in Figure 22.

Figure 23 confirms the behaviour of tested classifiers with the 12 previously defined feature types from another study (*Human leukaemia*): for KNN and SVM, there is no obvious impact of the normalisation method (3rth quantile TMM, RLE, DESeq2), but log2-transformation strongly improves the results, whilst further PC transformation neither improves nor reduces the accuracy. Random forest shows a markedly different behaviour: their performances are similar irrespective of the normalisation method and log2 transformation, but PC transformation strongly increases the error rates (this effect is even stronger than for the Breast cancer study seen in Figure 22).

Cerebral organoids and foetal neocortex study (SRP066834)

Figure 24 shows the results from the analysis of the *Cerebral organoids and foetal neocortex* dataset (SRP066834), which contains 3 classes (*Dissociated whole cerebral organoid*, *Foetal neocortex*, *Microdissected cortical-like ventricle from cerebral organoid*), with an overall number of 729 "samples" (actually single cells). With this cerebral organoids and foetal neocortex case, SVM behaviour clearly shows the impact of the log2 transformation (and the derived PC), which returns 0% of the MER. The KNN and RF give the same findings, as in the previously discussed Human leukaemia dataset **Figure 24**, but their performances are clearly inferior to SVM. KNN achieves the best results (MER=11%) with log2-transformation after DESeq2 normalisation, and RF returns ~7% MER with log2-transformed data, irrespective of the normalisation method use, but shows a marked increase in MER with PC-transformed data.

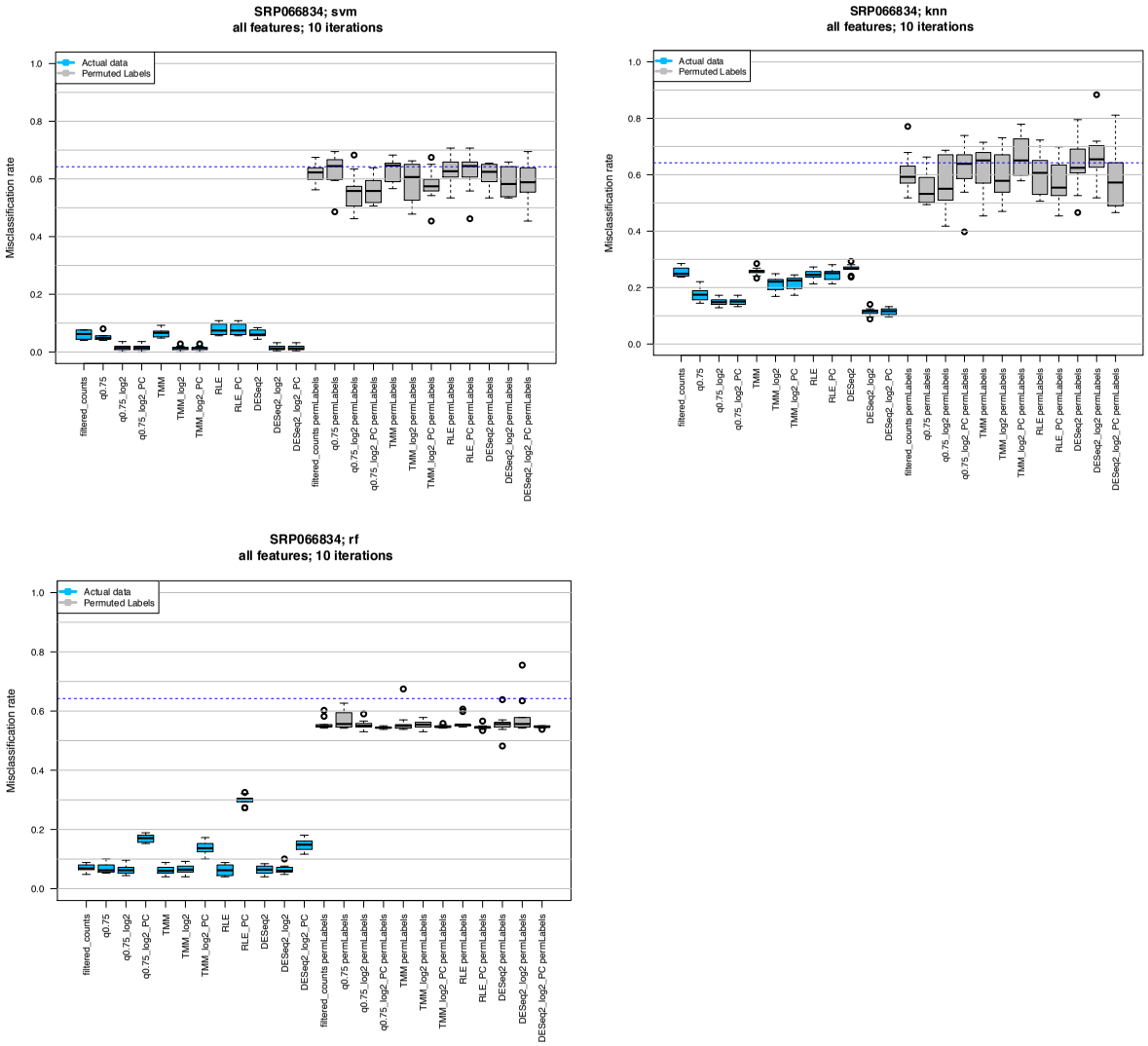


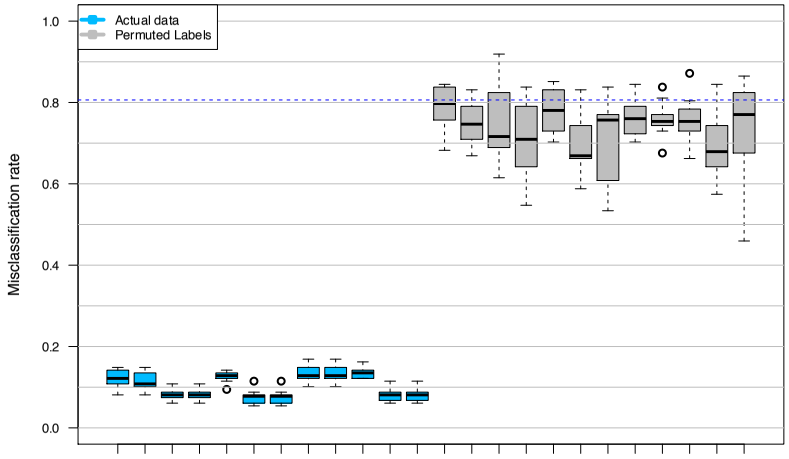
Figure 24. Assessing the performance of classifiers through by misclassification error rate for the *cerebral organoids and foetal neocortex* study (SRP066834).
Legends as in Figure 22.

Adult and foetal human brain study (SRP057196)

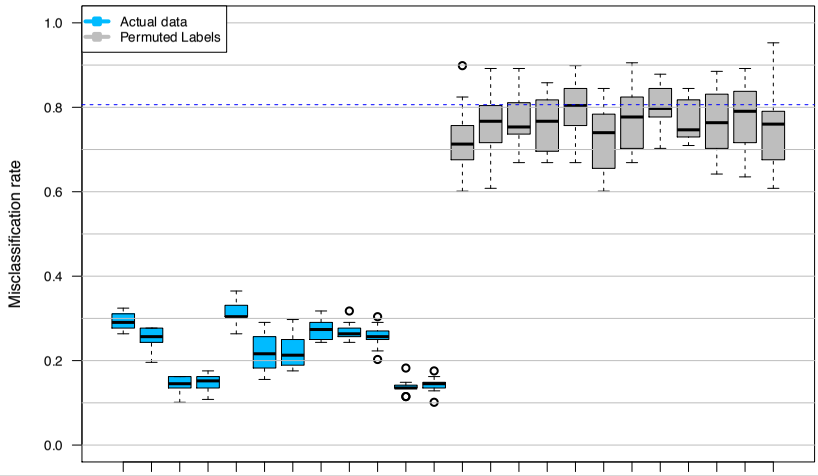
In the single-cell dataset *Adult and foetal human brain* (SRP057196), the three classifiers show the same trends as for the previous cases: the log2 provides a strong improvement for SVM and KNN but not for RF, and the PC transformation of log2-counts has no impact on SVM or KNN but deteriorates the performances of RF. Here as well, SVM outperforms KNN and RF.

Figure 25. Assessing the performance of classifiers through by misclassification error rate for the *Adult and foetal human brain* study (SRP057196). Legends as in Figure 22.

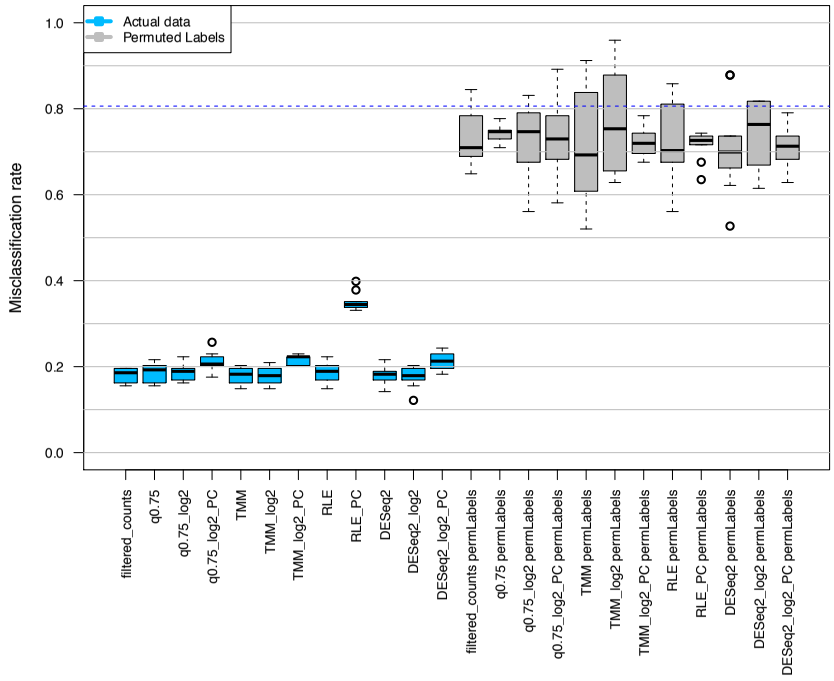
SVM



KNN



RF



Psoriasis study (SRP035988)

With the *Psoriasis* dataset (SRP035988) the situation is quite different: SVM returns 0% MER with all 12 features types, and KNN achieves results that are as good, but only with log2-transformed counts. RF gets the same perfect classification irrespective of log2 transformation, but its MER increases with PC transformation. These remarkable performances suggest that the transcriptome profiles of these 2 classes differ so much that it is easy for any classifier to discriminate them perfectly, almost irrespective of the chosen pre-processing procedure.

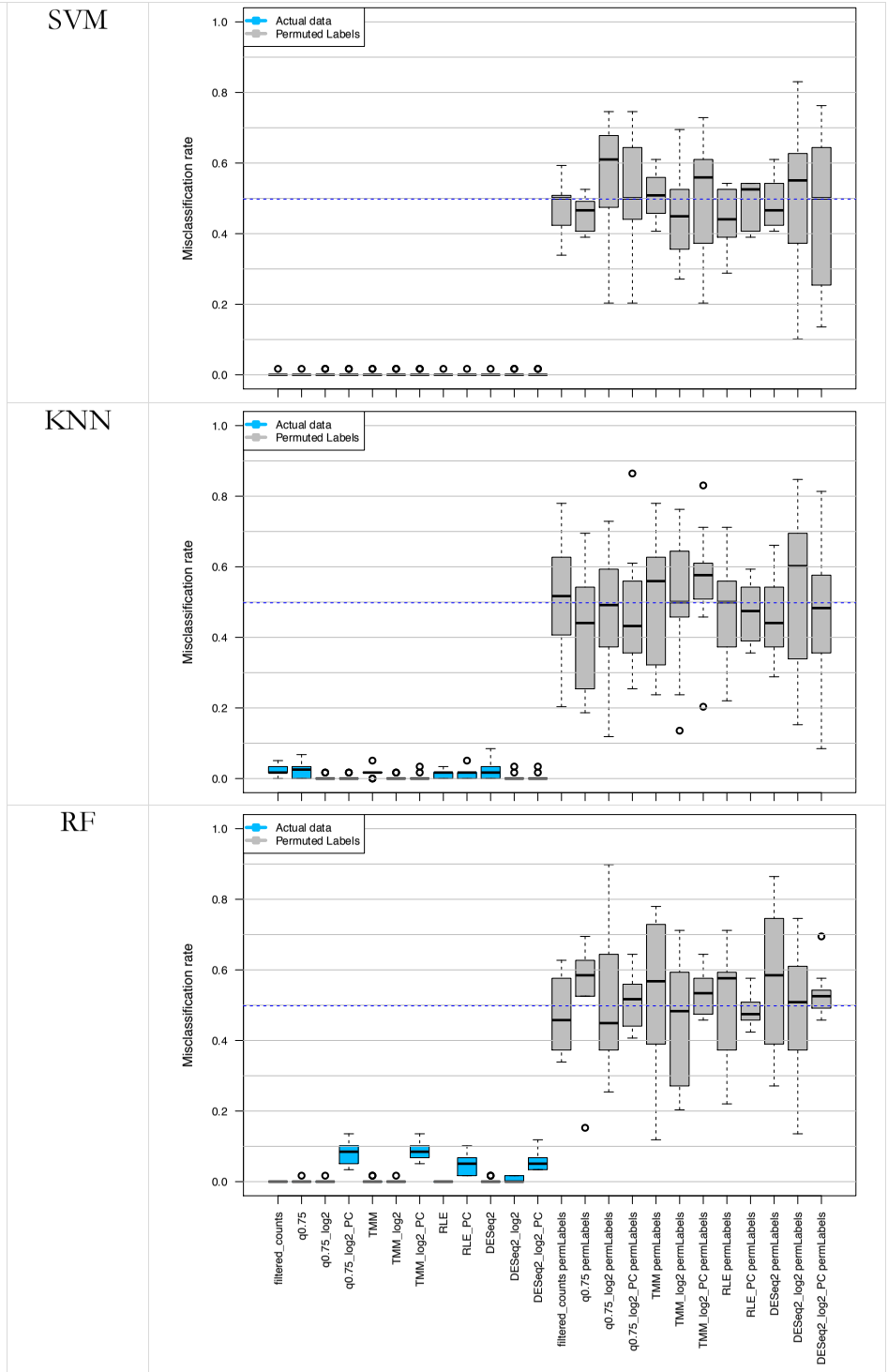


Figure 26 Impact of Normalisation on the misclassification error rate of classifiers on the *Psoriasis* study (SRP035988). Legends as in Figure 22.

Lupus erythematosus study (SRP062966)

With the *Lupus erythematosus* study (**Figure 26**), an astonishing observation is that all of the classifiers return better results with permutated class labels (grey boxplots) than the MER expected at random (dotted line), thereby suggesting that the classifiers were able to learn something with randomly permutated class labels. This improvement is observed for almost all of the classifiers, and with all of the feature types (except for SVM with non-log2 transformed counts). Note that the random expectation for this study is rather low (~27%), due to the strong imbalance between the two classes. In such conditions, a trivial strategy to achieve good performances is to assign all samples to the major class, which ensures better results than a random assignment balanced by the prior class frequencies. However, there must be some other effect playing here, since we observe the same effect of learning from the permutated labels in the next study, although the classes are balanced. We still ought to investigate the reason for this surprising capability of the 3 classifiers to learn from the permutated class labels.

Despite this, all of the classifiers achieve better results with the actual class labels (blue boxplots) than with the permutated ones (grey boxplots), thereby showing that the training is effective.

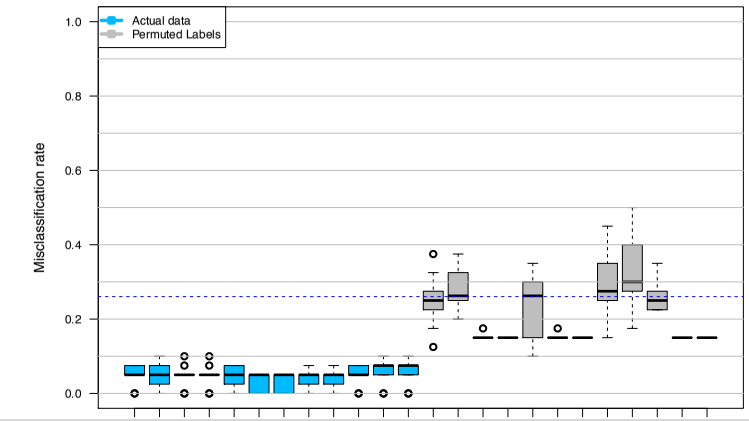
Figure 26 shows the following:

- The expected and observed MER are low in all cases.
- The MER achieve with permutated labels is inferior to the expectation, except for SVM with non-log2 transformed data. (in this case, the permutated labels fit the random expectation-
- In all cases, the actual class labels give better results than permutated labels, which indicates that there is something to be learned from the permutated labels.

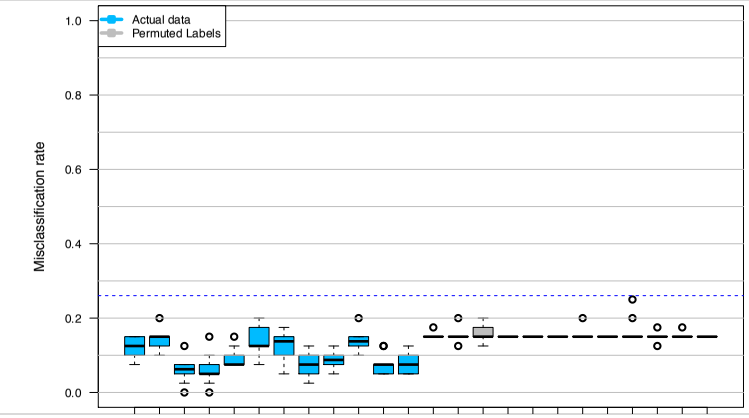
My hypothesis is that the apparent "learning" effect with permutated labels may result from a trivial assignment of all of the samples to the major class. Since the dataset is strongly imbalanced, this "strategy" would achieve better results than a random assignment balanced on the prior class frequencies

Figure 26 Impact of normalisation on the misclassification error rate of classifiers on the *Lupus erythematosus* study (SRP062966). Legends as in Figure 22.

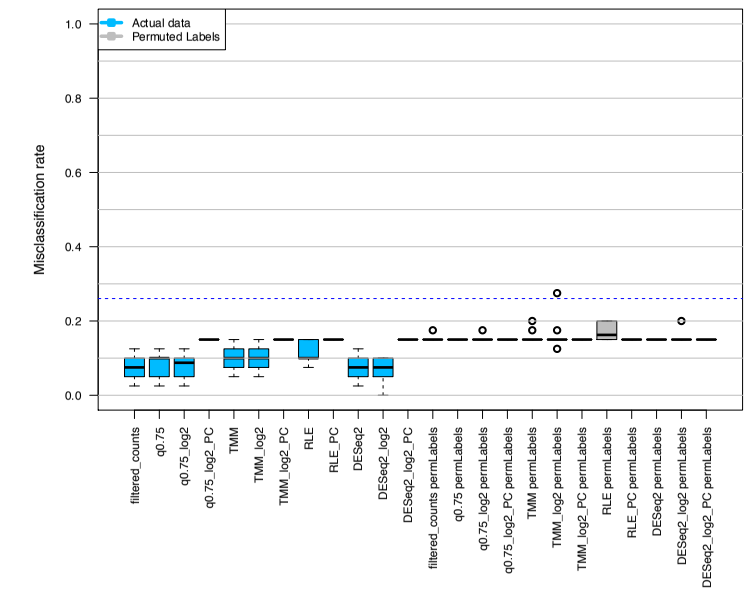
SVM



KNN



RF



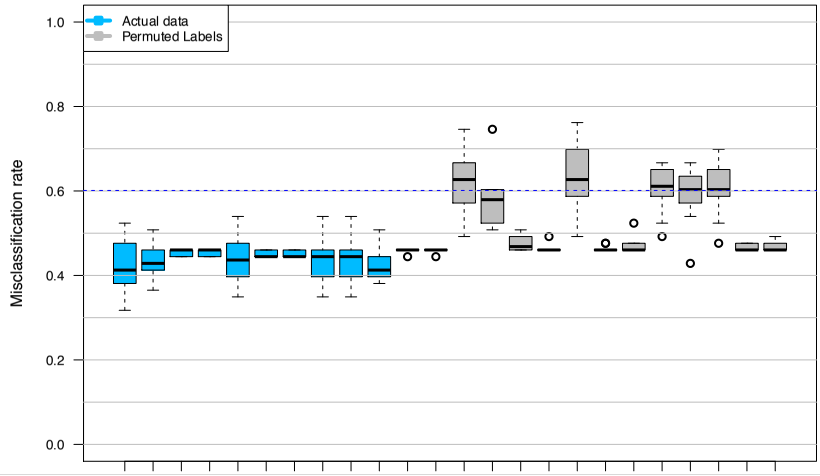
Cancer disease types study (SRP061240)

This dataset shows the same striking behaviour as the *Lupus erythematosus* dataset, but with an even stronger impact: the percent of MER is the same for the actual and permutated class label, which indicates there is no benefit of the training process to increase the efficiency of the classifier.

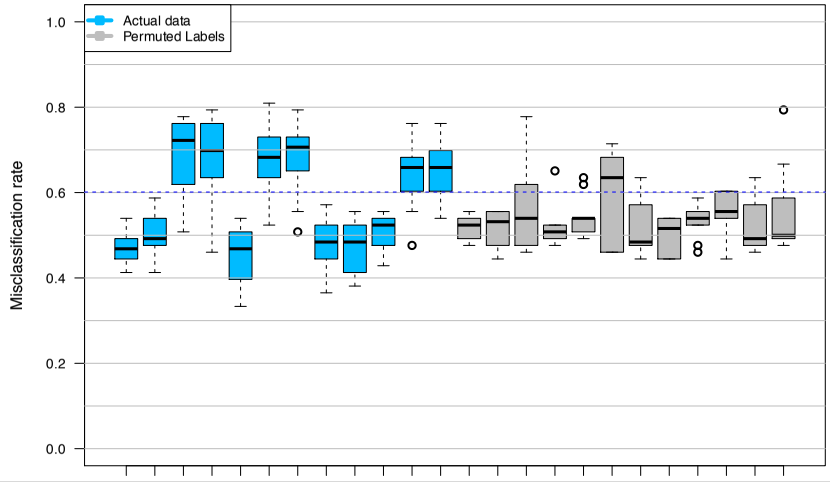
Another surprising observation is that with log2 transformed data, KNN gives a higher MER than the random expectation, suggesting that the trained classifier is able to learn … the wrong answers

Figure 27 **Impact of Normalisation on the misclassification error rate of classifiers on the *Cancer disease types* study (SRP061240).** Legend: see Figure 22.

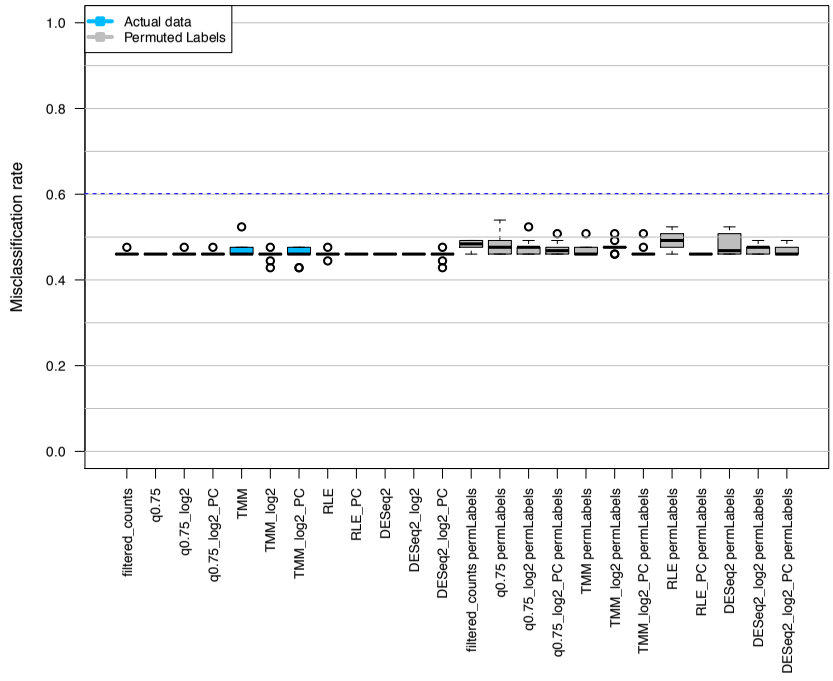
SVM



KNN



RF



Impact of Principal Component transformation on classifier performances

Principal Component analysis (PCA) is useful in linear feature extraction where it is considered a multivariate data analysis (Jin and Bie).

PCA finds a linear transformation $Y = WX$ such that the retained variance is maximised. It can be also viewed as a linear transformation that minimises the reconstruction error (Diamantaras and Kung, 1996). Each row vector of W corresponds to the normalised orthogonal eigenvector of the data covariance matrix.

We use PCA can be used to reduce the large RNA-Seq feature space to lay a reasonable number of dimensions with little information loss.

In most recount repositories, which contain around 2000 RNA-seq experiment project datasets, each has a large RNA-Seq dataset, with different numbers of samples (individuals) in each experiment; this depends on the published experiment and 58000 features (genes). Consequently, the separation between training and testing sets enforces the over-dimensionality, since we have taken 2/3 of the data for training and the remaining 1/3 is for testing.

The real problem resulting from the over-dimensionality of the feature space is that some methods are sensitive to it, and their model is over-fit to particular cases used for the training stage.

An additional advantage of PCA transformations is that it strongly reduces the computing time, since the number of components (and thus of features used afterwards) cannot exceed the number of individuals (samples).

The important point here is that the number of feature dimensions is very large (close to 58000) which will take too long to train and test. The aim of PCA-based multidimensional scaling is to reduce the number of features to b dimensions, thereby preserving most of the variance (so that we don't lose relevant vital information while doing so) of the actual data while b is still significantly small.

A common way to grasp the impact of PC transformation is to draw a scatter graph where individuals are plotted according to their values on the two principal components.

By applying PCA in the whole data set, that implicitly means that we are attempting to project the data onto first two principal components - though orthogonal to each other. It should be noted though that two Eigen vectors of these data will be the basis for a totally different two-dimensional subspace.

Although the number of dimensions we are projecting the data onto is the same, the data alone is different and might end up being projected onto different dimensions.

The Random Forest (RF) method was developed by Breiman (2001) and relies on the combination of multiple tree predictors, each of which was trained on values of a random subset of individually sampled features and individuals. It assumes that distribution is the same for all trees in the forest. The generalization error for forests converges and tends to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them.

The mechanism of random forests in classification accuracy is growing an ensemble of trees and letting them vote for the most popular class, where the generated random vectors govern the growth of each tree in the ensemble.

When a great number of trees is generated, they vote for most popular class, in a procedure called random forests.

We could notice from **Figure 23** that all classifiers (SVM, KNN, and RF) have been affected more by log2 transformation than PCs transformations, such lead us to realize that the log2-transformation is appropriate to analyse RNA-seq data, but that a further PC transformation does not bring any further improvement. For RF, the PC transformation has even a deleterious effect, since it increases the MER. Therefore the filter and normalization would not make it possible to further improve the performance of the decision tree classification, especially with the RF classifier. This is due to their nature to create groups of samples depending on certain ranges of values.

On the other hand, the PC transformation led to substantial reduction of the training and testing time of classifiers, due to the strong reduction of dimensionality (the number of PCs cannot exceed the number of samples, which is always much lower than the number of genes).

It has to be noted that in this section we keep all the PCs, we will see later that the result differ when we select the first PCs as predictor features.

Generally, we can say that RF does not care about normalisation as it is an ensemble classifier consisting of a large number of different tress. Each tree is trained in different samples and a fixed number of randomly chosen features is used in each node. In RF, we should define the number of trees that the algorithm will build and the number of randomly chosen features that will be used in each node of the tree.

SVM and KNN are the opposite, as SVM uses the kernel trick to deal with nonlinearly separated data. SVM maps the initial data to a higher dimensional space, using a proper kernel

function, in which the data are linearly separable; the kernel function that we used in training svm classifier is linear.

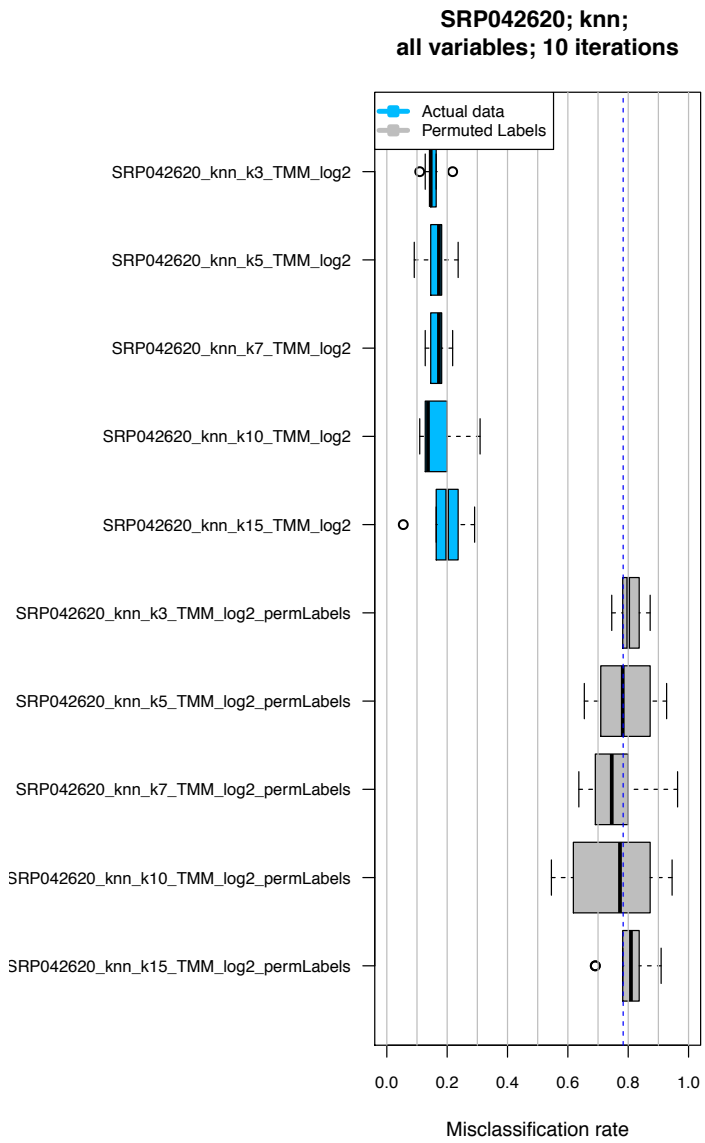
When SVM classification is applied to linearly separable data, the optimum separation hyperplane (OSH) is the hyperplane with the maximum margin for a given finite set of learning patterns. The OSH computation with a linear support vector machine is tested in our analysis to improve from the efficiency of classification. To control in this margin that is used to refine from the efficiency of SVM by controlling in the parameter "C-classification". This corresponds to KNN, which checked the k nearest samples of the test instance (nearest using Euclidean distance). It decides in which class the instance belongs by using a majority voting schema.

Impact of the number of neighbours (k) on KNN performances

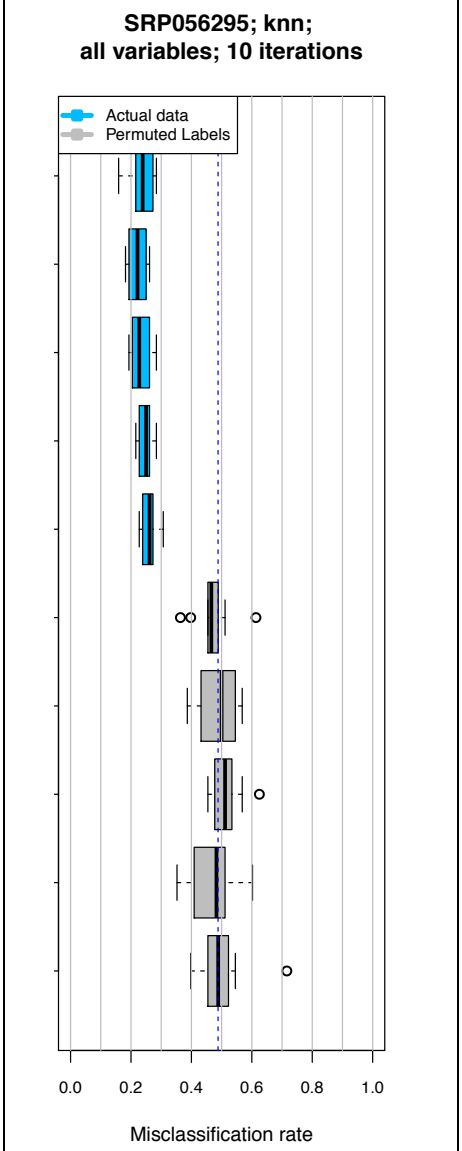
The main parameter for the KNN classifier is the number of neighbours (*k*) taken into consideration to assign a class to new individuals. In this section, we evaluate the impact of this parameter on the performances of KNN with the 7 studies. We tested the following values for *k*: 3, 5, 7, 10, and 15.

Figure 28 clearly shows the influence of k, which depends on the type of dataset: the minimal MER is obtained for $k = 10$ for breast cancer, $k = 7$ for Leukaemia, $k = 15$ for cancer types, and $k = 10$ for Lupus. For the controls with permutated class labels, the value of k generally does not affect the MER. However, for cancer disease type, the best classification is achieved with permutated labels and $k = 15$, which performs even better than when KNN is trained with the actual class labels. This seems to be a spurious effect for this dataset where the KNN is unable to learn the training classes (the blue boxplots are at the same level as the average background of permutated class labels). The other studies are provided in Appendix A4.

Breast cancer (SRP042620)



Human Leukaemia (SRP056295)



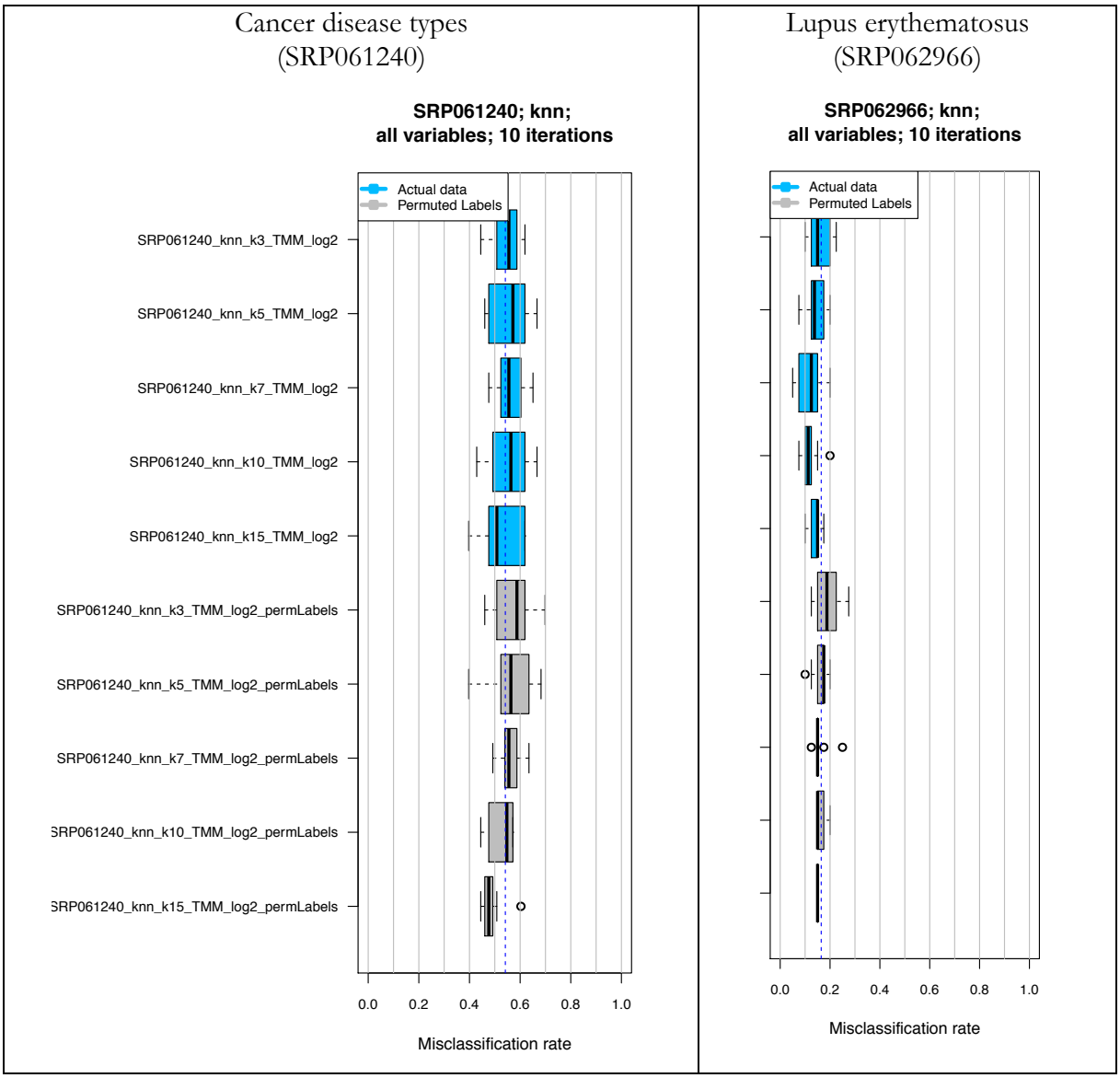


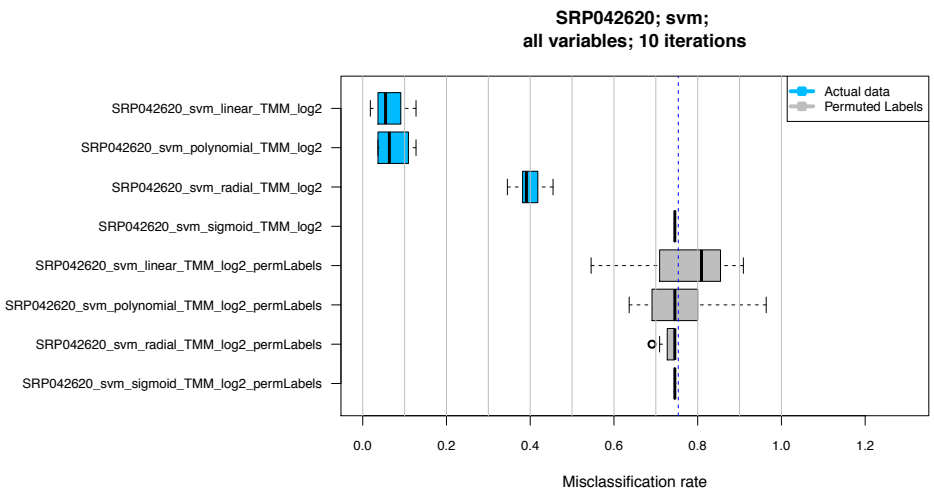
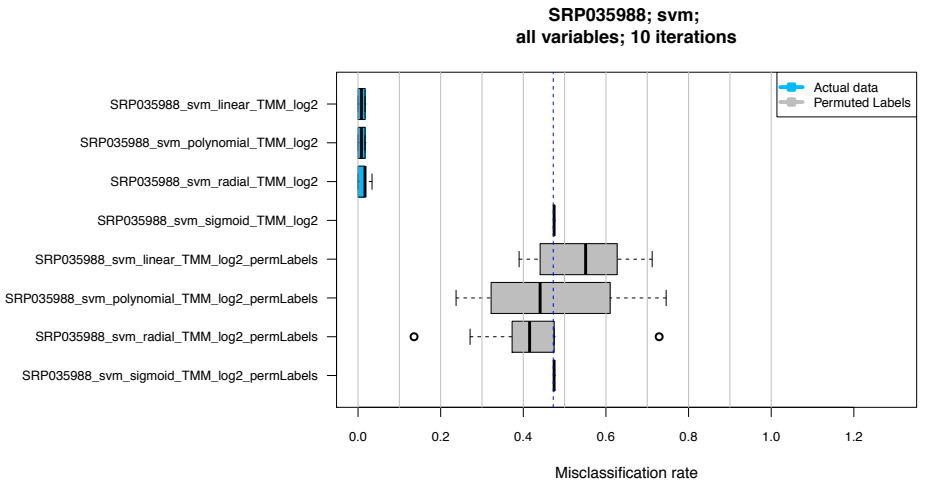
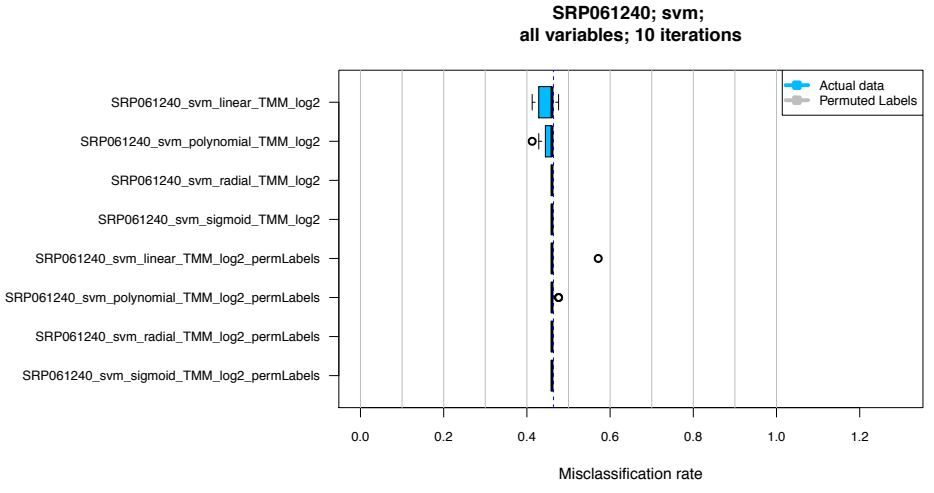
Figure 28 impact of K (nearest neighbour) of KNN into classifier accuracy measured by misclassification error rate.
The ordinate indicates increasing values of the *k* parameter (3, 5, 7, 10, and 15), the abscissa shows the misclassification error rates (MER). Each boxplot corresponds to one testing-training experiment (10 iterations). In all cases, we used the TMM-normalised log2-transformed counts. Blue boxes: result from the analysis of the actual datasets. Grey boxes: random expectation estimated by permuting class labels during the training and testing. Dotted line: mean MER value for the permutated class labels, indicating the background level of misclassification without training.

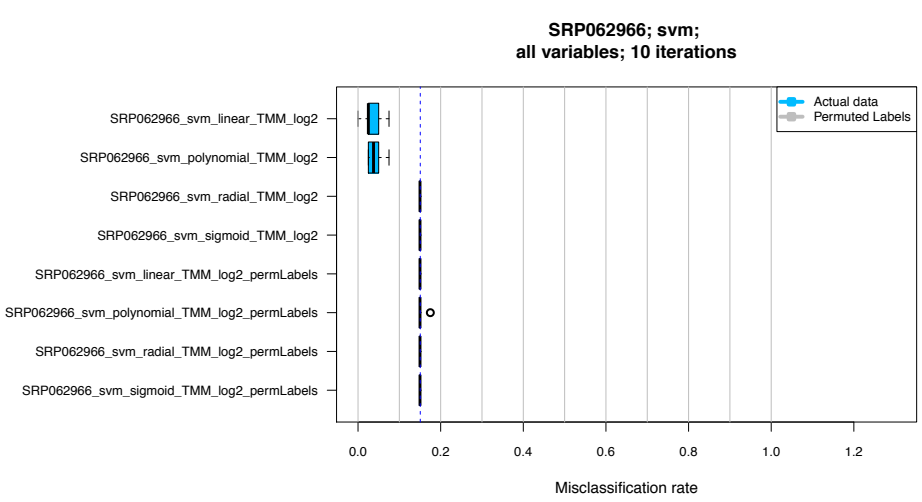
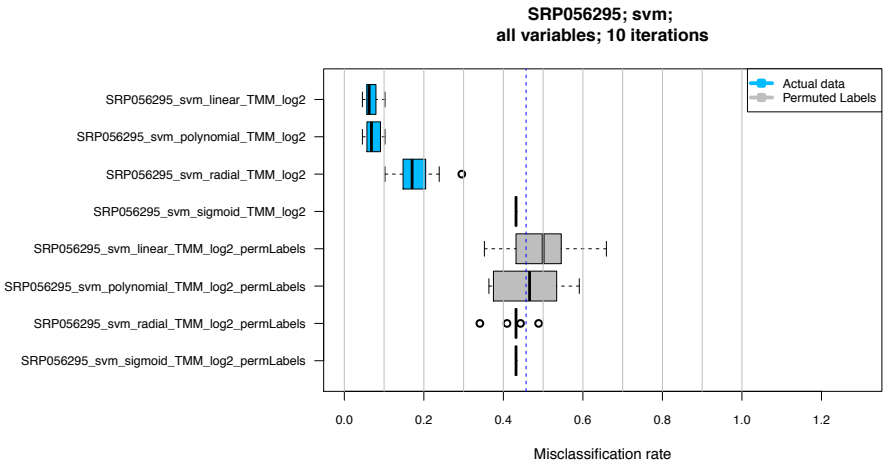
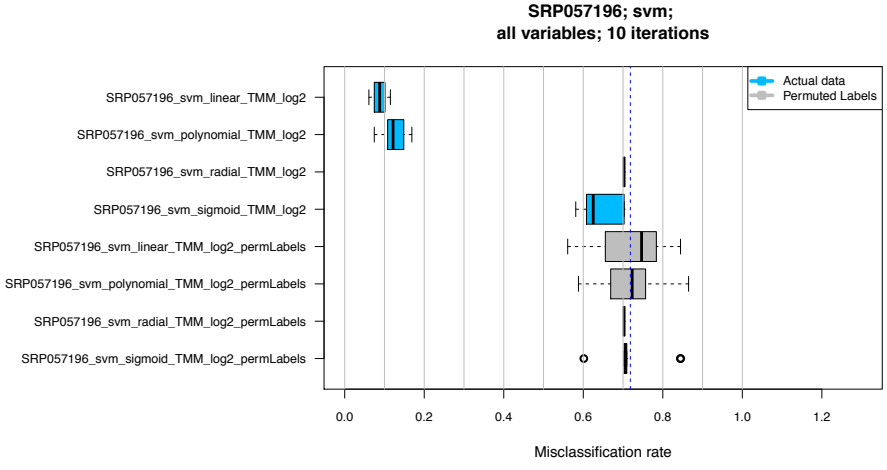
Impact of the kernel on SVM performance

Figure 29 illustrates the impact of the kernel of SVM performance. For all of the studies, the best performances are obtained with the linear kernel, immediately followed by the polynomial kernel. The radial kernel generally gives much poorer performances with all the datasets except for Psoriasis (which is a very easy-to-learn case) and for cancer disease, where anyway all the kernels

fail to learn the classes. The worst performer is sigmoid, which always gives similar MER as the classifiers trained with permutated label.

It should be noted that the radial kernel, which performs poorly with almost all our datasets, is the default value for the svm() function of the R (implemented in the e1071 package), and was also the kernel used by Johnson in their evaluation of classifiers with RNA-seq (Johnson et al., 2018). This throws some doubt onto the relevance of their conclusions, since they did not consider the SVM classifier with a linear filter, which provides the best results for all of the studies and with all of the pre-processing options in our evaluation.

Breast cancer (SRP042620)	
Psoriasis (SRP035988)	
Cancer disease types (SRP061240)	

Lupus erythematosus (SRP062966)	
Human Leukaemia (SRP056295)	
Adult and foetal Human brain cells (SRP057196)	

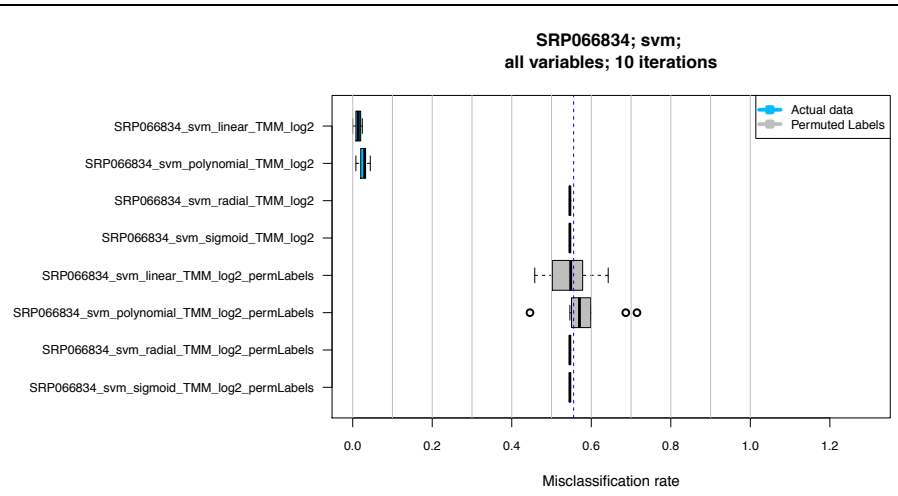
Cerebral organoids and foetal neocortex (SRP066834)	

Figure 29 impact of kernel of SVM into classifier accuracy measured by misclassification error rate. The ordinate indicates the values of the kernel parameter used for SVM (linear, polynomial, radial, sigmoid, respectively), the abscissa indicates the misclassification error rates (MER). Each boxplot corresponds to one testing-training experiment (10 iterations) with TMM-normalised log2-transformed counts. Blue boxes: result from the analysis of the actual datasets. Grey boxes: random expectation estimated by permuting class labels during the training and testing. Dotted line: background level of MER estimated with permutated class labels (untrained classifier).

Summary of the results

The assessment of the selected classifiers with the remaining studies are available in Appendix A (Supplementary Figures), and the results are consistent with the examples from previous sections.

Table 7. Summary of classifier performances for the 7 studies.
For each classifier (SVM, RF, RF) we retain the MER obtained with the log2-transformed TMM-standardised dataset. Numbers indicate the Misclassification Error Rate (MER) in percent.

Study		Trained with actual class labels			Trained with permutated class labels		
ID	**Short name**	**SVM**	**KNN**	**RF**	**SVM**	**KNN**	**RF**
SRP042620	Breast cancer	10	20	10	80	75	75
SRP057196	Adult & foetal human Brain	8	22	18	66	75	75
SRP056295	Human Leukaemia	6	26	10	46	45	45
SRP035988	Psoriasis	0	0	0	45	50	50
SRP061240	Cancer disease types	45	68	46	47	64	46
SRP062966	Lupus erythematosus	5	13	10	15	15	15
SRP066834	Cerebral organoids and foetal neocortex	0	23	7	60	55	55

Table 7 summarises the assessment results for the 3 classifiers with the 7 studies. We only considered the log2-transformed TMM data to achieve a fair comparison. Indeed, as discussed above, log2 transformation strongly reduces the MER for SVM and KNN, without affecting RF in any direction, whereas PC transformation of the log2-transformed data does not bring any further improvement for KNN and SVM, but strongly deteriorates the accuracy of RF. The choice of TMM is somewhat arbitrary, since classifier performances are globally similar irrespective of the method (q0.75, TMM or DESeq2) used to standardise library sizes.

Based on this comparison, we can draw some general conclusions from our comparative assessment as follows.

SVM is an optimal classifier to separate RNA-seq samples based on their expression profiles; the pre-processing procedures are also required to improve the efficiency of the SVM classifier. Besides, SVM is also taken as the first grade in the time of execution in the training process.

RF takes the second place in the ranking, wherein it classifies samples of RNA-seq data as well having additional advantages. It does not require pre-processing procedures, since it achieves the same MER with or without pre-processing. This particularity deserves to be noted, and suggests the need to recommend RF in cases of doubt about the choice of pre-processing options. However, we notice here the high cost in execution for the training process.

KNN is the worst-performing classifier in our tests. It is somewhat faster than RF but slower than SVM. Moreover, the findings of MER were also acceptable for initial evaluation and to give the quickest test of the RNAseq dataset.

Generally, supervised classifiers are trustworthy for classifying the RNA-seq data, where reliable results are produced which are used with RNA-seq data. In particular, principal component analysis was not able to classify such RNA-seq data, as shown in Chapter 4. Data pre-processing, in particular with the Cerebral organoids and foetal neocortex study, as clarified in **Figure 26** which clearly show the inability of PC to segregate the samples according to their respective class. Consequently, there is a need for investment of advanced supervised classification methods to completely segregate the sample according to their respective classes.

The general outcome from this chapter can be summarised as follows.

- SVM and KNN both need the log2-transformation;
- PC transformation does not further improve the performances of log2-transformed data for SVM and KNN, but it provokes an astonishing decrease of the execution time in training the classifiers.

- In contrast, with RF the PC transformation decreases the effectiveness of RNA-seq data classification.
- With RF, the log2-transformation and PC does not show much improvement in the classification process. This is unfortunate because a gain of processing time would have been particularly significant for RF, as it takes a long time to execute classification process, that is because its principle relies on the random splitting of the training set to create trees; in turn, that requires a long time, but pre-processing stage does the execution-time acceptable and worthwhile, as such shown by the MER ratio for RF with PC doesn't make any sense in improve the effectiveness of RF classifier.

Furthermore, there are no fixed expected ratios for MER within the analysis of RNA-seq data, which really depends on the nature of the RNA-seq data, as the nature of such data varies from one dataset to another. The underlying essence is that the RNA-seq data will play a primary role, affecting the value of MER and consequently affecting the efficiency of the classifier.

CHAPTER 6: IMPACT OF FEATURE SELECTION ON CLASSIFIER ACCURACY

Feature selection is used to extract the relevant features to be fed into classifiers, and remove the non-discriminant and/or redundant features to reduce the curse of dimensionality. This would make the learning process for classification time-efficient and increase the performance of classification algorithms by reducing the over-fitting (Hoi et al., 2012). For next-generation sequencing (NGS) data feature selection process, such as RNA-seq data, both supervised and unsupervised learning can be implemented to make decisions about the subset of features to be retained. One of the simplest feature selection approaches is *variable ordering*, which consists of sorting the features according to their relevance to the classification methods and retaining the top-ranking ones to enhance the performance of classification methods (Tan et al.).

Expression level can be used as the basis for diagnosis by assigning samples to different classes based on a prior analysis of the changes in expression level between classes. With microarrays, many studies adopting this technique have been developed to determine the important features (biomarkers) in order to predict the class of disease types, phenotypes, and tissues types (Jayawardana et al., 2015; Strbenac et al., 2016; Marisa et al., 2013).

We have tested three alternative criteria to rank the features: Principal Components (PC), Differential Expression (DE), and Variable Importance (VIMP), which are assigned during the first pass of the random forest training/testing cycle.

Feature selection based on Principal Components

Principal Component Analysis can be used as a dimension reduction approach, by decomposing the data into new variables (called components) that are linear combinations of the original variables. The first components capture most of the variance, and supposedly explain most of the differences between individuals. By contrasting of clustering methods that may classify individuals in RNA-seq data as with classification methods, the PCA did not identify these clusters, but instead focuses on those methods that extract the linear relationships that best explain the correlated structure across data sets. Moreover, PCA has the ability to illustrate the variability both within and between features (variables) and may highlight data issues such as batch effects or

outliers. In our study, we applied PCA in order to generate new data sets comprised of a number of first-ranking components which ranged from 2 (only keeping the two first components) to the number of individuals in the considered dataset (which is the maximal number of components produced by PC transformation). Thereby, a new resultant dataset will have at most the same number of features as individuals. **Figure 30** shows the impact of PC-based feature selection on the performances of the different classifiers. The blue boxplots represent the distribution of the MER generated from 10 iterations for each of the targeted classifiers:

SVM: from left to right, the boxplots show the results obtained when classifiers are fed with increasing numbers of PCs (starting from PC2, PC3, etc.). We analyse the impact the number of PCs on the efficiency of SVM (top panel of **Figure 30**), and observed that the MER progressively decreases when the number of component increases, and that SVM will provide the best results when it is fed with at least 40 components, which gives an MER of around 2%. Besides that, this gives us the first impression which implicitly indicates that SVM can overcome the overfitting; we noticed that MER decreases when the number of PCs increases, and it stabilizes around 2% when number of PCs exceeds 40. But with the permutated labels for the classes of samples, we noticed that there was no effect of increasing the number of PCs: whether the number of PCs is 3 or 162, the MER will be more or less the same, with a value of around 78%, which corresponds to the theoretical random expectation of 78%.

KNN: which such classifiers, the situation is drastically different from SVM; for 2 PCs, the MER was almost 44% and then declined to reach 8% with the first 4 PCs. After that, when the number of PCs increased, then the MER also increased, suggesting an effect of overfitting. When the number of PCs was 140, the MER reached 50%. This shows that KNN has no intrinsic capability to control overfitting; afterwards, with PCs (150, 160 and 162) the MER again declined to arrive at 37%, in contrast with SVM. The permutated class labels show the same results as with SVM: the MER corresponds to the random expectation, irrespective of the number of components.

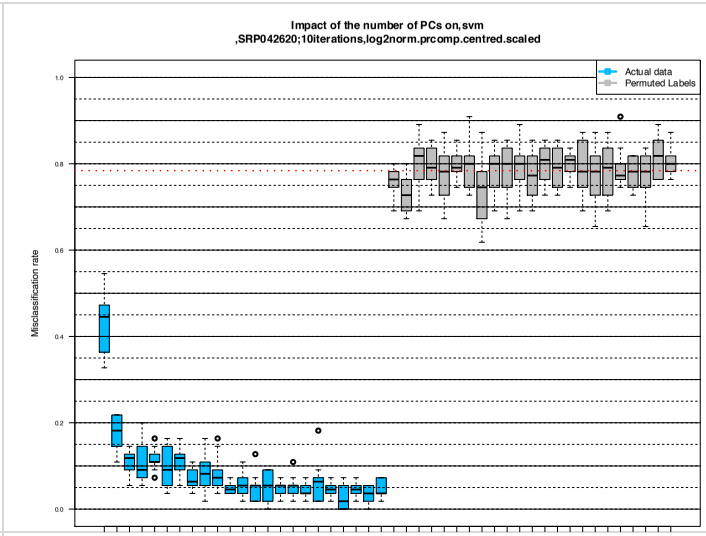
RF: has an essentially different behaviour, which confirms our conclusion that RF does not care about the normalisation, and that the number of PCs has a slight effect on the efficiency of RF. With PC2, the MER was very high, 43%, which confirms the inability of RF to extract the information from PC2. Consequently, with PC3, the MER was almost 18%, but a visible decline in MER was around 13% with PCs (3, to 30). Afterwards, there is fluctuation in the MER, first increasing and then decreasing with the PCs (40 to 162), where the MER returned the following values corresponding to 10%, 15%, 12%, 11%, 13%, 12%, 12%, 14%, 14%, 17%, 13%, 16%, 13%,

14%, and 18%. It turns out PCs have a modest effect on RF, as the situation with permutated class labels does not differ from the prior, wherein it was the same as with SVM and KNN, 78%.

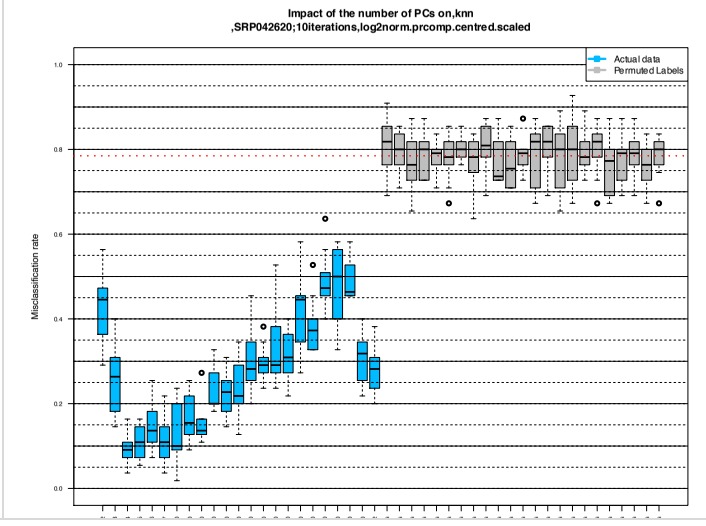
For further illustration of the impact of the number of PCs on the efficiency of the targeted classifiers, Appendix A3. Supplementary figures clearly shows the overfitting issue: when the number of PCs increase, it leads to a decrease in ratio of MER up to a given number of PCs, but afterwards, when the number of PCs continues to increase, this leads to increase the MER, and a further increase of PC leads again to decrease the MER. This unstable behaviour clearly indicates to sensitivity to overfitting issue. More results are in Appendix A3. Supplementary figures.

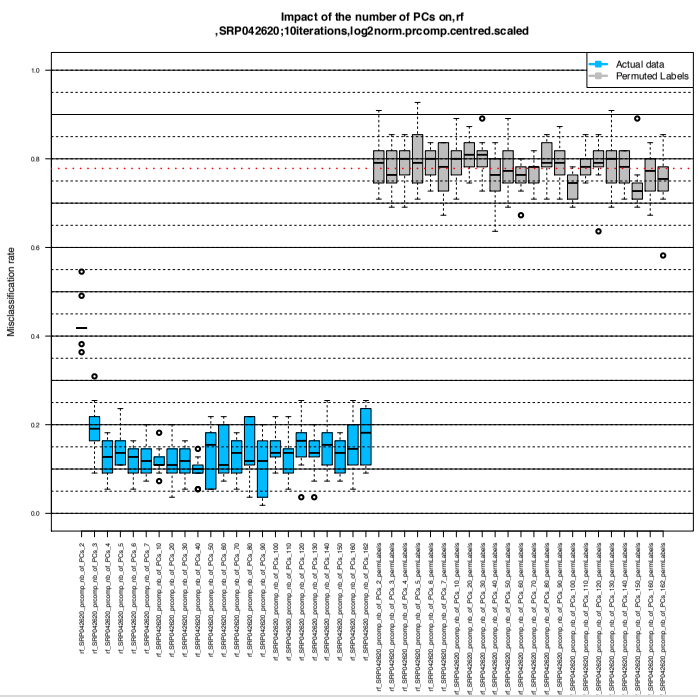
Figure 30 Impact of the feature selection approach based on principal components on the misclassification error rate of classifiers for the breast cancer study (SRP042620). The ordinate indicates the misclassification error rates. Each boxplot corresponds to one testing-training experiment (10 iterations) and one particular feature selection based on PCs method. **Blue boxes**: results from the analysis of the actual datasets. The blue box plots from left to right correspond to one series of training/testing sets with increasing numbers of first components, ranging from 2 (left) to the maximal number of components (162 for this study). Number of components: 2, 3, 4, 5, 6, 7, 10, and all values from 20 to 160 by steps of 10. **Grey boxes**: random expectation estimated by permuting class labels during the training and

SVM



KNN



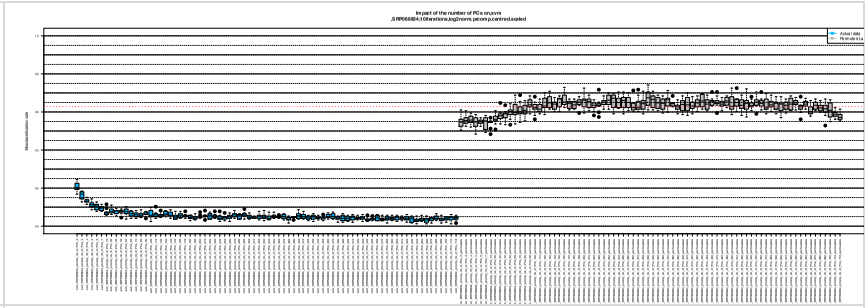
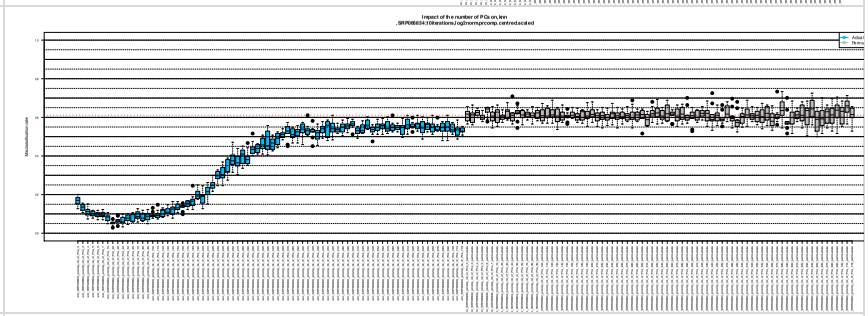
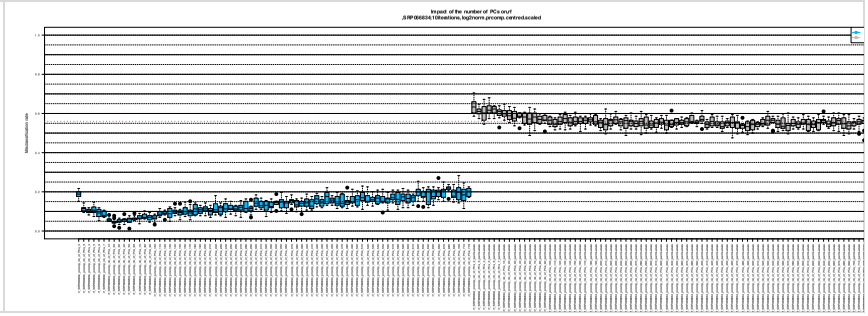
testing. **Top**: Support Vector Machines (SVM); **Middle**: K-nearest neighbours (KNN); **Bottom:** Random Forest (RF).	RF	

In order to evaluate how each classifier tackles the problem of over-dimensionality of the data, **Figure 30** show the behaviour of each classifier when the number of principal components progressively increases.

SVM: is insensitive to over-fitting; this is notable when the number of PCs varies from 2 to 10, as the MER starts to improve. It starts at 21% with 2 PCs and declines to 4% with 150 PCs; afterward, it stabilised with even with greater than 700 PCs. That is a good indicator of the ability of SVM to overcome the overfitting issue.

KNN: is very sensitive to over-fitting; as can be seen when the number of features is greater than 70, which clearly shows an inability for KNN to solve the overfitting issue. With the number of PCs being 20, the MER was almost 5%; afterwards, it started to progressively increase, finally reaching 56% when the number of features was 450. Then, it stabilised to 56% with 718 PCs. That indicates that KNN is not able to overcome the overfitting.

RF: random forest actually gives better results with a selected subset of the PCs than when all of them are used. This is especially true when the number of components is large, as shown in **Figure 32**. When the classifier is fed with 20 PCs, the MER decreases to 4%, whereas it starts to increase when the number of PCs increases. These results show that RF is somewhat sensitive to over-fitting when it is fed with PCA-transformed data. In such cases, the selection of the first components can bring improvements in accuracy.

Figure 31. Impact of the selection of principal components on the performances of classifiers for the *Cerebral organoids and foetal neocortex* (SRP066834). Legend: see **Figure 30**.	SVM	
	KNN	
	RF	

Feature selection based on Differential Expression analysis

The difference in expression level (**differential expression analysis**, **DE**) of genes has proven to be helpful in classification problems, enabling individual characteristics to be identified (Strbenac et al., 2016) and samples to be distinguished according to their group membership. For RNA-seq data, we have been ranking the genes in ascending order according to the adjusted p-values reported by two R packages for differential expression analysis: DESeq2 and edgeR, respectively. We then generated a series of subset datasets that only contain the features with the highest significance based on the adjusted p-values to retain relevant features and evaluate their effectiveness for classification methods. The blue boxplots in **Figure 32** indicate the evolution of the misclassification error rate (MER) accordingly, when we select increasing numbers of top-ranking features based on the DE adjusted p-values. Each figure shows the result with 3 series of ordering criteria: DESeq2 adjusted p-value, edgeR adjusted p-value, and random ranking of the genes. The latter are used as a control to evaluate the relevance of DEG-based ordering. The grey boxplots show the MER obtained when the classifiers are trained with randomly permutated class labels and provide an estimation of the expected error rate with untrained classifiers.

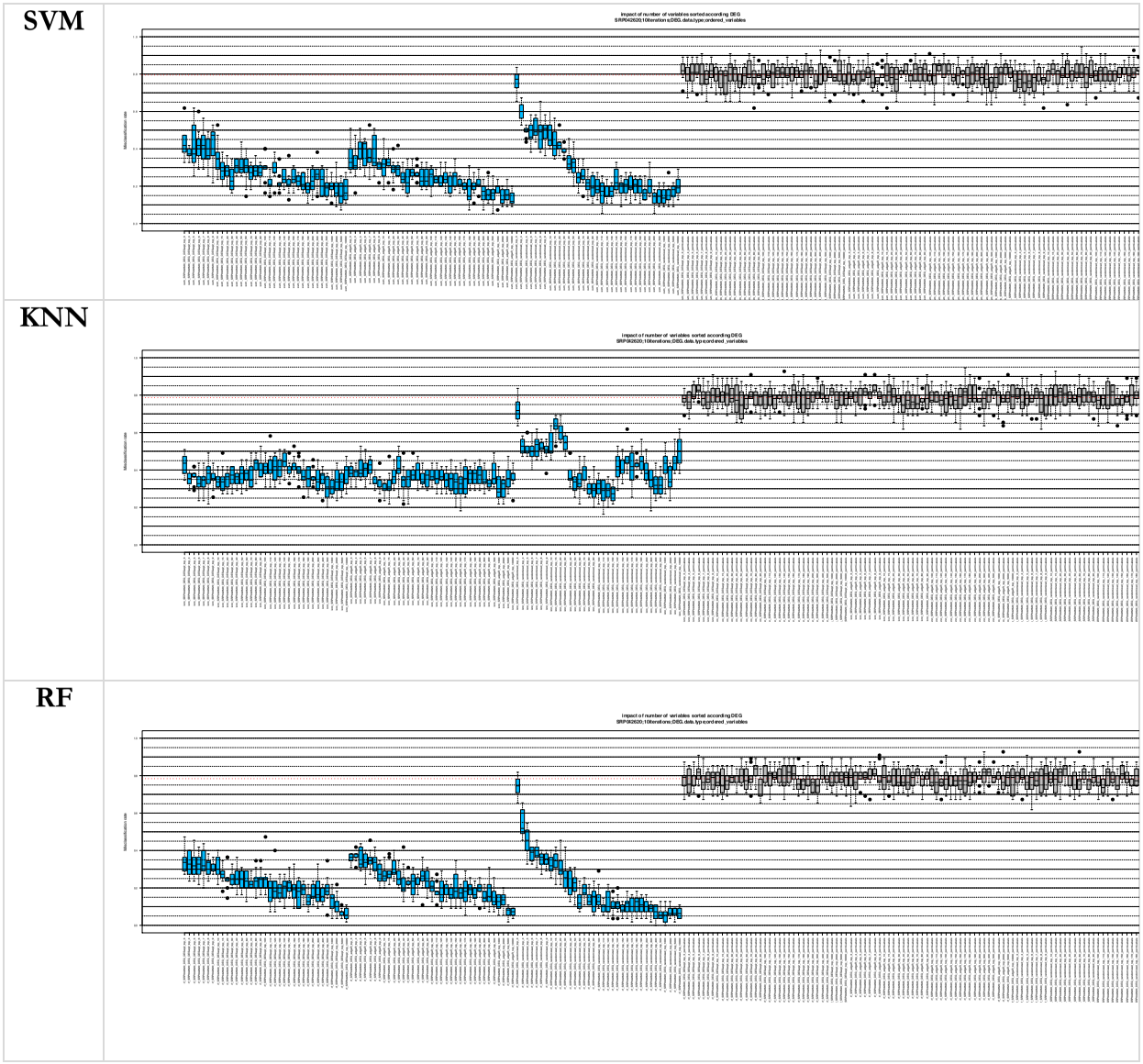


Figure 32 Impact of feature selection on the misclassification error rate of classifiers for the *Breast cancer* study (SRP042620).
The ordinate indicates the misclassification error rates. Each boxplot corresponds to one testing-training experiment (10 iterations) and one particular feature selection method. **Blue boxplots:** From left to right, the 3 series of feature selection respectively correspond to DESeq2, edgeR or random ordering of the features. Within each series, the number of top-raking features progressively increases from left to right (numbers of top-ranking genes: 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 300, 400, 500, 1000, 2000, 5000, and 10000). **Grey boxes:** random expectation estimated by permuting class labels during the training and testing. **Top:** Support Vector Machines (SVM); **Middle:** K-nearest neighbours (KNN); **Bottom:** Random Forest (RF).

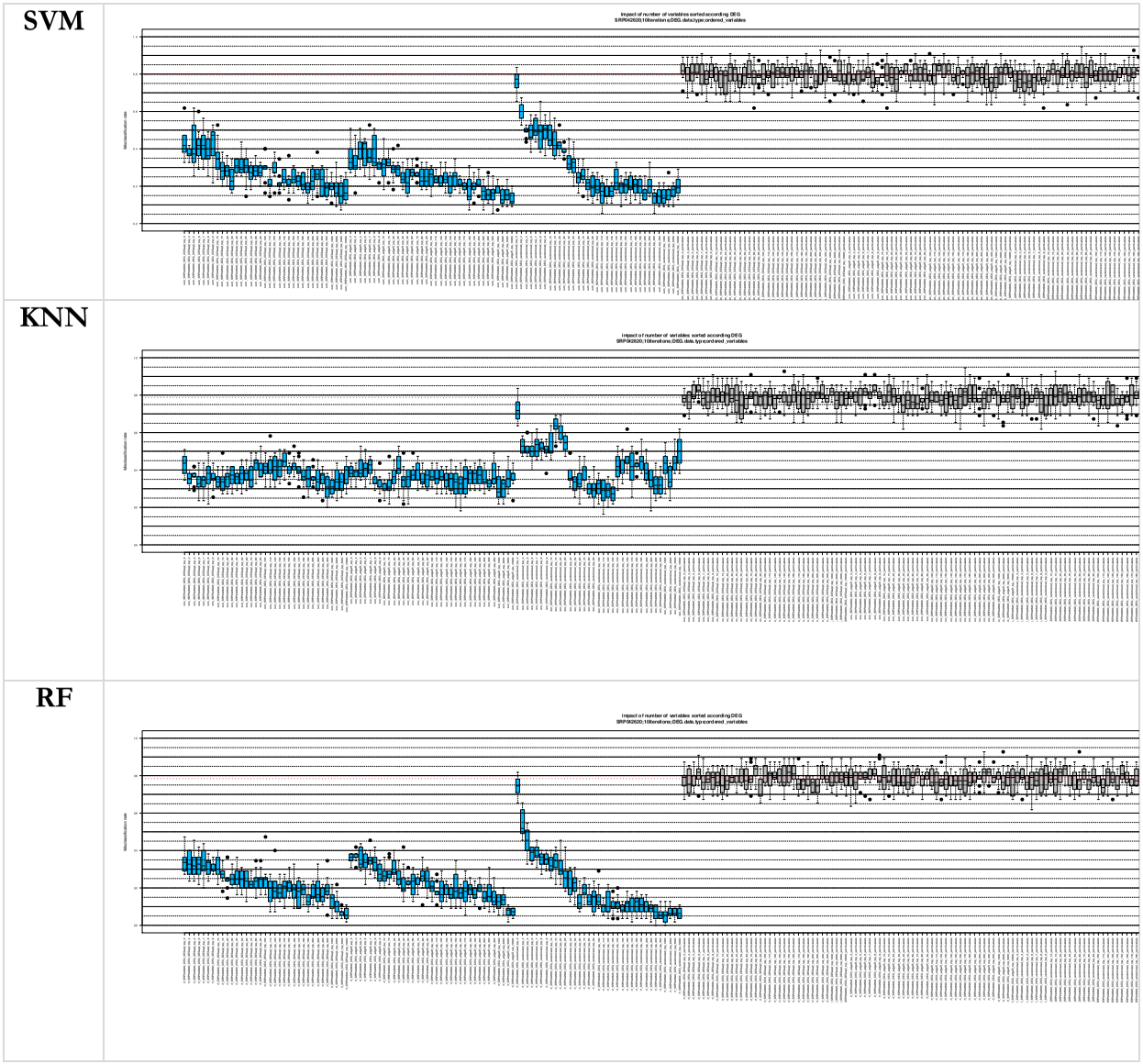


Figure 32 shows the behaviour of SVM classifiers trained with subsets of increasing sizes of top-ranking genes ordered by ascending p-values.

SVM (top panel): we tested the SVM with three groups of ranking features; the first series of boxplots (which numbers of genes ranging from 3 to 10000) corresponds to DEseq2-based ranking. The MER steadily decreases when the number of features increases, showing an increase in the efficiency of SVM. When fed with the 3 top-raking differentially expressed genes, the MER starts at 34%, and with 10,000 features it ends up with 17% MER. This decrease gives us evidence of the role of DESeq2 to enhance the effectiveness of SVM. With the second series of blue boxplots, which corresponds to features ranked based on edgeR, we notice the same effect: the MER progressively decreases from 33% to 11% as the number of features increases from 3 to 10,000. In addition, we tested SVM with the same subset of 10,000 top-ranking features reported by DESeq2, but we permutated their order randomly. With these randomly ordered features, SVM

gave somewhat inferior results with the initial subsets of the series (from 3 to 120 genes kept) but there was a rapid decline of MER afterwards. When the SVM is trained with 130 to 300 features, we noticed a slight increase to 23% MER, followed by a decrease with 400 to 1000 to reach 14%, and again increased with 2000 to 10000 features, to arrive at 18% MER. The lower effectiveness of random features indeed shows the relevance of the ranking with DESeq2 and edgeR to improve from effectiveness of SVM. The behaviour of such classifiers with the permutated class labels reached 80%, which was around the theoretical random expectation.

KNN (middle panel): the case here is quite different from SVM. We notice that the MER for KNN trained with DEG-based ordered genes (two first series of blue boxes) starts at 44% with first 3 features, then decreases to 33% and afterwards increases to 45% with 150 features, then drops down to 31% with 500 features, and reaches 34% with 10,000 features. With randomised features (third series of blue boxes) we observe the influence of overfitting: when the number of features varies from 4 to 30 the MER was high (around 58); afterwards, when the number of features varies from 30 to 120 it falls to around 27%, but when the number of features increases from 140 to 190, the MER rises to 43%. With 200 to 500 features, the MER decreases again to 33% but with 500 to 10,000 features it rises again to 43%. This suggests some overfitting effect; indeed, KNN has no built-in ability to overcome it, in contrast with SVM. With the permutated class labelled (grey boxes), the KNN returns the same error rate as SVM (~80%), which corresponds to the random expectation for an untrained classifier.

RF (bottom panel). Random forest shows the same trends as SMV, with the first series of blue boxes (DEseq2 ordering), but with 10,000 features, RF returns a smaller value for the MER (5%). The second series of blue boxes (edgeR ordering) also shows the same effect as SVM, but with a smaller value of MER (5%). With the third series (random ordering of the variables), RF clearly shows greater stability than SVM, where MER steadily decreases without fluctuations, as seen with SVM. That gives us clear evidence about the RF being more stable than SVM against overfitting, especially in the case without ordering for the best feature in the tested dataset.

In summary, the feature selection-based DEG is more effective for RF than SVM or KNN. Besides that, it provides evidence about the ability of each classifier to overcome the overfitting, and shows that KNN is more prone to overfitting. With permutated class labels, all classifiers return the same MER, corresponding with the random expectation.

We have targeted a simple comparison of two tested DEGs (DESeq2 and edgeR); we notice that the top-ranked features perform a bit better by increasing adjusted p-values reported by DESeq2 than with edgeR, suggesting that the most significant features extracted from DESeq2 are more relevant than those that returned by edgeR.

The behaviour of each ranked feature by DE (DEseq2 and edgeR) with the different classifiers for all seven datasets is provided in Appendix A3.2.

Feature selection based on variable importance returned by a first pass of random forest

An interesting property of RF is that it provides a rapidly computable internal measure of **variable importance** (**VIMP**), which can be used to rank features (variables) and use them for a second round of supervised classification, based on any classifier method of our choice. This feature is especially useful for high-dimensional genomic data. Two commonly evaluated importance measures are node impurity indices (such as the Gini index) and permutation importance. In classifications, the importance of the Gini index is based on the node impurity measure for node splitting. The importance of a variable is defined as the Gini index reduction for the variable summed over all nodes for each tree in the forest, normalised by the number of trees (Genuer et al., 2010).

Figure 33 shows the behaviour of the three classifiers (SVM, KNN and RF) when fed with increasing subsets of features selected according to their variable importance returned by a first round of RF.

SVM (top panel): shows a good performance with increasing numbers of VIMP-ordered features, where the MER starts from 42% with the 3 more top-ranking features, and then decreases progressively to reach 13% with the 10,000 most important features, which is subset from whole ranked dataset. This indicates once again the aptitude of SVM to solve the overfitting issues when number of feature are much more from the number of the individuals.

KNN: shows a different behaviour with the features ordered according its VIMP, where it shows non-monotonic changes: with a small number of features (from 3 to 7) the MER was almost 42%, and then declines to 25% with the top 80 to 160 features, and down to 21% with 170 to 400 features; beyond this, it gradually increases to 26% with 10,000 features. That drove us to observe that KNN is not able to overcome the overfitting with the features ranked based on the VIMP.

Besides this, we could see the behaviour of KNN with these subsets of the best significant features, it is obvious the sensitivity of KNN to overfitting phenomena. that mean when the KNN be prone to overfitting, we can see the MER will firstly decrease with first of group from the most significant variables and when the count of significant variable ascending increase then that will

lead the MER to increased that mean the behaviour of KNN will be a little worse, and continuing to that the MER again will decrease with increasing the count of variable. the summary KNN is high sensitive for overfitting issue.

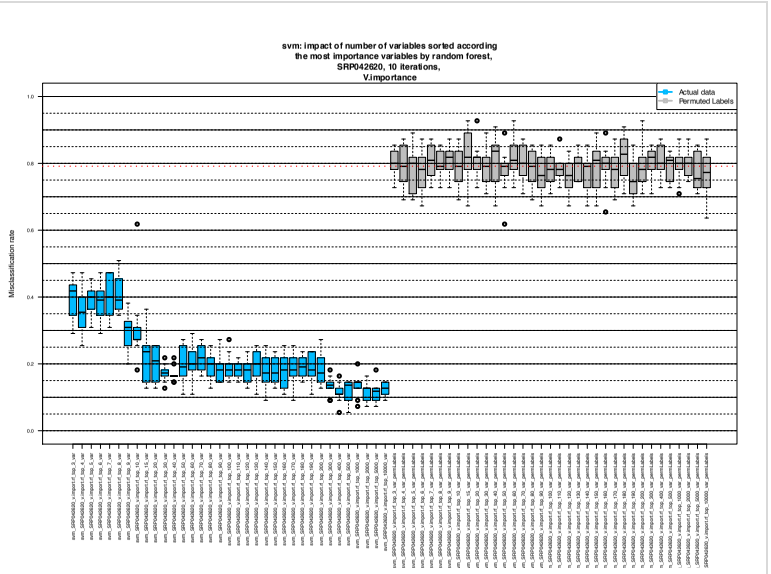
RF: has a more much monotonic attitude than the two previous classifiers; RF visibly showed great behaviour with ranked-features, which was the most significant according to VIMP. We clearly showed that when then number of features increased, this led to the direct decrease in MER. It started at 38% and kept falling to reach 5%. With the permutated class labelled, it showed the same situation as the previous classifiers.

We can conclude the importance of ranking features to boost the effectiveness of RF. Moreover, RF obviously has the power to tackle overfitting when the number of features becomes much greater than the number of individuals.

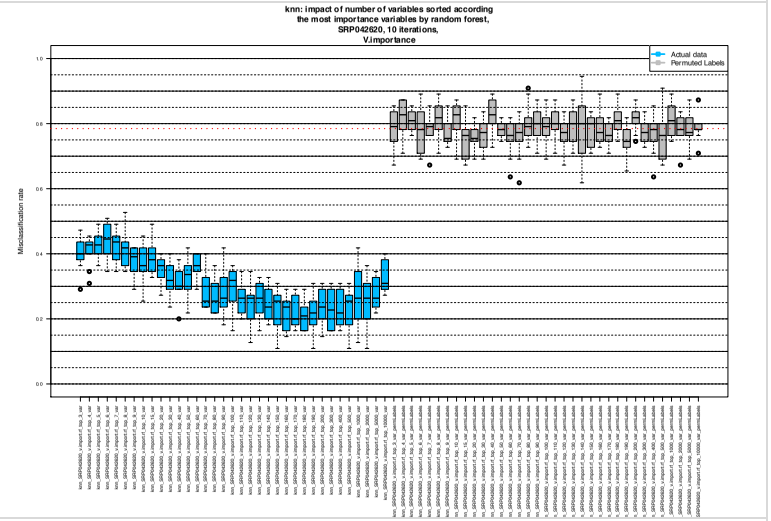
The complete results of feature selection based on variable importance generated from RF for the 7 studies are available in Appendix A.3.

Figure 33 Impact of the features selection approach based on the misclassification error rate of classifiers on the Breast cancer (SRP042620) relied on VIMP which are generated from RF. The ordinate indicates the misclassification error rates. Each boxplot corresponds to one testing-training experiment (10 iterations) and one particular feature selection method. Abbreviations: top-3-var-v.important: raw data ordered according to variable importance outputs from RF and number 3 indicate top significant feature based on the variable importance generated from RF; permLabels: random permutation of the sample labels, used to estimate the random expectation for the misclassification error rate. Blue boxes: result from the analysis of the actual datasets. Grey boxes: random expectation estimated by permuting class labels during the training and testing. Top: Support Vector Machines (SVM); Middle: K-nearest neighbours (KNN); Bottom: Random Forest (RF).

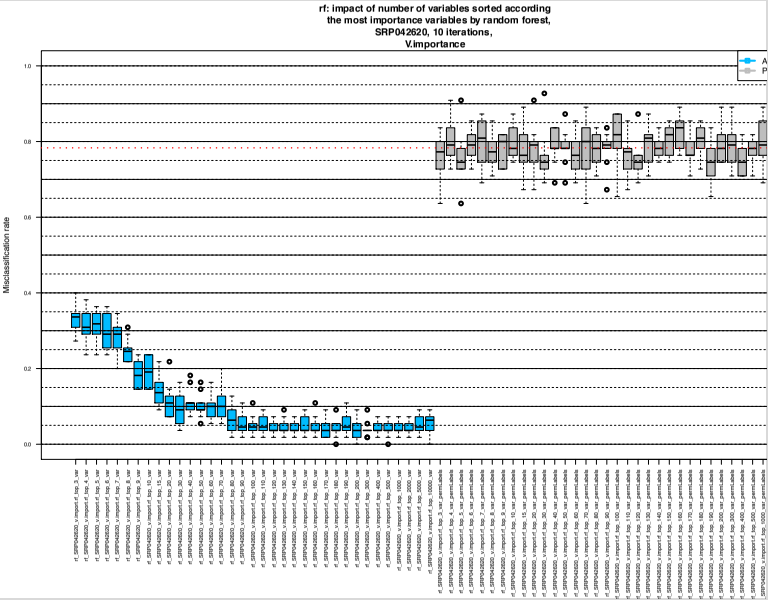
SVM



KNN



RF



Summary: impact of feature selection on classifier accuracy

We could remark that the MER of each classifier relied on the feature selection criteria, number of selected significant features and type of classifier, wherein:

SVM: was fulfilled with PC transformation the lowest possible percentage of MER, which might guide us to the more convenient selection criteria of PC transformation, which returned the lowest level of MER. Moreover, when the number of features is almost 10,000, the SVM have achieved the best performance as its MER was almost 5%.

KNN: has a somewhat dissimilar behaviour, especially with PC transformation; when the number of PCs reached 4, the MER was around 8%, which is the lowest possible value achieved for all of the features selected with KNN.

RF: has worked perfectly with ordering of variables by the VIMP selection criteria, wherein RF with the first 90 features have achieved the lowest MER value (~4%), and remaining stable at this level until the number of ordering variable reaches 10,000 feature. that This explicitly means that with RF classifier the first subset of top-ordering variables based on VIMP, the optimal number from the first ordering variable is 10,000 variable, and the rest of variables contains less information and it may be unrequired in classification process.

Briefly, using a selected subset of features based on PC transformation achieved better results with SVM and KNN than to handle all features. However, when we take into consideration the number of PCs that optimises the performance of the classifiers. the ideal first subset from the number of ordering variables shows strong variations depending on the type of classifier and the nature of the RNA-seq data.

Whereas with SVM was the ideal first subset from the RF VIMP-ordered variables was 140 variables, but with KNN the ideal first subset from ordering variables was around 40 variables, eventually with RF the ideal first subset was 80 variables. in brief, the count of the first from the first subset from significant variables depend on the type of classifier, moreover the nature of the RNA-seq data, thereby there are no ideal way to identify the count of the most significant variables that will satisfy the inferior level from MER to the classifier.

However, with KNN, the number of PCs was 4. Interestingly, the selected subset of features based on the RF achieved the best performance when the subset of significant features reached 90. In the end, there were no ideal feature selection criteria that rely on the type of classifier, the type of feature selection method and the number of subsets from the most significant features.

CHAPTER 7: GENERAL DISCUSSION

RNA sequencing (RNA-seq) technology was developed in 2007 and rapidly emerged as a replacement for microarrays for gene expression analyses. RNA-seq is preferred over microarrays because it has a higher sensitivity and dynamic range, with lower technical fluctuations, and thus higher precision than microarrays. Supervised and unsupervised approaches for analyzing microarray data have been well developed, but they are not suitable for RNA-seq data because microarrays measure gene expression in continuous intensities, whereas RNA-seq enables the absolute quantification of RNA levels by using discrete counts of reads per feature (gene, transcript). In recent literature, limited work has been done on supervised and unsupervised classification methods from RNA-seq raw count data for the classification of samples. In particular, problems might arise when data are over-dispersed, contain many zeros values, and have high dimensionality properties. In this thesis, we assessed and evaluated some of the most popular supervised and unsupervised classification approaches to classify samples based on their RNA-seq expression profile.

Supervised and unsupervised classification methods are valuable for classifying samples to improve diagnosis and prognostics for diseases, or to predict phenotypes and classify samples by tissues, among other uses. When we started this work, no research had dealt with the evaluation of the supervised classification analysis of RNA-seq data, with a focus on how to refine the performance of classification methods through gene expression analysis and variable ordering. Furthermore, no studies have focused on how optimal parameters that will help improve the performance of classifiers can be identified.

At the end of my thesis, during the last weeks of the writing of this manuscript, Johnson and co-authors published an article addressing the same question (Johnson et al., 2018) . However, their work relied on different study cases (from Human and rat), normalization methods, and classifiers. In contrast to our work, their study did not include a comparative assessment of the normalization method. Moreover, he did not perform a study of the impact of classifier parameters on their performances. In particular, for the SVM classifier, which outperforms KNN and RF in our study, Johnson and co-workers used the default kernel (radial) of the R svm implementation, which prove to be inefficient on almost all our study cases. We thus think that the conclusions of their study are questionable, at least for human data. One advantage of their study is that they compared the classifier performances with gene-based or transcript-based counts, and showed that transcripts

and isoforms provide better results. Their conclusions might however be affected by the choice of suboptimal parameters for the classifier.

We have seen that the results are variable: some datasets provided a very good basis for learning, almost irrespective of the classifier (e.g., psoriasis), whereas others, such as that on human leukemia, showed excellent learning with SVM and RF (SVM seems to outperform RF) but poor results with KNN; this is essentially caused by the overfitting problem. Some other (cellular complexity of adult and fetal human brain, lupus erythematosus, cerebral cancer type) datasets give very deceiving results with all the classifiers tested here.

We strived here to valorize the interest of using a diversity of study cases, as this would prevent us from drawing conclusions from a particular dataset that would not be applicable in general.

We concluded from our study that supervised classification methods are a powerful tool for classifying RNA-seq raw data. Aside from the ability of supervised classifiers to assign samples to their respective class, they facilitate novel findings on the role of genes based on the importance of assigning samples to relevant classes, which is the main objective of machine learning methods. In this work, we showed the capability of machine learning methods to meet the general goal of classifying RNA-seq data.

The non-normality of features in RNA-seq raw data led us to evaluate the adequacy of alternative pre-processing approaches on the performance of classifiers. There is a variety of pre-processing approaches that can be applied as a prerequisite step, notably those that rely on the nature of the targeted RNA-seq dataset, which is based on the type of sequencing used in the experiment and the resulting output from the trial. In our work, we were confronted with the common observation that RNA-seq data contain many zero values. Furthermore, datasets have high dimensionality, with more genes than samples. Expression levels also span a wide dynamic range of values, in which a few genes are expressed at a very high level (hundreds of thousands of counts per gene), whereas many others are barely detectable (a few counts per sample) or undetected. For this reason, we performed filtering at the gene, sample, and class levels for two main goals: the first is to perform the pre-processing procedures, machine learning methods will sometimes not work properly without this step, in addition that in some cases, the classifier will not work at all (program crashes); and when it does work, it may give us strange and unreliable results. The second goal is to make comparative assessment for effectiveness of pre-processing to determine which pre-processing approach will lead to the optimum performance for the classifier and for clustering.

Our finding is that SVM and KNN need log2-transformation but with PC-transformation (PC reduction urged RF to return bad findings) to improve the effectiveness of classifying RNAseq data.

This implicitly means that pre-processing is a required step for the analysis of RNA-seq raw data with SVM and KNN. Furthermore, pre-processing results in significant improvements in the reduction of the execution time, which is particularly useful here with the RF, as it may take a lengthy execution time.

To optimize the effectiveness of classifiers, we sought to find the suitable method for feature selection, which could lead to emphasis on the relevance of supervised classification methods to employ multi-variate analysis to analyze RNA-seq data (for example, clustering the genes resulting from supervised classification). Our role in the study is to find a suitable feature selection approach that will be more appropriate for the RNA-seq raw data. We therefore tested the following mechanisms. First, using differential expression analysis, we extracted and ranked the variables based on their respective importance relative to the p-value adjusted, and then we monitor how the feeding classifier with the important variables will improve with the performance of the classifier. Second, we tested the relevance of the important variables returned by first passed of RF method to enable us to rank these variables. We then fed the ordered variables to the classifier in order to evaluate if classifiers fed with top-ranking variables would lead to improved performance. In addition to mitigating the problems arising from high dimensionality, the feature selection approach provides information on how feature selection enhances classifier performance.

By utilizing feature selection methods based on DE, we identified the best subset of features (genes) of feature selection, and we sorted these features based on their differential expression values (adjusted p-value) to improve the performance of the classifiers. in other side pertaining the find optimal parameter for each classifier, we noticed that there are not unified parameters for each classifier which may give rise to obtain best results, but for each data set we need to customize certain value for the targeted parameter to each classifier, we accordingly that need identify certain parameter for each classifier based on the study case to enhance from efficiency of classifier in classification RNA-seq data, that in turn, leads to obtaining better results than using the default parameter for each classifier.

By using principal component analysis (PCA), we sought to overcome and find a suitable solution for the issue of high dimensionality. We used PCA to transform the original dimension of the PCs , and then fed each classifier with the subsets of the increasing ordered PC to test if transformed data will give us more information instead of the original data in order to evaluate if the PC will improve the performance of the classifier. Our results showed that the PC helped enhance the accuracy of the classifier.

In the end, there are no ideal feature selection criteria; instead, that depend on the type of the classifier, the kind of feature selection methods used, and the number of subsets from the most significant features.

To improve the performance of the supervised algorithms on RNA-seq data, we proposed some methods and assessed their performance in selecting the initial values to start the parameter estimation.

For the RNA-seq data, there is no one commonly accepted supervised classification method for choosing the optimal classifier that will always lead to classification with a minimal MER. Therefore, we could typically determine the most favorable classifier that would enable powerful classification with a bit error rate. Future work on this area can include identifying an ideal classifier with optimal parameters for classifying RNA-seq data. Our methods were implemented in this way, there is no monotonic behavior for each classifier. An extensive evaluation of other different classifiers will be beneficial for the future development of supervised classifiers, and using deep learning for this purpose may be worthy.

Another topic that was not investigated in this dissertation is unsupervised classification (clustering). Examining the performance of clustering methods on RNA-seq data would be interesting, particularly a comparison of the performance of supervised and unsupervised classification methods. The purpose is to evaluate whether clustering methods can re-discover what we already know about the biological classes of samples and whether they can potentially discover new properties (e.g., identify subclasses of the main classes that are defined by biologists).

As a by product of this research, we created a package called *RNAseqMVA*, which is specifically designed to study the impact of pre-processing, feature selection, and classifier parameters on the accuracy of supervised classification methods with RNA-seq data.

The aims of this dissertation are to highlight the role of machine learning algorithms in these advances and to motivate researchers on how to make supervised and unsupervised classifications for technological accomplishments in DNA sequencing technologies, specifically those related to genomics.

BIBLIOGRAPHY

Adams, M. D., Kerlavage, A. R., Fleischmann, R. D., Fuldner, R. A., Bult, C. J., Lee, N. H., Kirkness, E. F., Weinstock, K. G., Gocayne, J. D. and White, O. (1995). Initial assessment of human gene diversity and expression patterns based upon 83 million nucleotides of cDNA sequence. *Nature* 377, 3–174.

Aggarwal, C. C. and Reddy, C. K. (2013). *Data Clustering: Algorithms and Applications*. CRC Press.

Alonso, N. (2015). Big data challenges in bone research: genome-wide association studies and next-generation sequencing. *BoneKEy Reports* 16.

Anders, S. and Huber, W. (2010a). Differential expression analysis for sequence count data. *Genome Biol.* 11, R106.

Anders, S. and Huber, W. (2010b). Differential expression analysis for sequence count data. *Genome Biol.* 11, R106.

Anders, S. and Huber, W. (2010c). Differential expression analysis for sequence count data. *Genome Biol* 11, R106.

Anders, S. and Huber, W. (2010d). Differential expression analysis for sequence count data. *Genome Biology* 11, R106.

Anders, S., McCarthy, D. J., Chen, Y., Okoniewski, M., Smyth, G. K., Huber, W. and Robinson, M. D. (2013). Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. *Nat Protoc* 8, 1765–1786.

Bainbridge, M. N., Wang, M., Burgess, D. L., Kovar, C., Rodesch, M. J., D'Ascenzo, M., Kitzman, J., Wu, Y.-Q., Newsham, I., Richmond, T. A., et al. (2010). Whole exome capture in solution with 3 Gbp of data. *Genome Biology* 11, R62.

Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J., Brown, C. G., Hall, K. P., Evers, D. J., Barnes, C. L., Bignell, H. R., et al. (2008). Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* 456, 53–59.

Berkhin, P. (2006). A Survey of Clustering Data Mining Techniques. In *Grouping Multidimensional Data: Recent Advances in Clustering* (ed. Kogan, J.), Nicholas, C.), and Teboulle, M.), pp. 25–71. Berlin, Heidelberg: Springer Berlin Heidelberg.

Bhattacharyya, D. K., Kalita, J. K. and Kalita, J. K. (2013). *Network Anomaly Detection : A Machine Learning Perspective*. Chapman and Hall/CRC.

Bloom, J. S., Khan, Z., Kruglyak, L., Singh, M. and Caudy, A. A. (2009). Measuring differential gene expression by short read sequencing: quantitative comparison to 2-channel gene expression microarrays. *BMC Genomics* 10, 221.

Breiman, L. (2001a). Random Forests. *Mach. Learn.* 45, 5–32.

Breiman, L. (2001b). Random Forests. *Machine Learning* 45, 5–32.

Bullard, J. H., Purdom, E., Hansen, K. D. and Dudoit, S. (2010a). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics* 11, 94.

Bullard, J. H., Purdom, E., Hansen, K. D. and Dudoit, S. (2010b). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics* 11, 94.

Cameron, A. C. and Trivedi, P. K. (1986). Econometric models based on count data. Comparisons and applications of some estimators and tests. *Journal of Applied Econometrics* 1, 29–53.

Caruana, R. and Niculescu-Mizil, A. (2004). Data Mining in Metric Space: An Empirical Analysis of Supervised Learning Performance Criteria. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 69–78. New York, NY, USA: ACM.

Chawla, N. V., Japkowicz, N. and Kotcz, A. (2004). Editorial: Special Issue on Learning from Imbalanced Data Sets. *SIGKDD Explor. Newsl.* 6, 1–6.

Collado-Torres, L., Nellore, A., Kammers, K., Ellis, S. E., Taub, M. A., Hansen, K. D., Jaffe, A. E., Langmead, B. and Leek, J. T. (2017a). Reproducible RNA-seq analysis using recount2. *Nat. Biotechnol.* 35, 319–321.

Collado-Torres, L., Nellore, A., Kammers, K., Ellis, S. E., Taub, M. A., Hansen, K. D., Jaffe, A. E., Langmead, B. and Leek, J. T. (2017b). Reproducible RNA-seq analysis using *recount2*. *Nature Biotechnology*.

Consortium, I. H. G. S. (2001). Initial sequencing and analysis of the human genome. *Nature* 409, 860–921.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13, 21–27.

Daly, A. K. (2010). Pharmacogenetics and human genetic polymorphisms. *Biochemical Journal* 429, 435–449.

Davis, J. and Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240. ACM.

Diamantaras, K. I. and Kung, S. Y. (1996). *Principal Component Neural Networks: Theory and Applications*. New York, NY, USA: John Wiley & Sons, Inc.

Dillies, M.-A., Rau, A., Aubert, J., Hennequet-Antier, C., Jeanmougin, M., Servant, N., Keime, C., Marot, G., Castel, D., Estelle, J., et al. (2013). A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics* 14, 671–683.

Drummond, C. and Holte, R. C. (2006). Cost curves: An improved method for visualizing classifier performance. *Mach Learn* 65, 95–130.

Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., et al. (2009). Real-time DNA sequencing from single polymerase molecules. *Science* 323, 133–138.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters* 27, 861–874.

Flowers, J. M., Hazzouri, K. M., Pham, G. M., Rosas, U., Bahmani, T., Khraiwesh, B., Nelson, D. R., Jijakli, K., Abdrabu, R., Harris, E. H., et al. (2015). Whole-Genome Resequencing Reveals Extensive Natural Variation in the Model Green Alga Chlamydomonas reinhardtii. *Plant Cell* 27, 2353–2369.

Friedländer, M. R., Chen, W., Adamidi, C., Maaskola, J., Einspanier, R., Knespel, S. and Rajewsky, N. (2008). Discovering microRNAs from deep sequencing data using miRDeep. *Nature Biotechnology* 26, 407–415.

Garber, M., Grabherr, M. G., Guttman, M. and Trapnell, C. (2011). Computational methods for transcriptome annotation and quantification using RNA-seq. *Nature Methods* 8, 469–477.

García, S. and Herrera, F. (2008). Evolutionary Training Set Selection to Optimize C4.5 in Imbalanced Problems. In *2008 Eighth International Conference on Hybrid Intelligent Systems*, pp. 567–572.

Genuer, R., Poggi, J.-M. and Tuleau-Malot, C. (2010). Variable selection using Random Forests. *Pattern Recognition Letters* 31, 2225–2236.

Giveki, D., Salimi, H., Bahmanyar, G. and Khademian, Y. (2012). Automatic Detection of Diabetes Diagnosis using Feature Weighted Support Vector Machines based on Mutual Information and Modified Cuckoo Search. *arXiv:1201.2173 [cs]*.

Goksuluk, D., Zararsiz, G., Korkmaz, S., Eldem, V., Klaus, B., Ozturk, A. and Karaagaoglu, E. MLSeq: Machine Learning Interface to RNA-Seq Data. 21.

Gonzalez-Angulo, A. M., Hennessy, B. T. J. and Mills, G. B. (2010). Future of Personalized Medicine in Oncology: A Systems Biology Approach. *J Clin Oncol* 28, 2777–2783.

Gurmu, S., Rilstone, P. and Stern, S. (1999). Semiparametric estimation of count regression models11An earlier version of this paper was presented at the North American Summer Meeting of the Econometric Society in Quebec, June 1994. *Journal of Econometrics* 88, 123–150.

Hall, D. B. (2000). Zero-Inflated Poisson and Binomial Regression with Random Effects: A Case Study. *Biometrics* 56, 1030–1039.

Hand, D. J. and Till, R. J. (2001). A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning* 45, 171–186.

Hardcastle, T. J. and Kelly, K. A. (2010). baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics* 11, 422.

Hart, P. (1968). The condensed nearest neighbor rule (Corresp.). *IEEE Transactions on Information Theory* 14, 515–516.

Hoi, S. C. H., Wang, J., Zhao, P. and Jin, R. (2012). Online feature selection for mining big data. In *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining Algorithms, Systems, Programming Models and Applications - BigMine '12*, pp. 93–100. Beijing, China: ACM Press.

Hossin, M. and Sulaiman, M. . (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process* 5, 01–11.

Hossin, M., Sulaiman, M. N., Mustapha, A., Mustapha, N. and Rahmat, R. W. (2011). A hybrid evaluation metric for optimizing classifier. In *2011 3rd Conference on Data Mining and Optimization (DMO)*, pp. 165–170.

Huang, J. and Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 17, 299–310.

Igartua, C., Turner, E. H., Ng, S. B., Hodges, E., Hannon, G. J., Bhattacharjee, A., Rieder, M. J., Nickerson, D. A. and Shendure, J. (2010). Targeted Enrichment of Specific Regions in the Human Genome by Array Hybridization. *Current Protocols in Human Genetics* 66, 18.3.1-18.3.14.

Introduction to Support Vector Machines — OpenCV 2.4.13.7 documentation.

Jabeen, A., Ahmad, N. and Raza, K. (2018). Machine Learning-Based State-of-the-Art Methods for the Classification of RNA-Seq Data. In *Classification in BioApps* (ed. Dey, N.), Ashour, A. S.), and Borra, S.), pp. 133–172. Cham: Springer International Publishing.

Jaskowiak, P. A., Costa, I. G. and Campello, R. J. G. B. (2018). Clustering of RNA-Seq samples: Comparison study on cancer data. *Methods* 132, 42–49.

Jayawardana, K., Schramm, S.-J., Haydu, L., Thompson, J. F., Scolyer, R. A., Mann, G. J., Müller, S. and Yang, J. Y. H. (2015). Determination of prognosis in metastatic melanoma through integration of clinico-pathologic, mutation, mRNA, microRNA, and protein information. *International Journal of Cancer* 136, 863–874.

Jin, X. and Bie, R. Random Forest and PCA for Self-Organizing Maps based Automatic Music Genre Discrimination. 4.

Johnson, N. T., Dhroso, A., Hughes, K. J. and Korkin, D. (2018). Biological classification with RNA-seq data: Can alternatively spliced transcript expression enhance machine learning classifiers? *RNA* 24, 1119–1132.

Joshi, M. V. (2002). On Evaluating Performance of Classifiers for Rare Classes. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.(ICDM)*, p. 641.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons.

Kouichi, I., Kenichi, M. and Kousaku, O. (1994). Identification of an active gene by using large-scale cDNA sequencing. *Gene* 140, 295–296.

Lambert, D. (1992). Zero-Inflated Poisson Regression, With an Application to Defects in Manufacturing. *Technometrics* 34, 1–14.

Lavesson, N. and Davidsson, P. (2008). Generic Methods for Multi-criteria Evaluation. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pp. 541–546. Society for Industrial and Applied Mathematics.

Leinonen, R., Sugawara, H. and Shumway, M. (2011). The Sequence Read Archive. *Nucleic Acids Res* 39, D19–D21.

Levy, S., Sutton, G., Ng, P. C., Feuk, L., Halpern, A. L., Walenz, B. P., Axelrod, N., Huang, J., Kirkness, E. F., Denisov, G., et al. (2007). The Diploid Genome Sequence of an Individual Human. *PLOS Biology* 5, e254.

Li, C.-S., Lu, J.-C., Park, J., Kim, K., Brinkley, P. A. and Peterson, J. P. (1999). Multivariate Zero-Inflated Poisson Models and Their Applications. *Technometrics* 41, 29–38.

Li, R., Fan, W., Tian, G., Zhu, H., He, L., Cai, J., Huang, Q., Cai, Q., Li, B., Bai, Y., et al. (2010). The sequence and *de novo* assembly of the giant panda genome. *Nature* 463, 311–317.

Liu, S., Lin, L., Jiang, P., Wang, D. and Xing, Y. (2011). A comparison of RNA-Seq and high-density exon array for detecting differential gene expression between closely related species. *Nucleic Acids Res* 39, 578–588.

Lonsdale, J., Thomas, J., Salvatore, M., Phillips, R., Lo, E., Shad, S., Hasz, R., Walters, G., Garcia, F., Young, N., et al. (2013). The Genotype-Tissue Expression (GTEx) project. *Nature Genetics* 45, 580–585.

Love, M. I., Huber, W. and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* 15,.

Luts, J., Ojeda, F., Van de Plas, R., De Moor, B., Van Huffel, S. and Suykens, J. A. K. (2010). A tutorial on support vector machine-based methods for classification problems in chemometrics. *Analytica Chimica Acta* 665, 129–145.

Machine Learning-Based State-Of-The-Art Methods For The Classification Of RNA-Seq Data | bioRxiv.

Marguerat, S. and Bähler, J. (2010). RNA-seq: from technology to biology. *Cell. Mol. Life Sci.* 67, 569–579.

Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y.-J., Chen, Z., et al. (2005a). Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437, 376–380.

Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y.-J., Chen, Z., et al. (2005b). Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437, 376–380.

Marioni, J. C., Mason, C. E., Mane, S. M., Stephens, M. and Gilad, Y. (2008a). RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res.* 18, 1509–1517.

Marioni, J. C., Mason, C. E., Mane, S. M., Stephens, M. and Gilad, Y. (2008b). RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res.* 18, 1509–1517.

Marisa, L., Reyniès, A. de, Duval, A., Selves, J., Gaub, M. P., Vescovo, L., Etienne-Grimaldi, M.-C., Schiappa, R., Guenot, D., Ayadi, M., et al. (2013). Gene Expression Classification of Colon Cancer into Molecular Subtypes: Characterization, Validation, and Prognostic Value. *PLOS Medicine* 10, e1001453.

Mattick, J. S. and Makunin, I. V. (2006). Non-coding RNA. *Hum. Mol. Genet.* 15 Spec No 1, R17-29.

McLachlan, G., Do, K.-A. and Ambroise, C. (2005). *Analyzing Microarray Gene Expression Data*. John Wiley & Sons.

Metzker, M. L. (2010). Sequencing technologies — the next generation. *Nature Reviews Genetics* 11, 31–46.

Meyer, D. Support Vector Machines. 8.

Mooney, M., Bond, J., Monks, N., Eugster, E., Cherba, D., Berlinski, P., Kamerling, S., Marotti, K., Simpson, H., Rusk, T., et al. (2013). Comparative RNA-Seq and Microarray Analysis of Gene Expression Changes in B-Cell Lymphomas of Canis familiaris. *PLOS ONE* 8, e61088.

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L. and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods* 5, 621–628.

Natsoulis, G., El Ghaoui, L., Lanckriet, G. R. G., Tolley, A. M., Leroy, F., Dunlea, S., Eynon, B. P., Pearson, C. I., Tugendreich, S. and Jarnagin, K. (2005). Classification of a large microarray data set: algorithm comparison and analysis of drug signatures. *Genome Res.* 15, 724–736.

Nellore, A., Jaffe, A. E., Fortin, J.-P., Alquicira-Hernández, J., Collado-Torres, L., Wang, S., Phillips III, R. A., Karbhari, N., Hansen, K. D., Langmead, B., et al. (2016). Human splicing diversity and the extent of unannotated splice junctions across human RNA-seq samples on the Sequence Read Archive. *Genome Biology* 17, 266.

Nellore, A., Collado-Torres, L., Jaffe, A. E., Alquicira-Hernández, J., Wilks, C., Pritt, J., Morton, J., Leek, J. T., Langmead, B. and Ratsch, G. (2017). Rail-RNA: scalable analysis of RNA-seq splicing and coverage. *Bioinformatics* 33, 4033–4040.

Ng, P. C., Murray, S. S., Levy, S. and Venter, J. C. (2009). An agenda for personalized medicine. *Nature*.

Novelli, G., Predazzi, I. M., Mango, R., Romeo, F. and Mehta, J. L. (2010). Role of genomics in cardiovascular medicine. *World J Cardiol* 2, 428–436.

Oshlack, A. and Wakefield, M. J. (2009). Transcript length bias in RNA-seq data confounds systems biology. *Biology Direct* 4, 14.

Pepke, S., Wold, B. and Mortazavi, A. (2009). Computation for ChIP-seq and RNA-seq studies. *Nat Methods* 6, S22–S32.

Prati, R. C., Batista, G. E. A. P. A. and Monard, M. C. (2011). A Survey on Graphical Methods for Classification Predictive Performance Evaluation. *IEEE Transactions on Knowledge and Data Engineering* 23, 1601–1618.

Provost, F. and Domingos, P. (2003). Tree Induction for Probability-Based Ranking. *Machine Learning* 52, 199–215.

Pussegoda, K. A. (2010). Exome sequencing: locating causative genes in rare disorders. *Clinical Genetics* 78, 32–33.

R: a language and environment for statistical computing.

Ranawana, R. and Palade, V. (2006). Optimized Precision - A New Measure for Classifier Performance Evaluation. In *2006 IEEE International Conference on Evolutionary Computation*, pp. 2254–2261.

Rehman, A. U., Morell, R. J., Belyantseva, I. A., Khan, S. Y., Boger, E. T., Shahzad, M., Ahmed, Z. M., Riazuddin, S., Khan, S. N., Riazuddin, S., et al. (2010). Targeted Capture and Next-Generation Sequencing Identifies C9orf75, Encoding Taperin, as the Mutated Gene in Nonsyndromic Deafness DFNB79. *The American Journal of Human Genetics* 86, 378–388.

Rios, J., Stein, E., Shendure, J., Hobbs, H. H. and Cohen, J. C. (2010). Identification by whole-genome resequencing of gene defect responsible for severe hypercholesterolemia. *Hum Mol Genet* 19, 4313–4318.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S. and Vert, J.-P. (2018). A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature Communications* 9,.

Robinson, M. D. and Oshlack, A. (2010a). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.* 11, R25.

Robinson, M. D. and Oshlack, A. (2010b). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.* 11, R25.

Robinson, M. D. and Smyth, G. K. (2007). Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics* 23, 2881–2887.

Robinson, M. D., McCarthy, D. J. and Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26, 139–140.

Rosset, S. (2004). Model Selection via the AUC. In *Proceedings of the Twenty-first International Conference on Machine Learning*, pp. 89–. New York, NY, USA: ACM.

Rudloff, U., Bhanot, U., Gerald, W., Klimstra, D. S., Jarnagin, W. R., Brennan, M. F. and Allen, P. J. (2010). Biobanking of Human Pancreas Cancer Tissue: Impact of Ex-Vivo Procurement Times on RNA Quality. *Ann Surg Oncol* 17,.

Schena, M., Shalon, D., Davis, R. W. and Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 270, 467–470.

Schurch, N. J., Schofield, P., Gierliński, M., Cole, C., Sherstnev, A., Singh, V., Wrobel, N., Gharbi, K., Simpson, G. G., Owen-Hughes, T., et al. (2016). How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use? *RNA* 22, 839–851.

Seo, H., Park, Y., Min, B. J., Seo, M. E. and Kim, J. H. (2017). Evaluation of exome variants using the Ion Proton Platform to sequence error-prone regions. *PLOS ONE* 12, e0181304.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Shendure, J. (2008). The beginning of the end for microarrays? *Nat. Methods* 5, 585–587.

Shiraki, T., Kondo, S., Katayama, S., Waki, K., Kasukawa, T., Kawaji, H., Kodzius, R., Watahiki, A., Nakamura, M., Arakawa, T., et al. (2003). Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. *PNAS* 100, 15776–15781.

Strbenac, D., Mann, G. J., Yang, J. Y. H. and Ormerod, J. T. (2016). Differential distribution improves gene selection stability and has competitive classification performance for patient survival. *Nucleic Acids Res* 44, e119–e119.

Sultan, M., Schulz, M. H., Richard, H., Magen, A., Klingenhoff, A., Scherf, M., Seifert, M., Borodina, T., Soldatov, A., Parkhomchuk, D., et al. (2008). A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science* 321, 956–960.

SummarizedExperiment-class function | R Documentation.

Suykens, J. A. K. (2009). Support vector machines and kernel-based learning for dynamical systems modelling. *IFAC Proceedings Volumes* 42, 1029–1037.

Tan, P.-N., Steinbach, M. and Kumar, V. Introduction to Data Mining. 169.

Tan, M., Tsang, I. W. and Wang, L. Towards Ultrahigh Dimensional Feature Selection for Big Data. 59.

Tang, M., Sun, J., Shimizu, K. and Kadota, K. (2015). Evaluation of methods for differential expression analysis on multi-group RNA-seq count data. *BMC Bioinformatics* 16, 361.

Team*, T. F. C. and the R. G. E. R. G. P. I. & I. (2002). Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cDNAs. *Nature* 420, 563–573.

Teer, J. K. and Mullikin, J. C. (2010). Exome sequencing: the sweet spot before whole genomes. *Hum Mol Genet* 19, R145–R151.

Tewhey, R., Nakano, M., Wang, X., Pabón-Peña, C., Novak, B., Giuffre, A., Lin, E., Happe, S., Roberts, D. N., LeProust, E. M., et al. (2009). Enrichment of sequencing targets from the human genome by solution hybridization. *Genome Biology* 10, R116.

The Cancer Genome Atlas Research Network, Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R. M., Ozenberger, B. A., Ellrott, K., Shmulevich, I., Sander, C. and Stuart, J. M. (2013). The Cancer Genome Atlas Pan-Cancer analysis project. *Nature Genetics* 45, 1113–1120.

Thompson, J. F. and Steinmann, K. E. (2010). Single molecule sequencing with a HeliScope genetic analysis system. *Curr Protoc Mol Biol* Chapter 7, Unit7.10.

Thompson, K. L., Pine, P. S., Rosenzweig, B. A., Turpaz, Y. and Retief, J. (2007). Characterization of the effect of sample quality on high density oligonucleotide microarray data using progressively degraded rat liver RNA. *BMC Biotechnol.* 7, 57.

Time-Series Analysis: Wearable Devices using DTW and kNN *SFL Scientific - Data Science Consulting & Artificial Intelligence*.

Torres, L. C. (2017). Reproducible RNA-seq analysis using recount2. *L. Collado-Torres*.

Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M. J., Salzberg, S. L., Wold, B. J. and Pachter, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* 28, 511–515.

Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., Kelley, D. R., Pimentel, H., Salzberg, S. L., Rinn, J. L. and Pachter, L. (2012). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nat Protoc* 7, 562–578.

Tu, W. and Zhou, X.-H. (1999). A Wald test comparing medical costs based on log-normal distributions with zero valued costs. *Statistics in Medicine* 18, 2749–2761.

Tucker, T., Marra, M. and Friedman, J. M. (2009). Massively Parallel Sequencing: The Next Big Thing in Genetic Medicine. *The American Journal of Human Genetics* 85, 142–154.

Tyler, A. D., Christianson, S., Knox, N. C., Mabon, P., Wolfe, J., Domselaar, G. V., Graham, M. R. and Sharma, M. K. (2016). Comparison of Sample Preparation Methods Used for the Next-Generation Sequencing of Mycobacterium tuberculosis. *PLOS ONE* 11, e0148676.

Vallender, E. J. (2011). Expanding whole exome resequencing into non-human primates. *Genome Biology* 12, R87.

Vapnik, V. (2000). *The Nature of Statistical Learning Theory*. 2nd ed. New York: Springer-Verlag.

Venter, J. C., Levy, S., Stockwell, T., Remington, K. and Halpern, A. (2003). Massive parallelism, randomness and genomic advances. *Nature Genetics* 33, 219–227.

Vieira, A. M. C., Hinde, J. P. and Demetrio, C. G. B. (2000). Zero-inflated proportion data models applied to a biological control assay. *Journal of Applied Statistics* 27, 373–389.

Voelkerding, K. V., Dames, S. and Durtschi, J. D. (2010). Next generation sequencing for clinical diagnostics-principles and application to targeted resequencing for hypertrophic cardiomyopathy: a paper from the 2009 William Beaumont Hospital Symposium on Molecular Pathology. *J Mol Diagn* 12, 539–551.

Vuk, M. and Curk, T. ROC Curve, Lift Chart and Calibration Plot. 20.

Walsh, T., Shahin, H., Elkan-Miller, T., Lee, M. K., Thornton, A. M., Roeb, W., Abu Rayyan, A., Loulus, S., Avraham, K. B., King, M.-C., et al. (2010). Whole Exome Sequencing and Homozygosity Mapping Identify Mutation in the Cell Polarity Protein GPSM2 as the Cause of Nonsyndromic Hearing Loss DFNB82. *The American Journal of Human Genetics* 87, 90–94.

Wang, J., Wang, W., Li, R., Li, Y., Tian, G., Goodman, L., Fan, W., Zhang, J., Li, J., Zhang, J., et al. (2008). The diploid genome sequence of an Asian individual. *Nature* 456, 60–65.

Wang, Z., Gerstein, M. and Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.* 10, 57–63.

Wang, L., Li, P. and Brutnell, T. P. (2010). Exploring plant transcriptomes using ultra high-throughput sequencing. *Brief Funct Genomics* 9, 118–128.

Wheeler, D. A., Srinivasan, M., Egholm, M., Shen, Y., Chen, L., McGuire, A., He, W., Chen, Y.-J., Makhijani, V., Roth, G. T., et al. (2008). The complete genome of an individual by massively parallel DNA sequencing. *Nature* 452, 872–876.

Wilson, S. W. (2001). Mining Oblique Data with XCS. In *Advances in Learning Classifier Systems* (ed. Luca Lanzi, P.), Stolzmann, W.), and Wilson, S. W.), pp. 158–174. Springer Berlin Heidelberg.

Wilusz, J. E., Sunwoo, H. and Spector, D. L. (2009). Long noncoding RNAs: functional surprises from the RNA world. *Genes Dev.* 23, 1494–1504.

Young, M. D., Wakefield, M. J., Smyth, G. K. and Oshlack, A. (2010). Gene ontology analysis for RNA-seq: accounting for selection bias. *Genome Biology* 11, R14.

Zhang, T., Ramakrishnan, R. and Livny, M. (1996). BIRCH: An Efficient Data Clustering Method for Very Large Databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pp. 103–114. New York, NY, USA: ACM.

Zhang, J., Chiodini, R., Badr, A. and Zhang, G. (2011). The impact of next-generation sequencing on genomics. *Journal of Genetics and Genomics* 38, 95–109.

Zywicki, M., Bakowska-Zywicka, K. and Polacek, N. (2012). Revealing stable processing products from ribosome-associated small RNAs by deep-sequencing data analysis. *Nucleic Acids Res* 40, 4013–4024.

APPENDICES

Appendix A. Supplementary Figures

For the sake of completeness, we provide hereafter the figures for each one of the 7 study cases. A subset of these is used and discussed in the manuscript.

A1. Distribution for the raw count data in each dataset from selected datasets

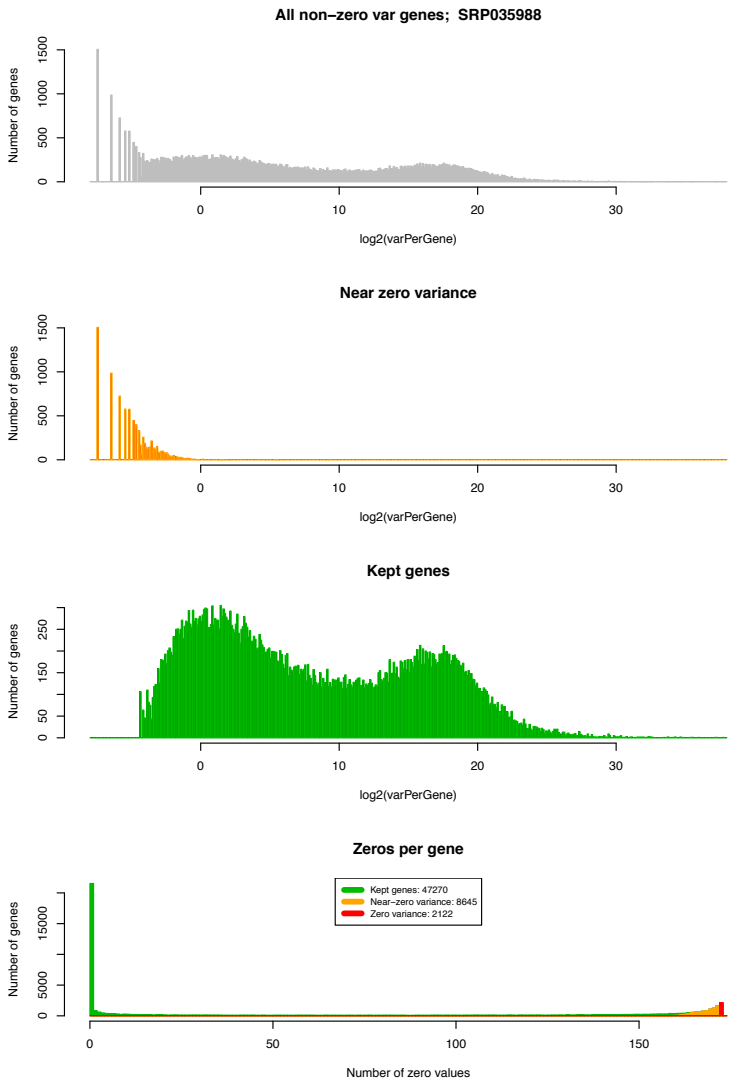


Figure 34 Impact of variance-based gene filtering on the study case *Psoriasis* (SRP035988). The histograms indicate the distribution of variances (abscissa) per gene (ordinates) in the raw data (**top panel, grey**), in the genes discarded by the near-zero filter (**second histogram, orange**), and in the genes kept after filtering (**third histogram, green**). The **bottom** histogram shows the number of genes (ordinate) as a function of the number of samples (abscissa) having a zero value for these genes.

Figure 35 Impact of variance-based gene filtering on the study case *Human Leukemia* (SRP056295).
The histograms indicate the distribution of variances (abscissa) per gene (ordinates) in the raw data (**top panel, grey**), in the genes discarded by the near-zero filter (**second histogram, orange**), and in the genes kept after filtering (**third histogram, green**). The **bottom** histogram shows the number of genes (ordinate) as a function of the number of samples (abscissa) having a zero value for these genes.

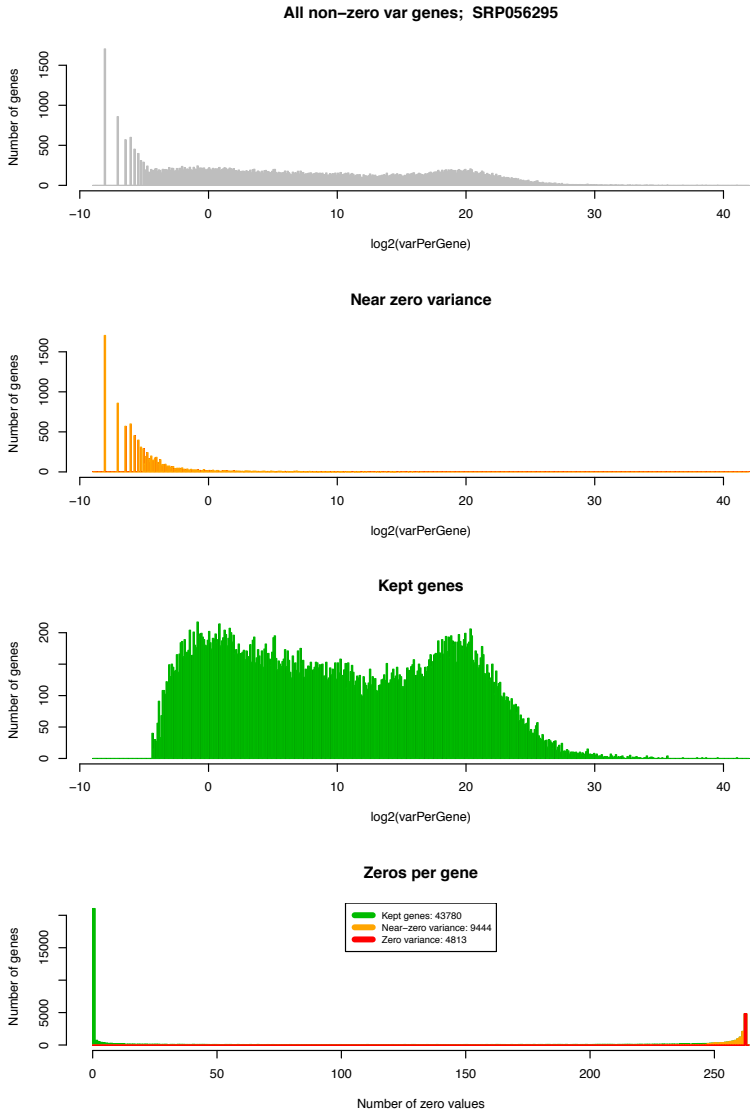


Figure 36 Impact of variance-based gene filtering on the study case *Cellular Complexity of the adult & fetal human brain* (SRP057196).
The histograms indicate the distribution of variances (abscissa) per gene (ordinates) in the raw data (**top panel, grey**), in the genes discarded by the near-zero filter (**second histogram, orange**), and in the genes kept after filtering (**third histogram, green**). The **bottom** histogram shows the number of genes (ordinate) as a function of the number of samples (abscissa) having a zero value for these genes.

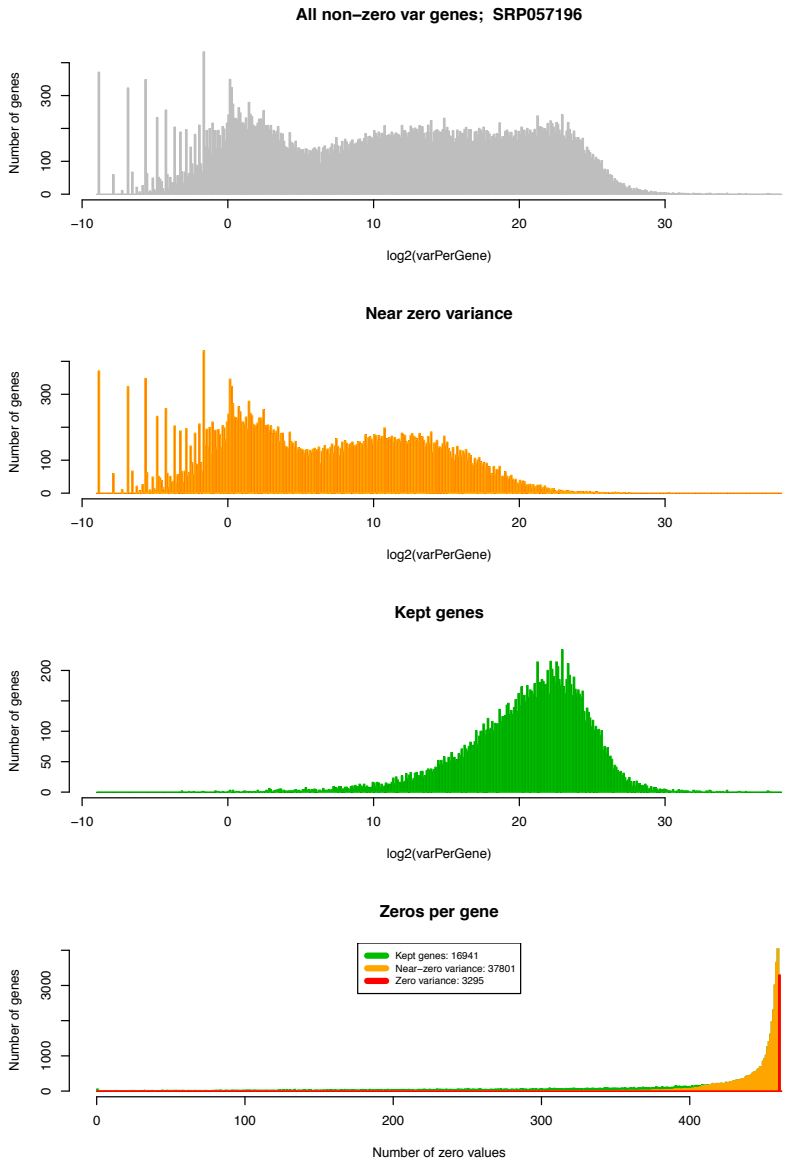


Figure 37 Impact of variance-based gene filtering on the study case *Cancer disease types* (SRP061240).
The histograms indicate the distribution of variances (abscissa) per gene (ordinates) in the raw data (**top panel, grey**), in the genes discarded by the near-zero filter (**second histogram, orange**), and in the genes kept after filtering (**third histogram, green**). The **bottom** histogram shows the number of genes (ordinate) as a function of the number of samples (abscissa) having a zero value for these genes.

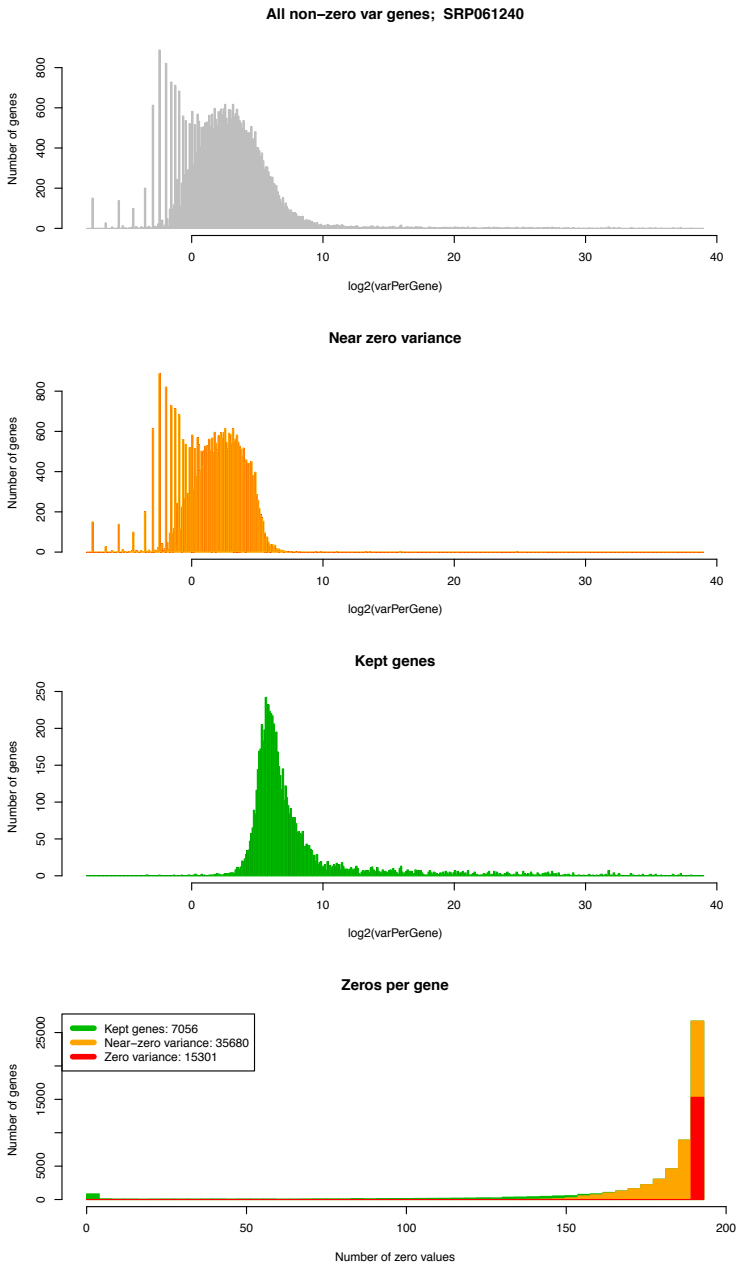


Figure 38 Impact of variance-based gene filtering on the study case *Lupus erythematosus* (SRP062966). The histograms indicate the distribution of variances (abscissa) per gene (ordinates) in the raw data (**top panel, grey**), in the genes discarded by the near-zero filter (**second histogram, orange**), and in the genes kept after filtering (**third histogram, green**). The **bottom** histogram shows the number of genes (ordinate) as a function of the number of samples (abscissa) having a zero value for these genes.

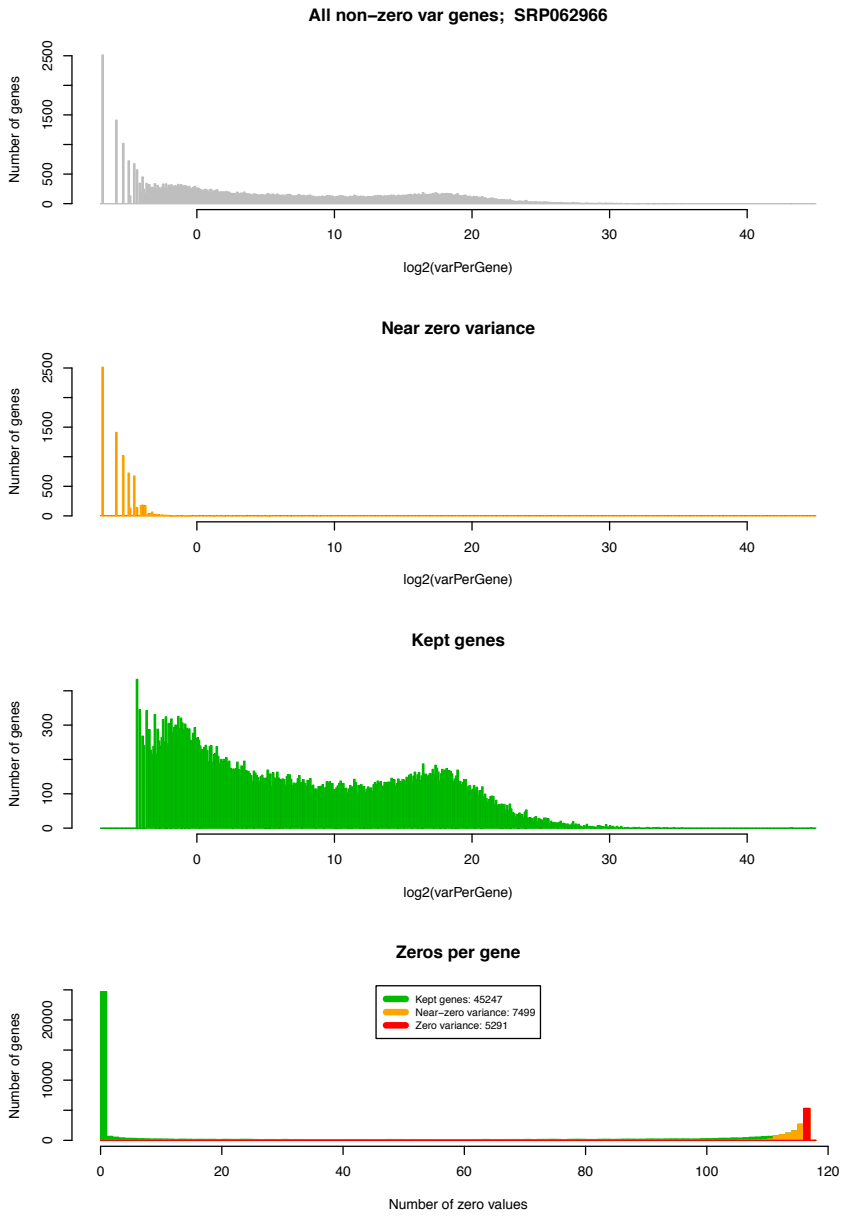
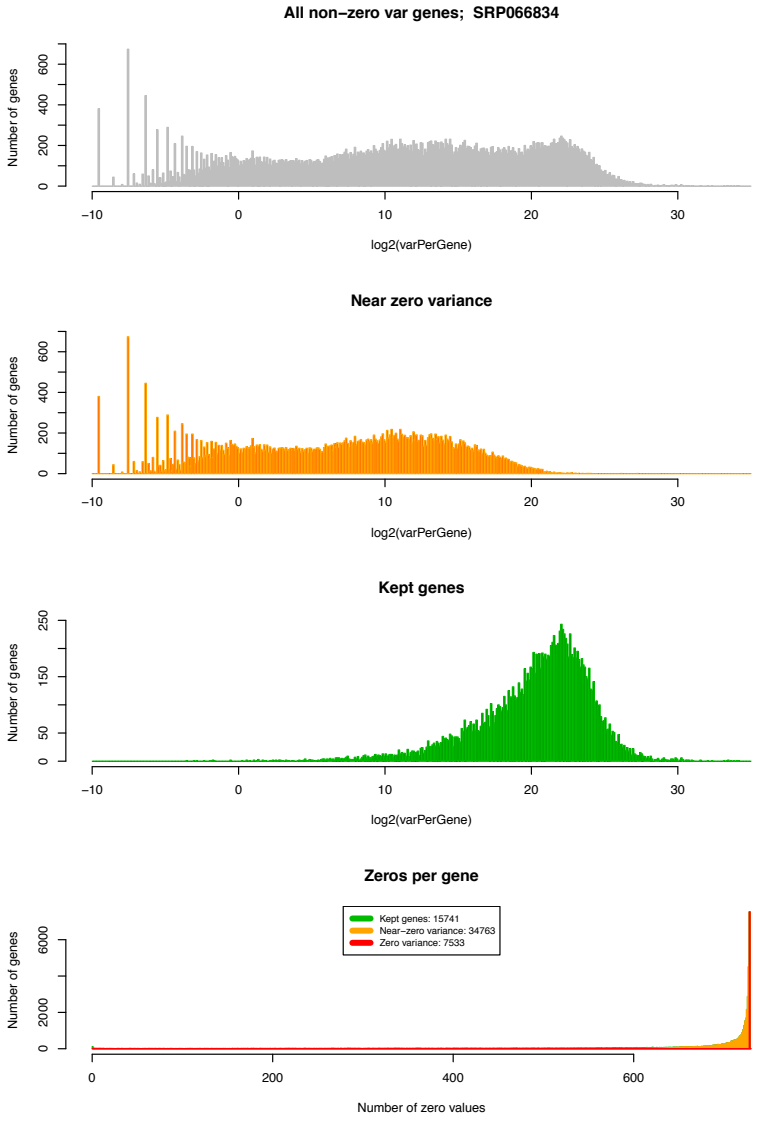


Figure 39 Impact of variance-based gene filtering on the study case *Cerebral organoids and fetal neocortex* (SRP066834).
The histograms indicate the distribution of variances (abscissa) per gene (ordinates) in the raw data (**top panel, grey**), in the genes discarded by the near-zero filter (**second histogram, orange**), and in the genes kept after filtering (**third histogram, green**). The **bottom** histogram shows the number of genes (ordinate) as a function of the number of samples (abscissa) having a zero value for these genes.



A2. Impact of Normalization processes for the 7 study cases

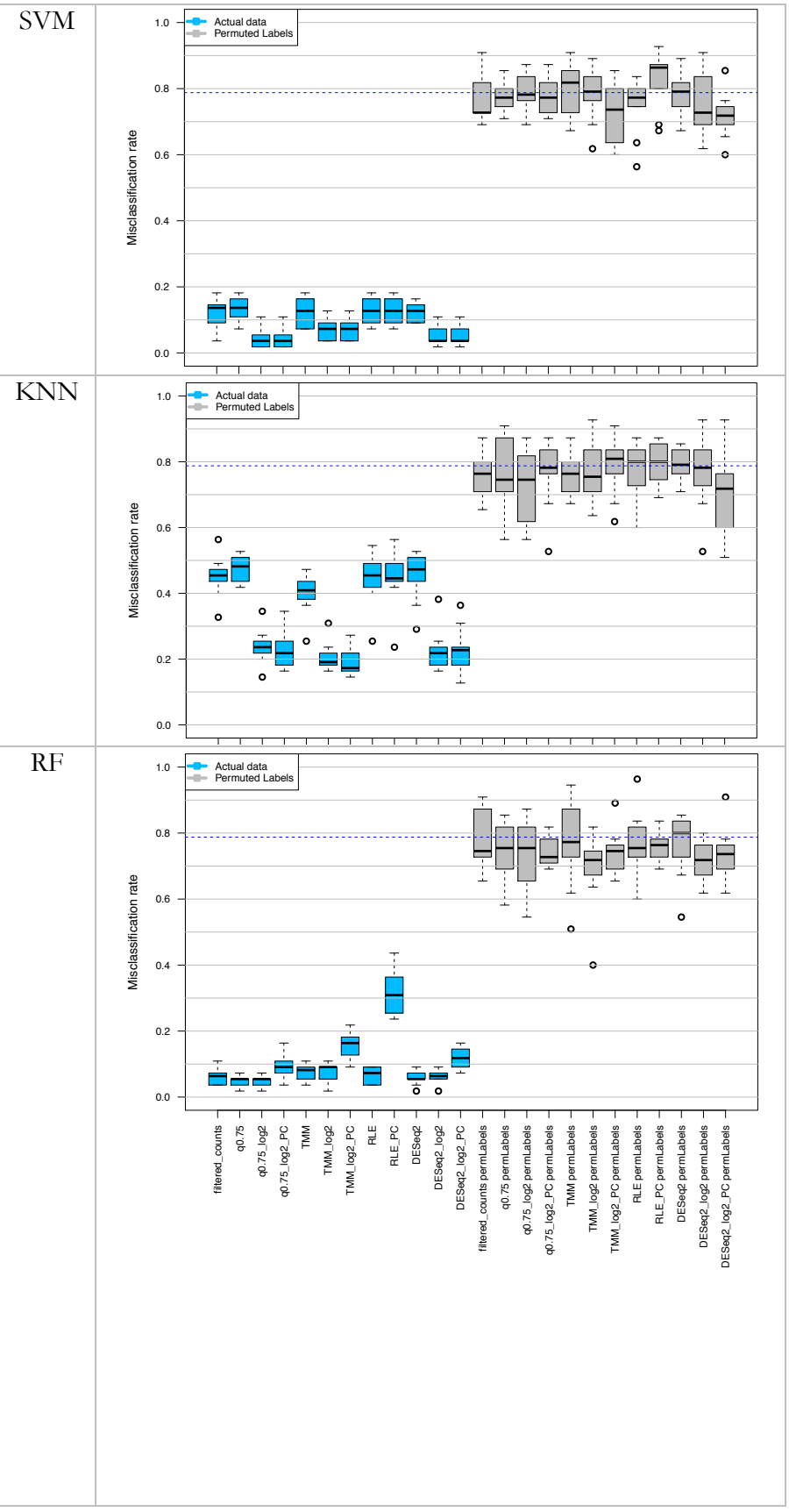


Figure 40. Impact of Normalization. on the misclassification error rate of classifiers on the *Human breast cancer* study case (SRP42620). The ordinate indicates the misclassification error rates. Each boxplot corresponds to one testing-training experiment (10 iterations) and one particular preprocessing method. **Abbreviations:** q0.75: third quantile normalization; log2: log2-transformed normalized data; PC: principal components derived from log2-transformed data; RLE: Relative Log Expression; DESeq2: normalization method from DESeq2; permLabels: random permutation of the sample labels, usedto estimate the random expectation for the misclassification error rate. **Blue boxes:** result from the analysis of the actual datasets. **Grey boxes:** random expectation estimated by permuting class labels during the training and testing. **Top:** Support Vector Machines (SVM); **Middle:** K-nearest neighbors (KNN); **Bottom:** Random Forest (RF).

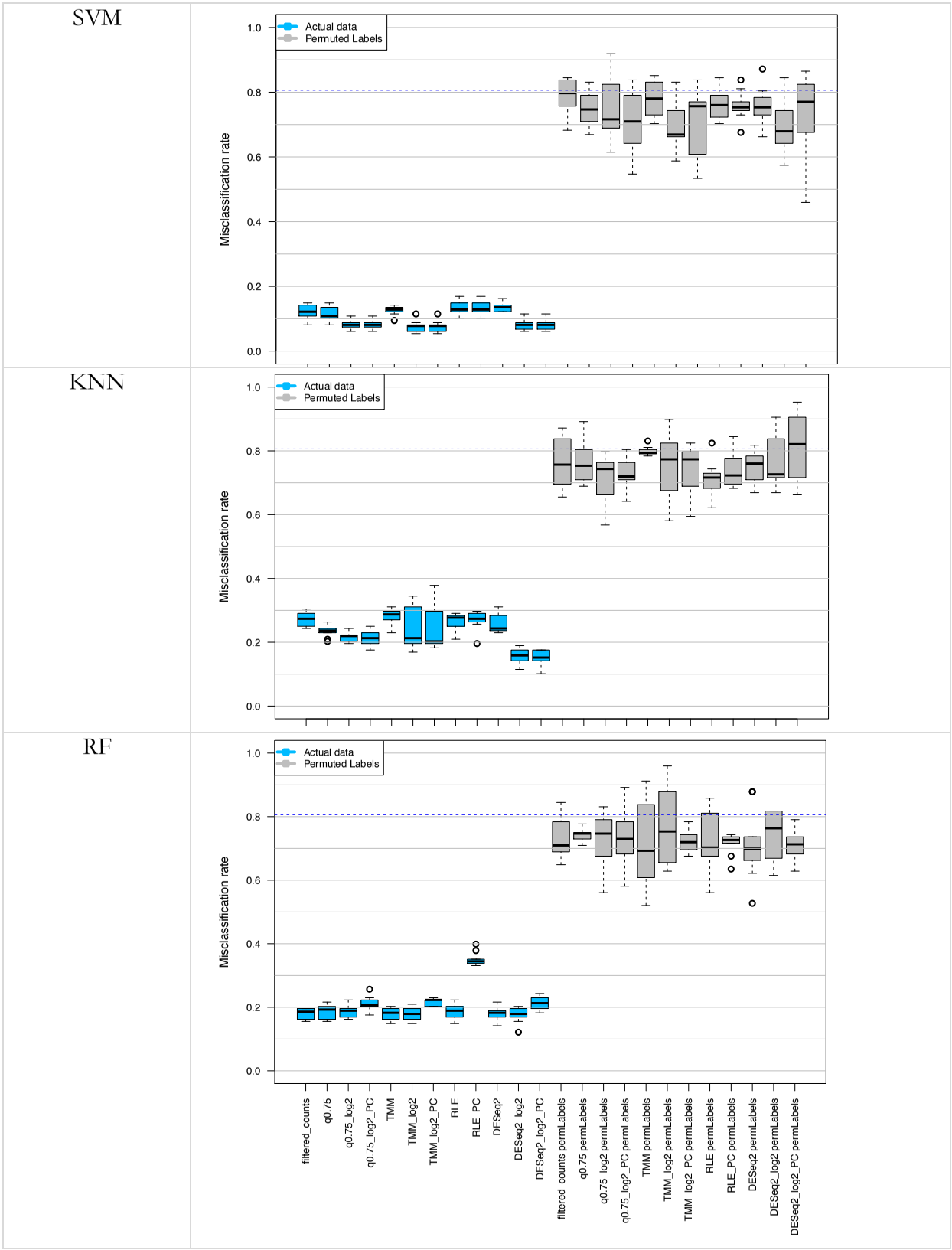


Figure 41 Impact of Normalization. on the misclassification error rate of classifiers on the *Cellular Complexity of the adult & fetal human brain* study case (SRP057196). Legend: see Figure 40.

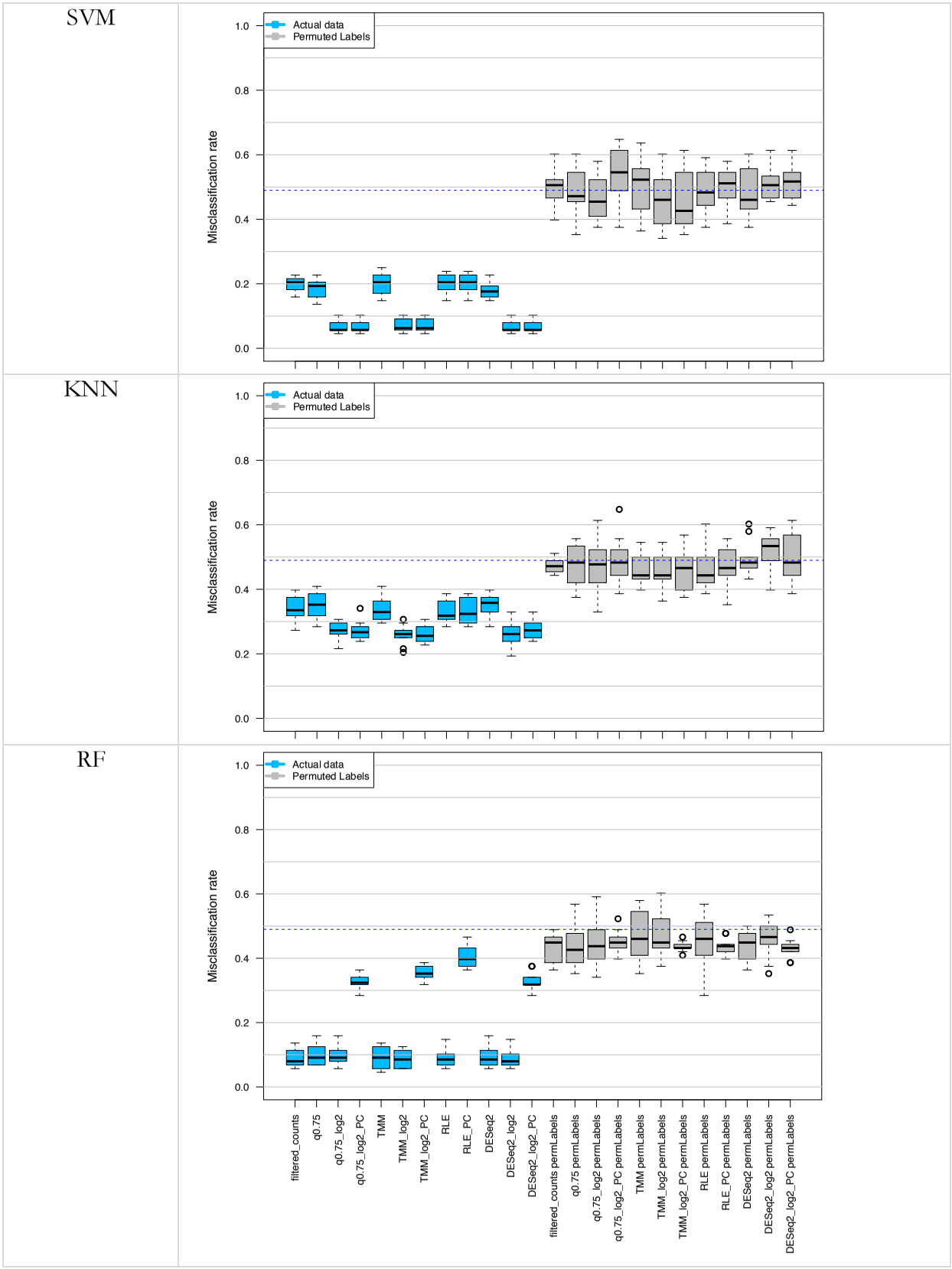


Figure 42 **Impact of Normalization on the misclassification error rate of classifiers on the *Human Acute Myeloid Leukemia* study case (SRP056295).** Legend: see Figure 40.

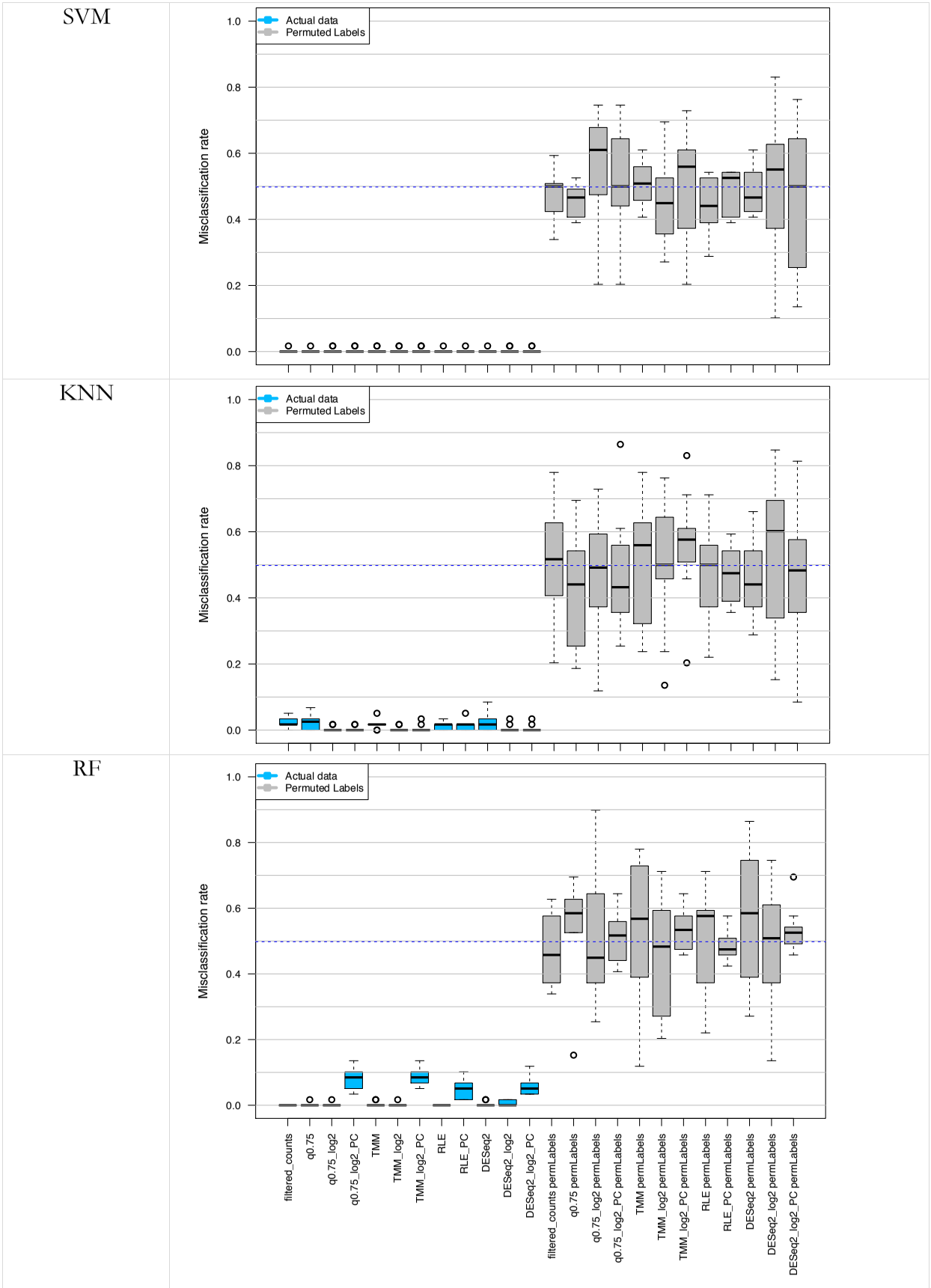


Figure 43 **Impact of Normalization on the misclassification error rate of classifiers on the *Psoriasis* study case (SRP035988).** Legend: see Figure 40.

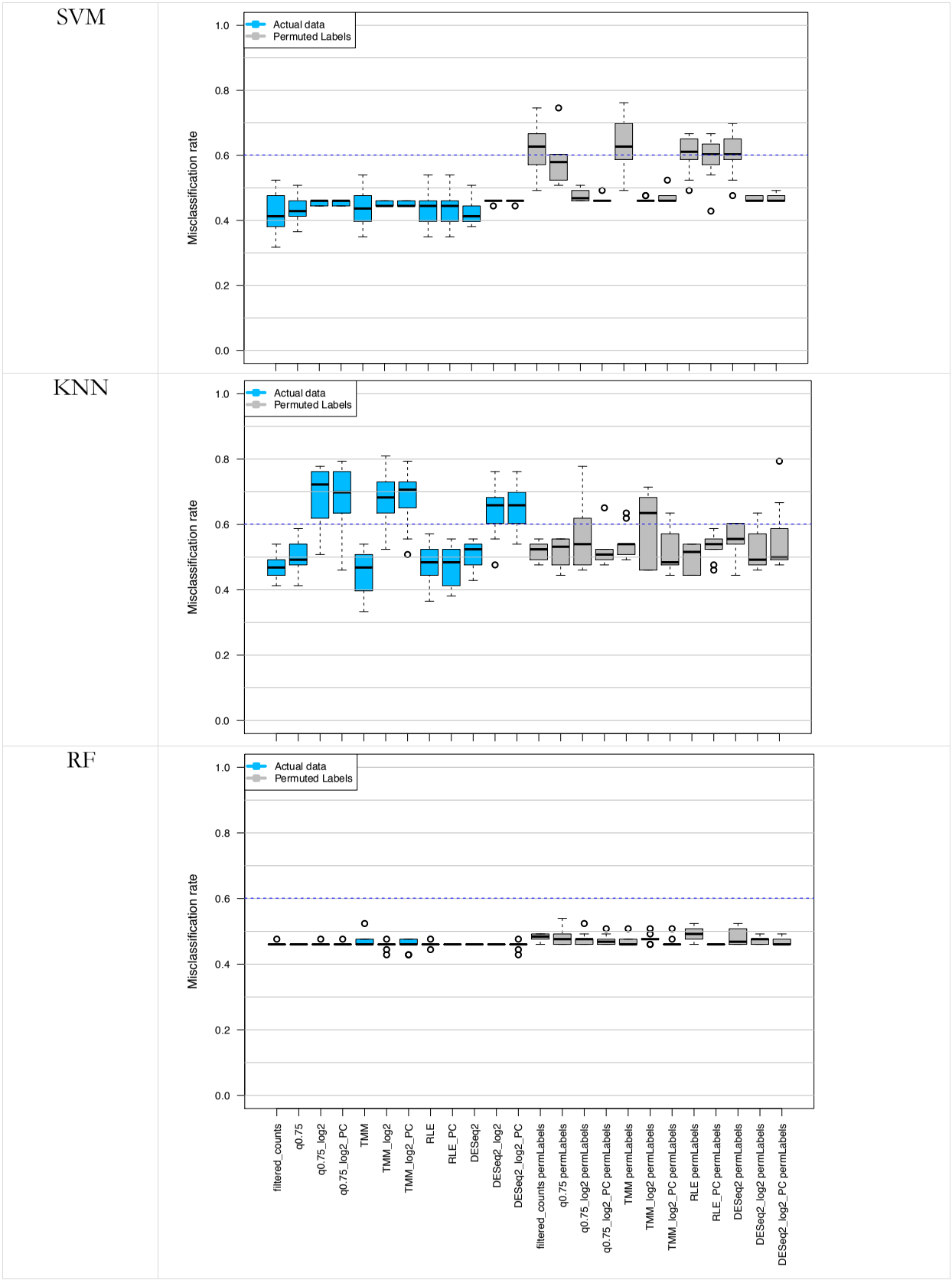


Figure 44 **Impact of Normalization on the misclassification error rate of classifiers on the *Cancer disease types* study case (SRP061240).** Legend: see Figure 40.

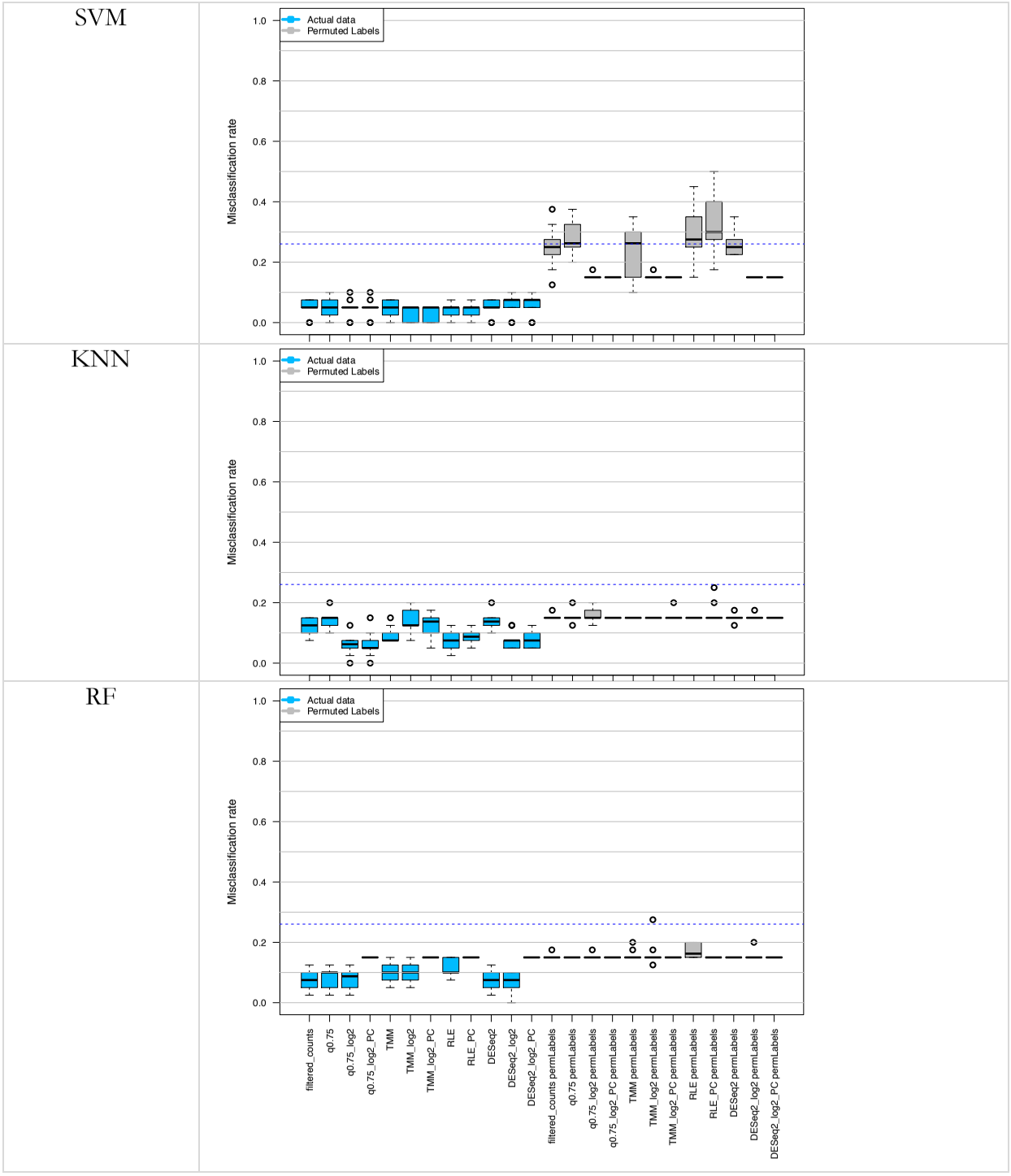


Figure 45 **Impact of Normalization on the misclassification error rate of classifiers on the *Blood Disease* study case (SRP062966).** Legend: see Figure 40.

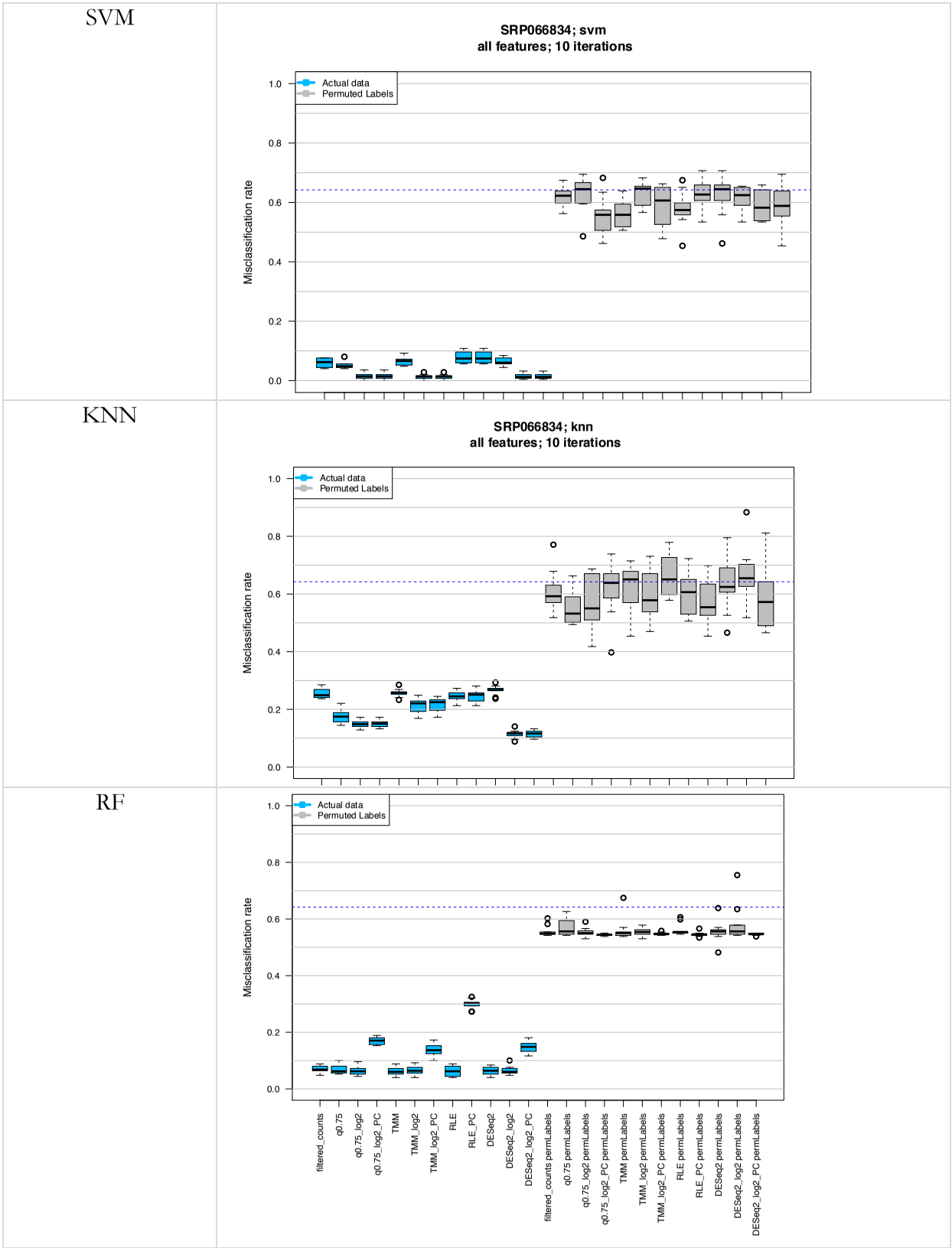


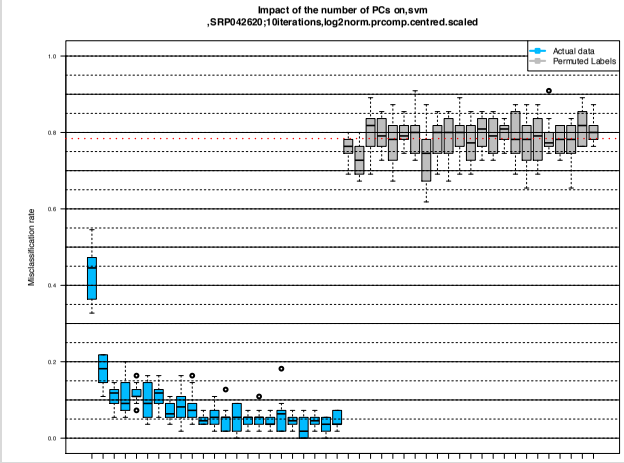
Figure 46 **Impact of Normalization on the misclassification error rate of classifiers on the *Cerebral organoids and fetal neocortex* study case (SRP066834).** Legend: see Figure 40.

A3. Impact of feature selection on classifier Accuracy

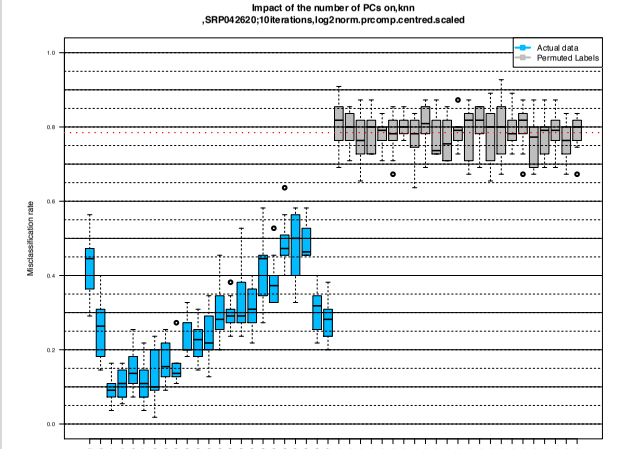
A3.1. Feature selection based on Principal Components

Figure 47 Impact of the selection of principal components on the accuracy of classifiers for the *Breast cancer* (SRP042620) study case. The ordinate indicates the misclassification error rates (MER). Each boxplot corresponds to one testing-training experiment (10 iterations) and one particular feature selection based on PCs method. Abbreviations: prccomp-nb_of_PCs_3: log2-transformed normalized counts are reduced through R prcomp() function. The suffix number 3 indicates the number of retained top PCs. **Blue boxes**: result from the analysis of the actual datasets. **Grey boxes**: random expectation estimated by permuting class labels during the training. **Top**: Support Vector Machines (SVM); **Middle**: K-nearest neighbors (KNN); **Bottom**: Random Forest (RF).

SVM



KNN



RF

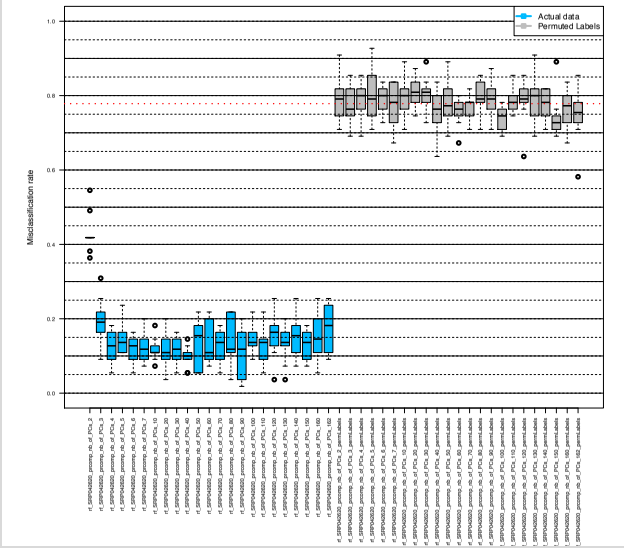
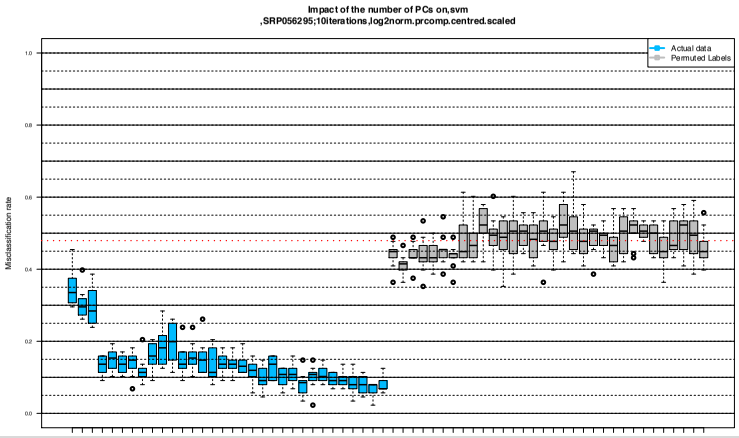
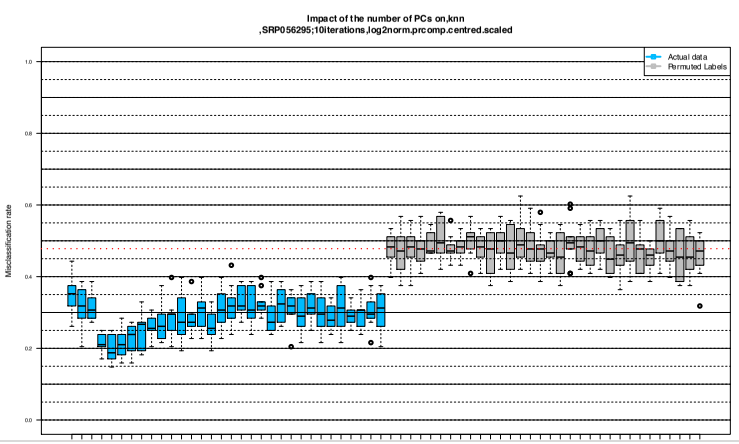


Figure 48 Impact of principal components on the performances of classifiers for the *Human Leukemia* (SRP056295). Legend: see **Figure 47**.

SVM



KNN



RF

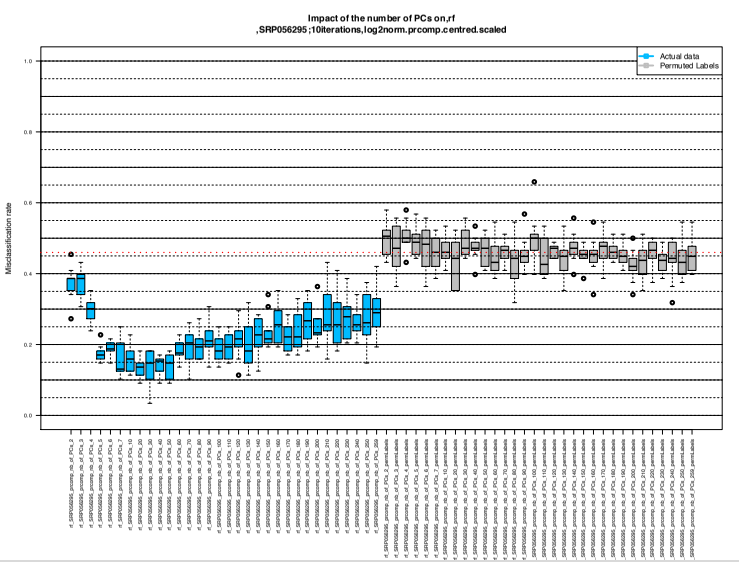
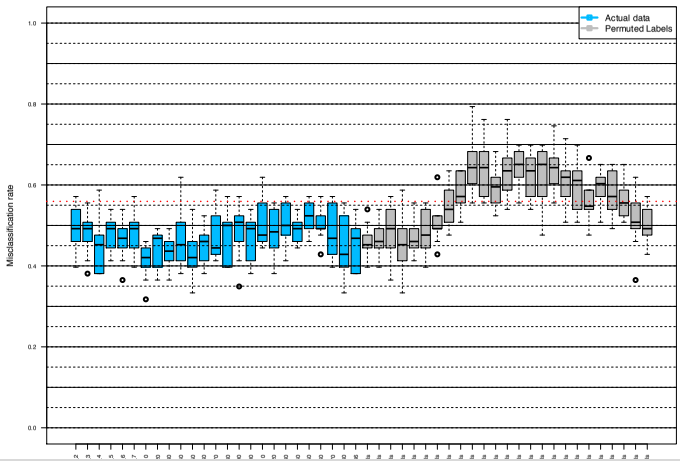
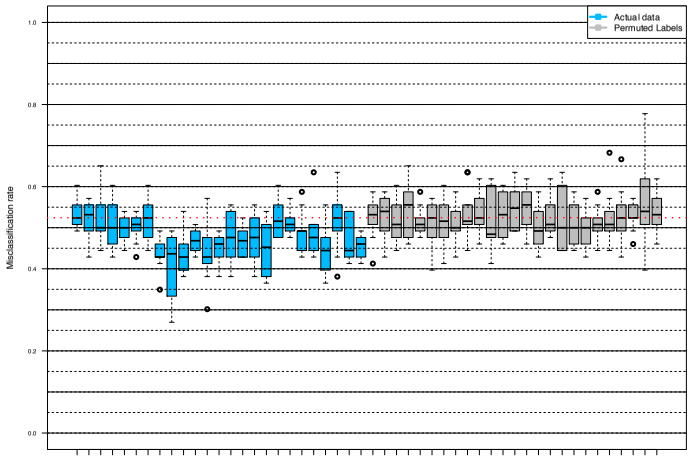


Figure 49 Impact of the of principal components on the performances of classifiers for the *Cancer disease types* (SRP061240). Legend: see **Figure 47**.

SVM



KNN



RF

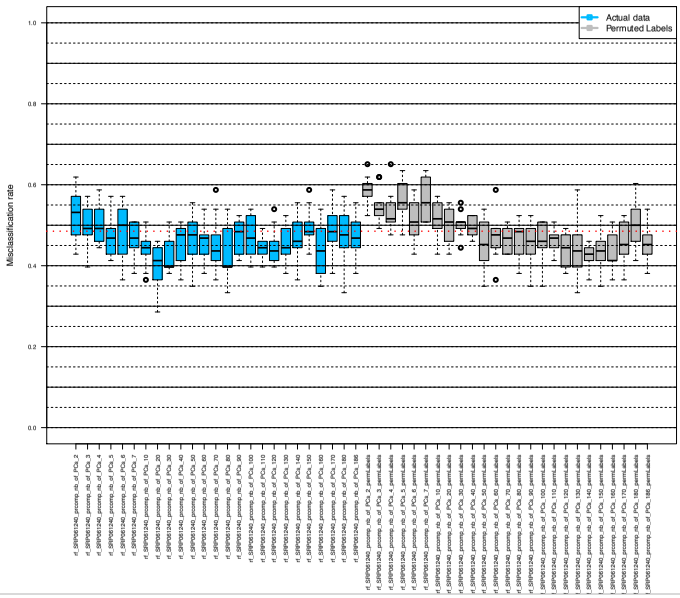
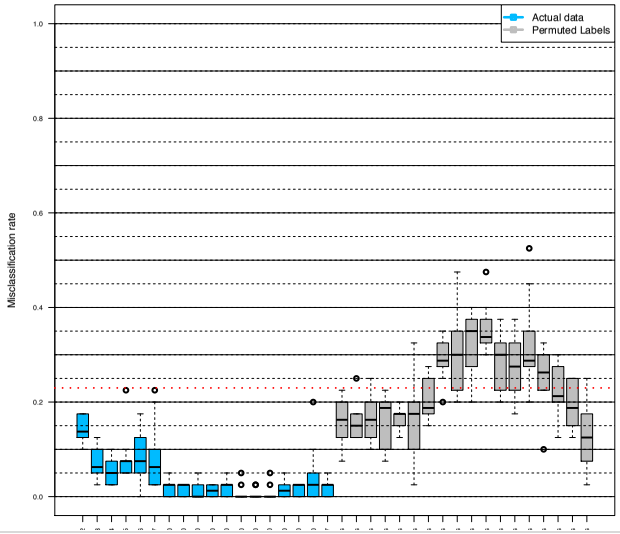
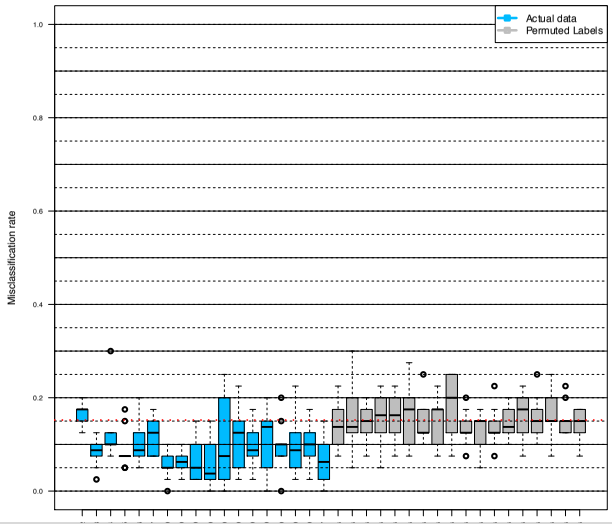


Figure 50 Impact of the features selection of principal components on the performances of classifiers for classifiers on the Blood Disease (SRP062966).
Legend: see **Figure 47**.

SVM



KNN



RF

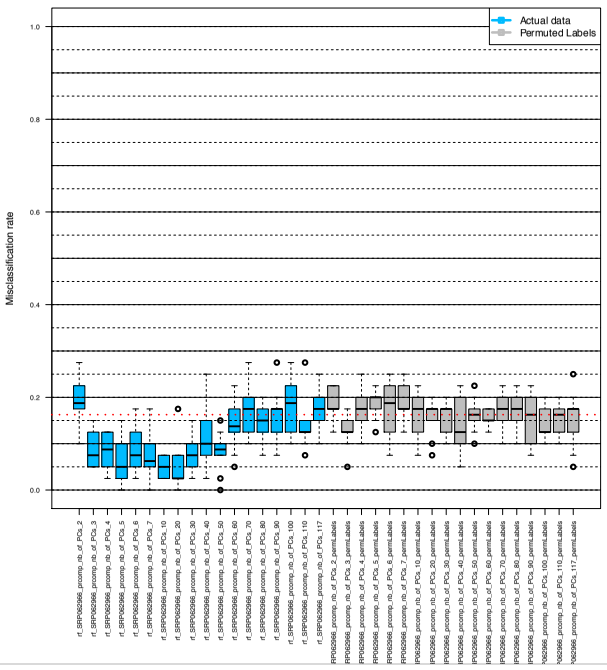
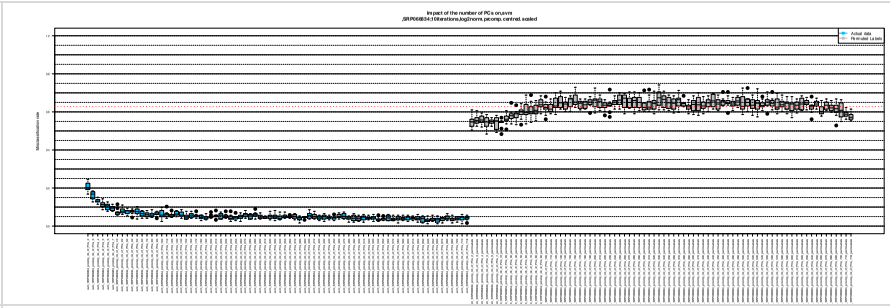
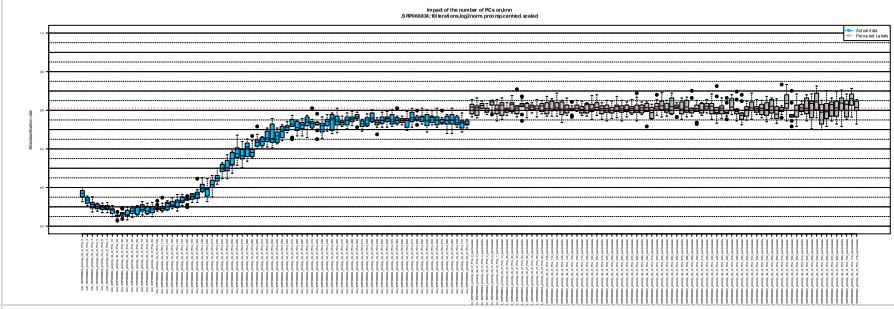


Figure 51. Impact of the selection of principal components on the performances of classifiers for the Cerebral organoids and fetal neocortex (SRP066834). Legend: see Figure 47.

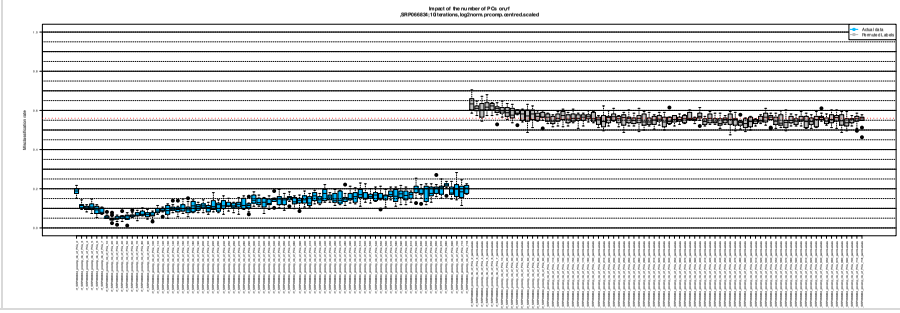
SVM



KNN



RF



A3.2. Feature selection based on the DESeq 2and edgeR

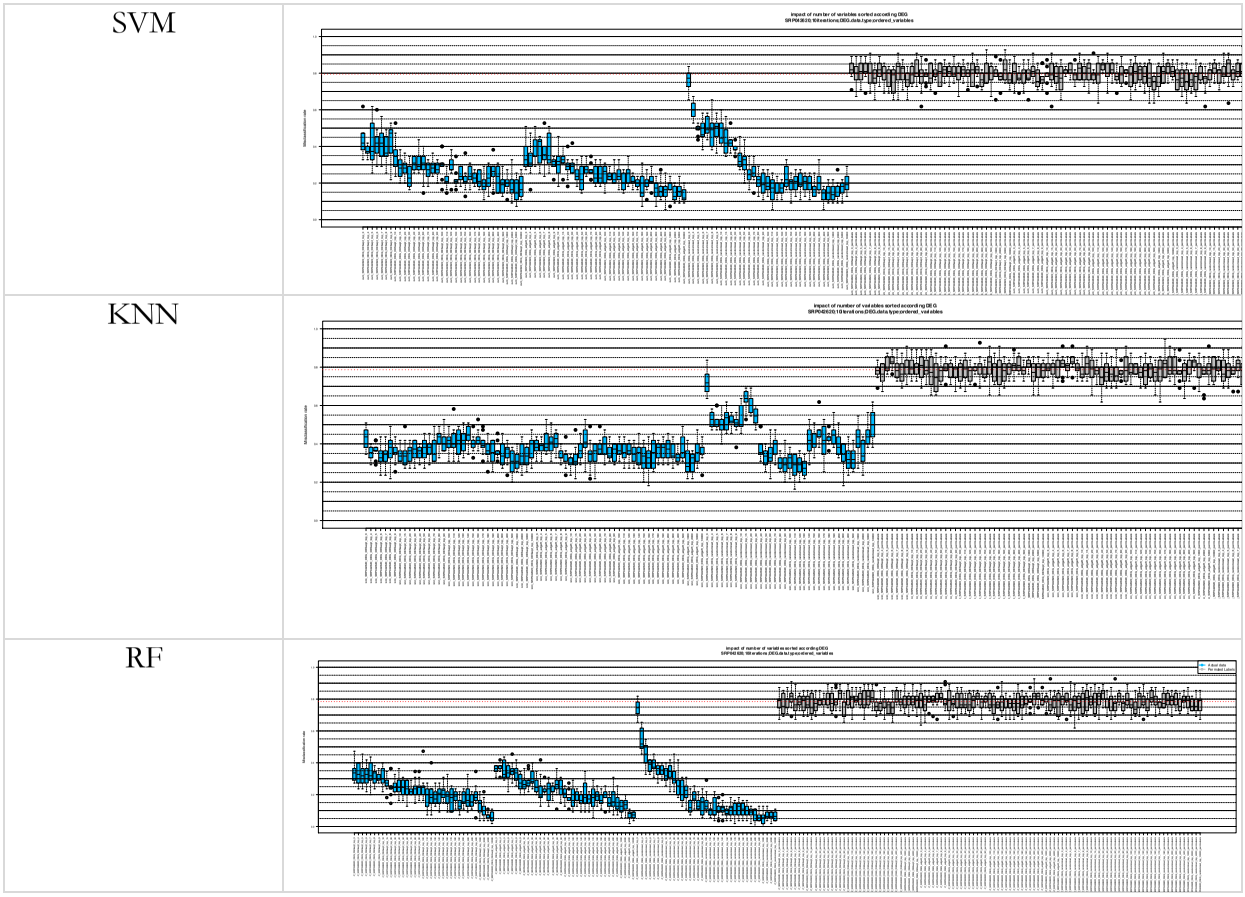


Figure 52 **Impact of feature selection on the misclassification error rate of classifiers for the *Breast cancer* study case (SRP042620).**
The ordinate indicates the misclassification error rates. Each boxplot corresponds to one testing-training experiment (10 iterations) and one particular feature selection method. **Blue boxplots:** From left to right, the 3 series of feature selection respectively correspond to DESeq2, edgeR or random ordering of the features. Within each series, the number of top-raking features progressively increases from left to right (numbers of top-ranking genes: 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 300, 400, 500, 1000, 2000, 5000, 10000). **Grey boxes:** random expectation estimated by permuting class labels during the training and testing. **Top:** Support Vector Machines (SVM); **Middle:** K-nearest neighbors (KNN); **Bottom:** Random Forest (RF).

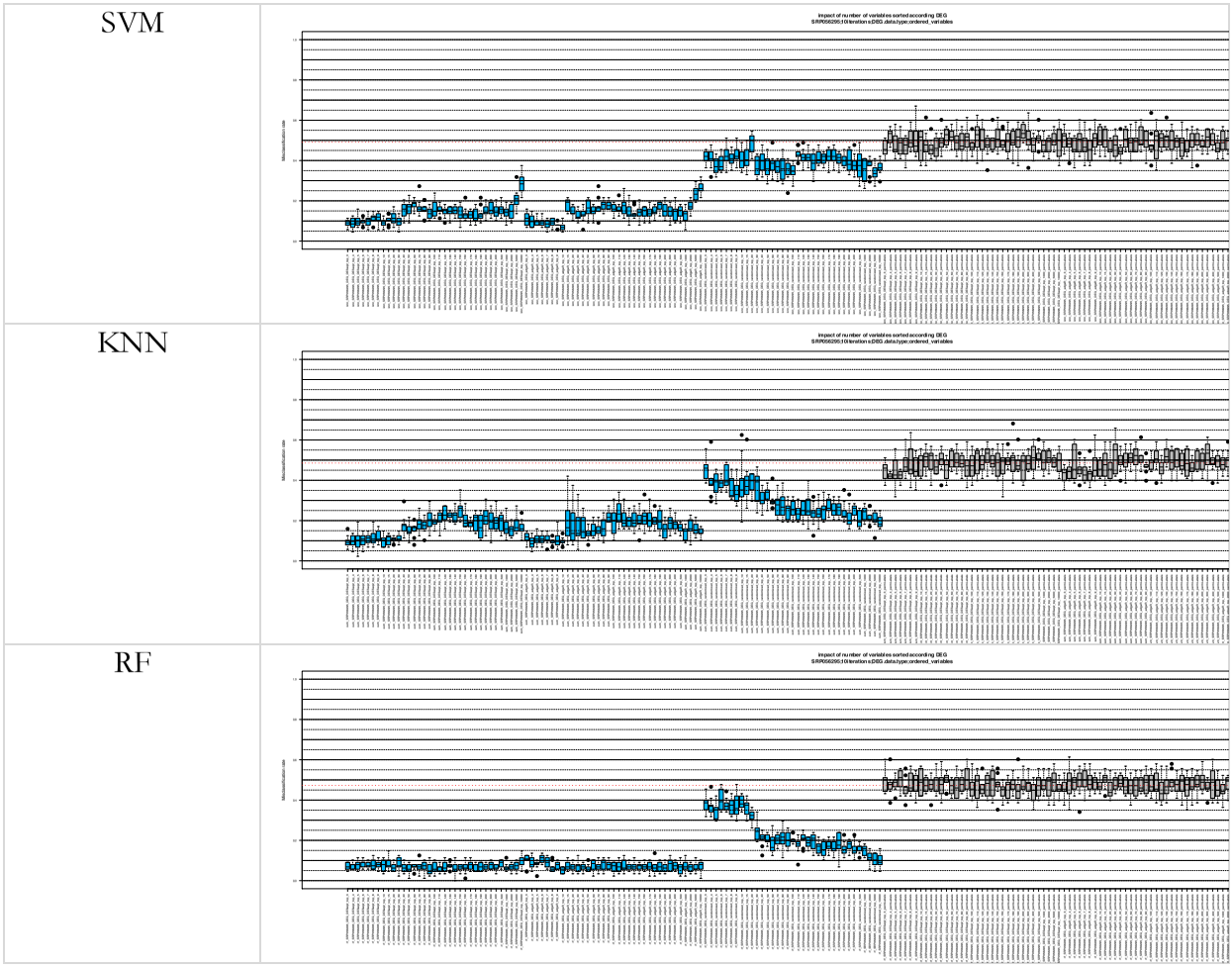


Figure 53 Impact of feature selection on the misclassification error rate of classifiers on the Human Leukemia (SRP056295). Legend: see Figure 52.

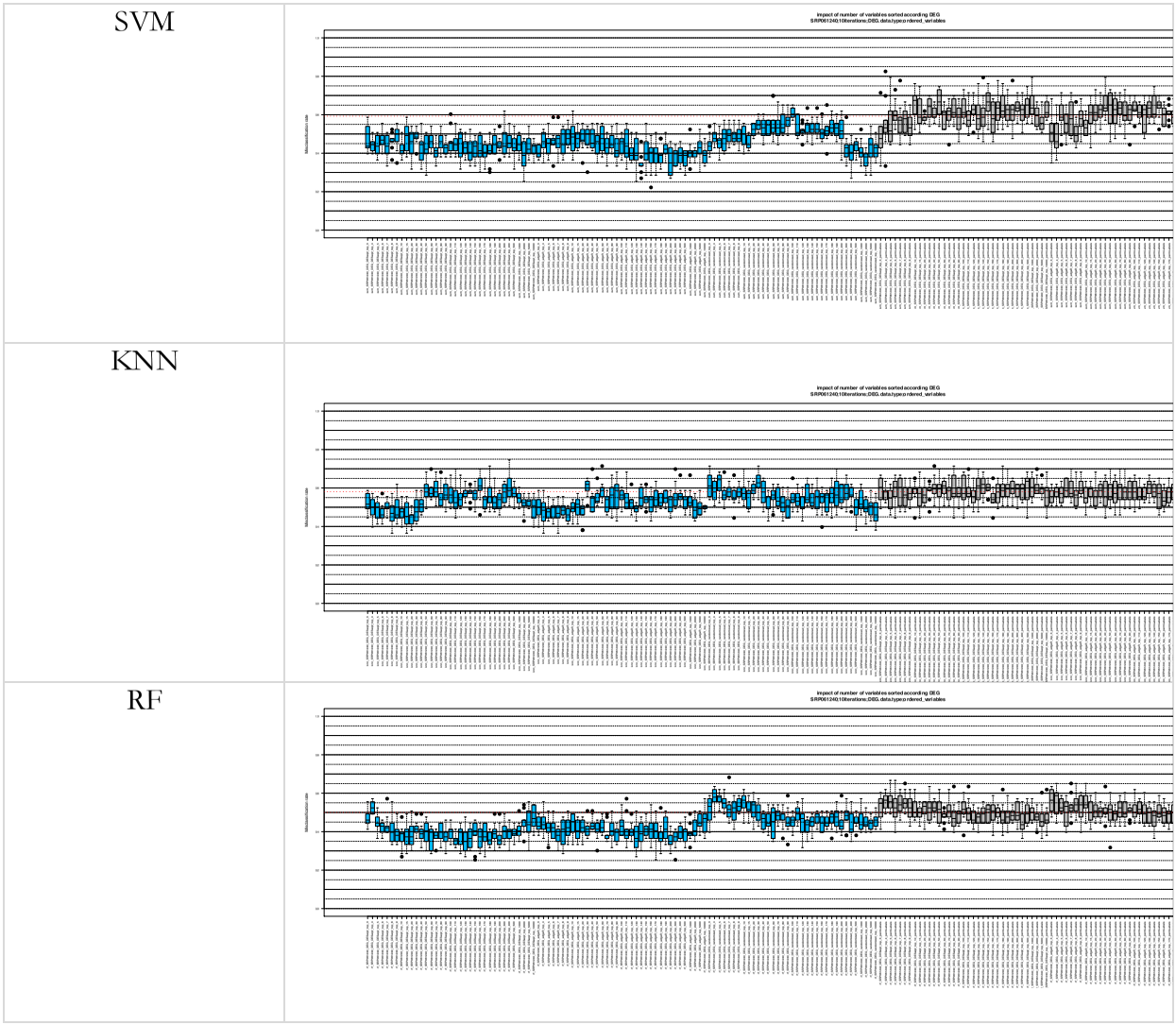


Figure 54 Impact of feature selection on the misclassification error rate of classifiers on the Cancer disease types (SRP061240). Legend: see Figure 52.

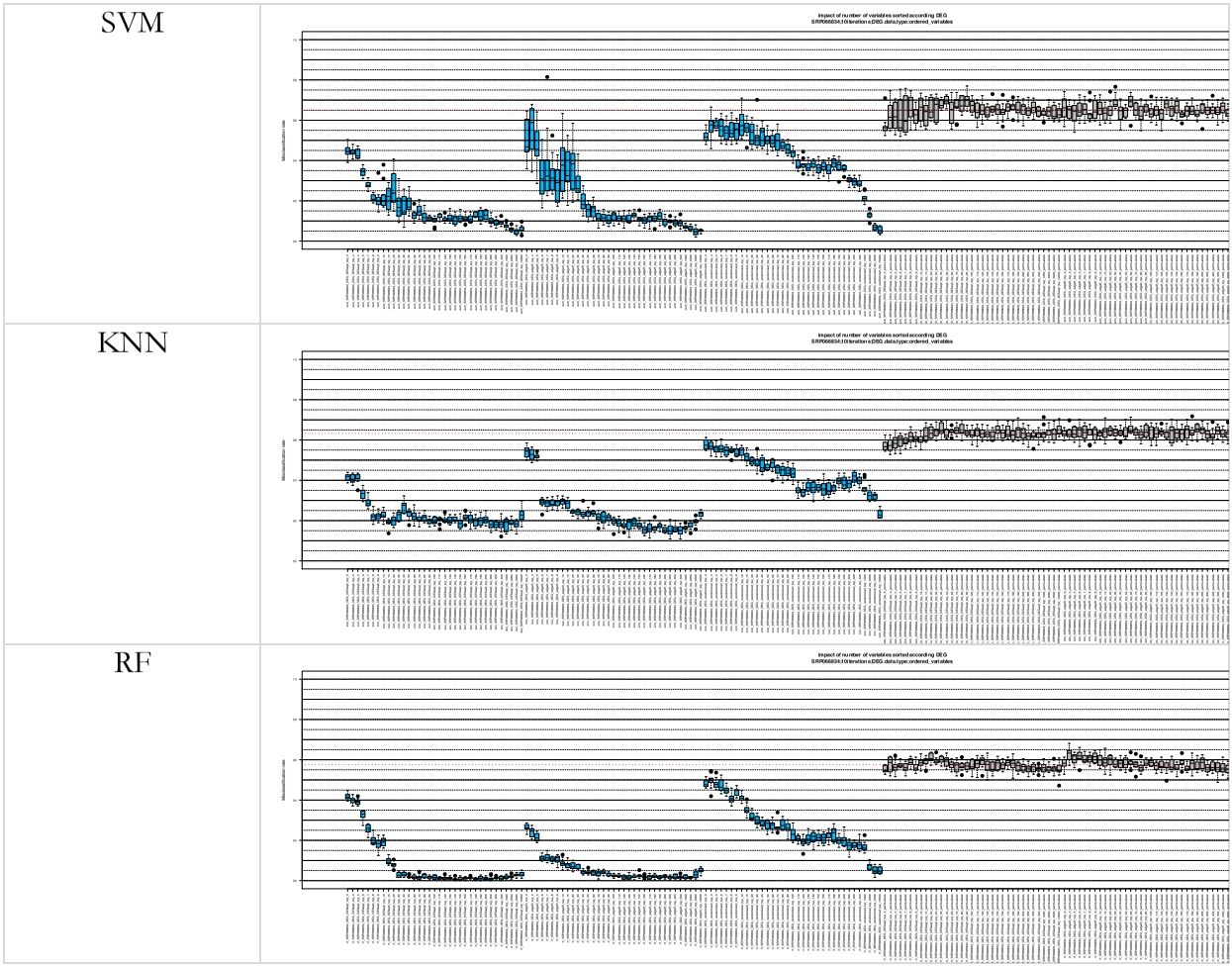


Figure 55 Impact of feature selection on the misclassification error rate of classifiers on the Cerebral organoids and fetal neocortex (SRP066834). Legend: see Figure 52.

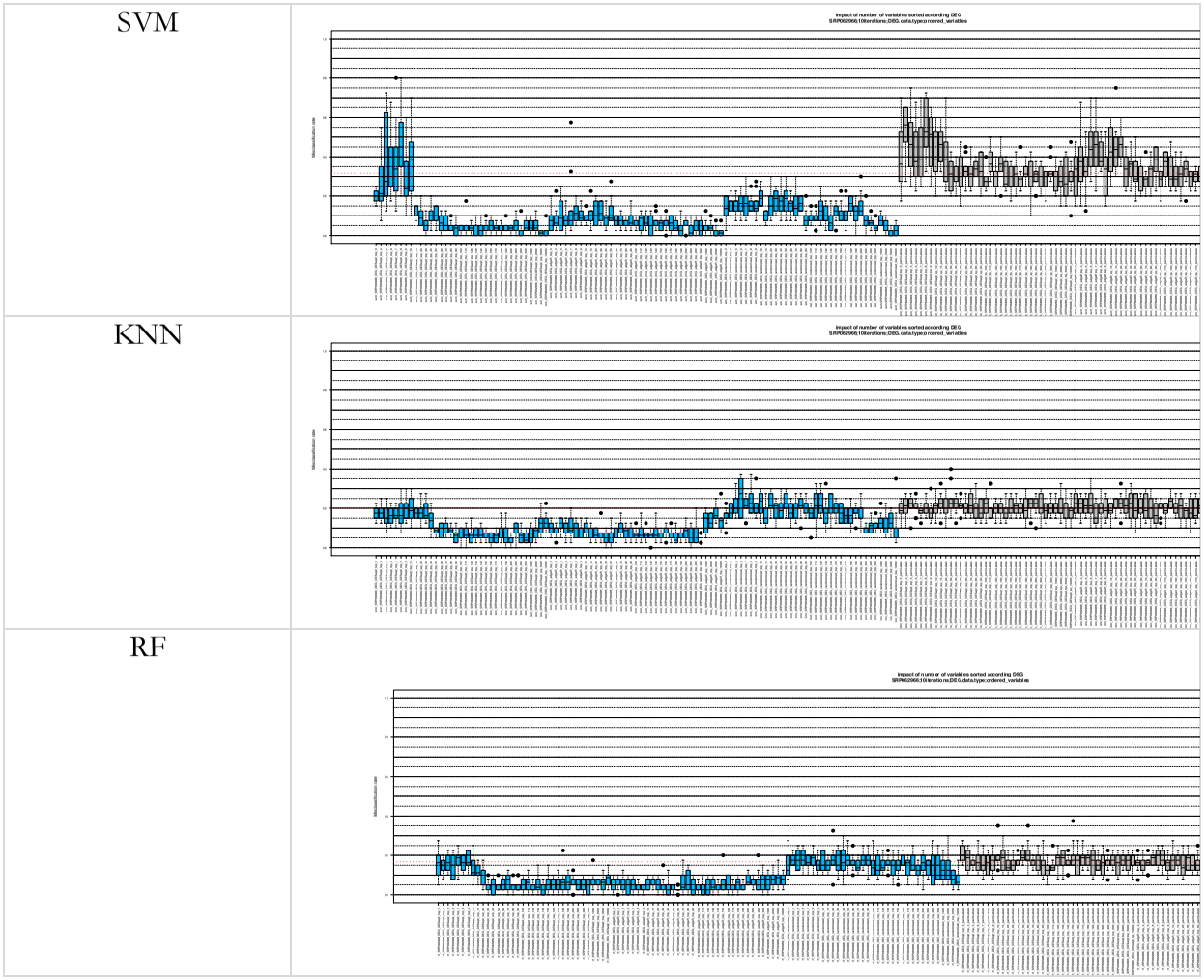


Figure 56 Impact of feature selection on the misclassification error rate of classifiers on the Bool Disease (SRP062966). Legend: see Figure 52.

A3.3. Feature selection based on the variables importance returned by a first pass of RF

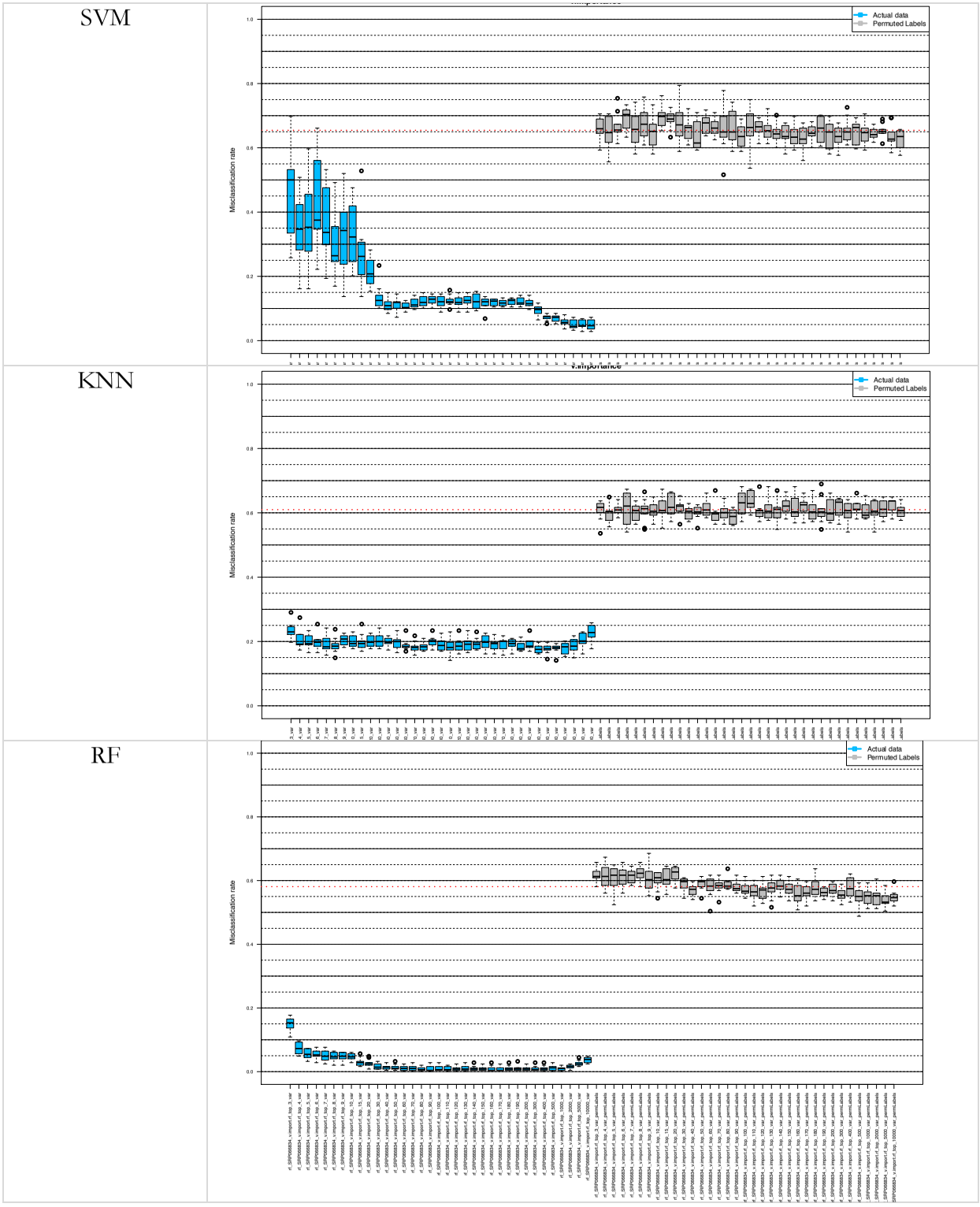


Figure 57. Impact of the features selection approach based on the misclassification error rate of classifiers on the Cerebral organoids and fetal neocortex (SRP066834) relied on VIMP which are generated from RF.

The ordinate indicates the misclassification error rates. Each boxplot corresponds to one testing-training experiment (10 iterations) and one particular feature selection method. Abbreviations: top-3-var-v.important: raw data ordered according to variable importance outputs from RF and number 3 indicate top significant feature based on the variable importance generated from RF; permLabels: random permutation of the sample labels, used to estimate the random expectation for the misclassification error rate. Blue boxes: result from the analysis of the actual datasets. Grey boxes: random expectation estimated by permuting class labels during the training and testing. Top: Support Vector Machines (SVM); Middle: K-nearest neighbors (KNN); Bottom: Random Forest (RF).

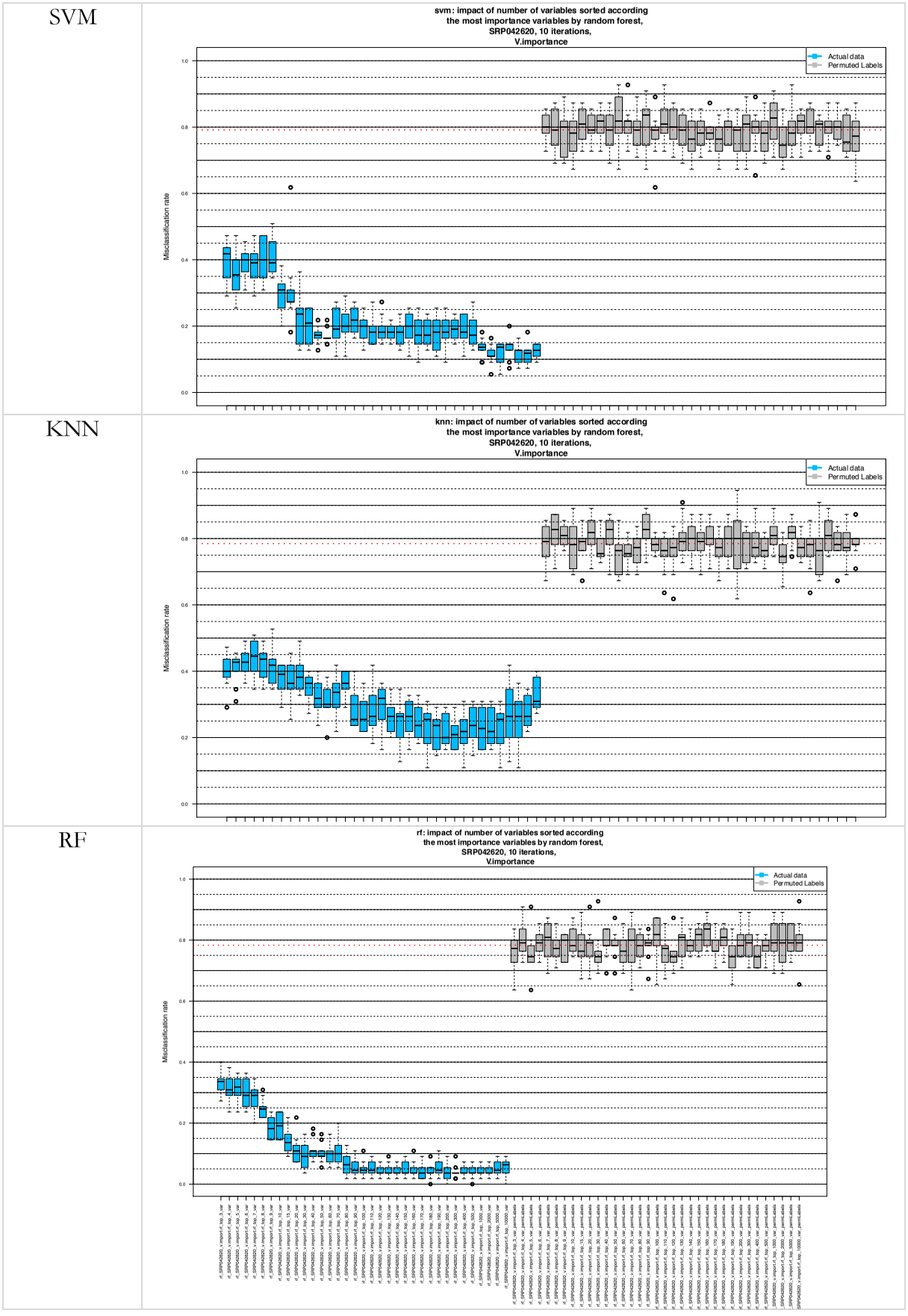


Figure 58 Impact of the features selection approach based on the misclassification error rate

of classifiers on the Breast Cancer (SRP042620) relied on VIMP which are generated from RF. Legend: see **Figure 57**.

A4. impact of K parameter into the KNN classifier

Figure 59 impact of K (nearest neighbor) of KNN into classifier accuracy for Cerebral organoids and fetal neocortex (SRP066834) measured by misclassification error rate. The ordinate indicate increased value of k parameter that are (3, 5, 7, 10, 15), abscissa indicates the misclassification error rates. Each boxplot corresponds to one testing-training experiment (10 iterations) and one particular pre-processing method is considered in this analyze is TMM_ log2 (. Blue boxes: result from the analysis of the actual datasets. Grey boxes: random expectation estimated by permuting class labels during the training and testing. Dotted line: theoretical value of the random expectation for the MER, based on class sizes.

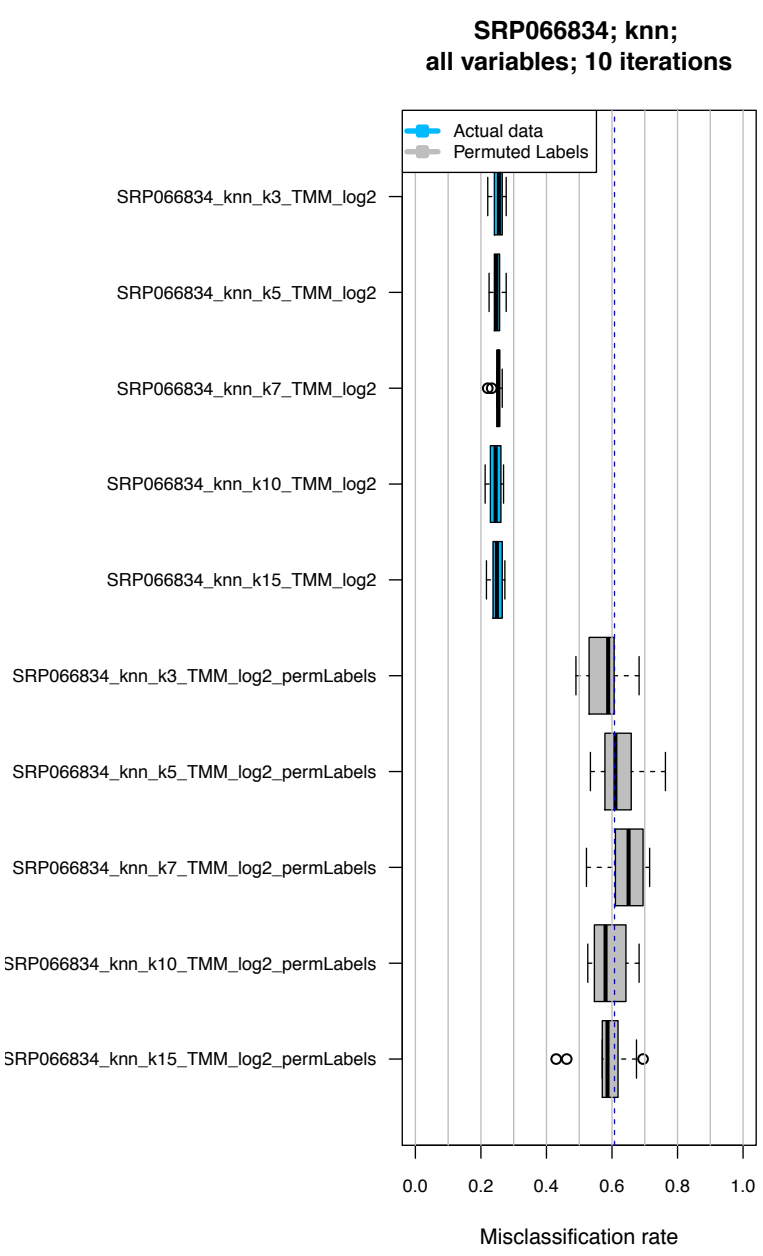


Figure 60 impact of K (nearest neighbor) of KNN into classifier accuracy for Psoriasis (SRP035988). Legend see Figure 59

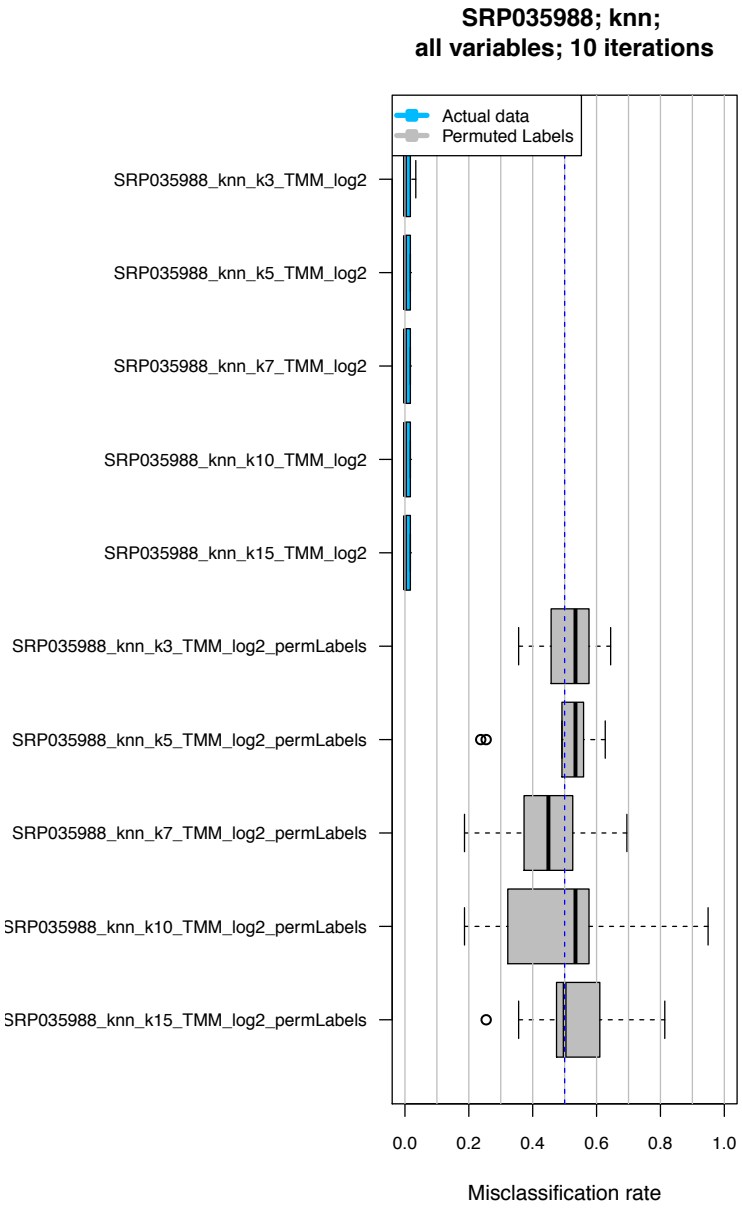
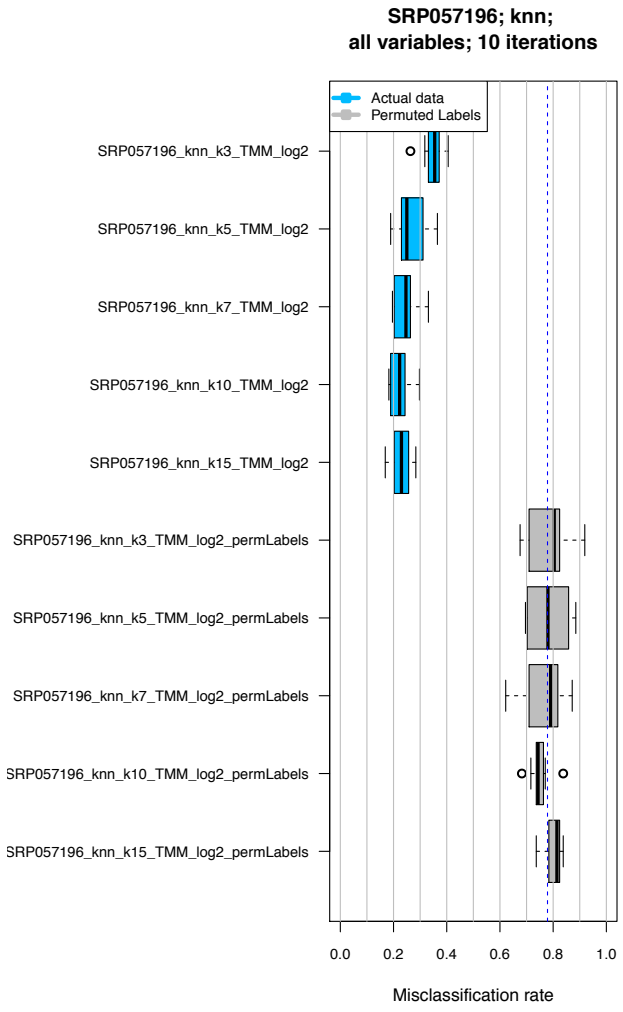


Figure61 impact of K (nearest neighbor) of KNN into classifier accuracy for Cellular complexity of the adult & fetal human brain (SRP057196). Legend see Figure 59



Appendix B: the RNAseqMVA package

Motivation for the RNAseqMVA package

RNAseqMVA is a comprehensive package for application of machine-learning algorithms in classification of next-generation RNA-Sequencing (RNA-seq) data. Researcher have invented RNAseqMVA for the various purposes, which include prediction of class labels of samples of cancer, disease, phenotype, tissues and etc. Preprocessing approaches include quartiles (Total count TC, Upper quartile UQ, median Med and third quartile Q3, trimmed mean of M TMM, Relative Log Expression RLE and Differential expression sequencing DEseq2) normalization methods can be used to correct systematics variations, as well as filtering methods at different levels which are genes, samples and classes to eliminate and mitigate the non-nature values NA, zeros and huge range in read counts. Principal component transformation can be used to bring discrete RNA-seq data hierarchically closer to microarrays to perform RNA-seq classification. Currently, RNAseqMVA package contains 3 RNA-seq-based classifiers. Besides these classifiers, RNAseqMVA package also include identification of best subset of features (genes) and sorting the features based on their differential expression analysis in one side and their variable importance generated from random forest. Researchers can build classification models, apply parameter optimization on these models, assessment and evaluate the model performance and compare the performances of different classification models. Moreover, the class labels of test samples can be predicted with the built models. RNAseqMVA is a user friendly, simple and surrently are published in GitHub repository as like the most comprehensive packages that are developed in the literature for RNA-seq classification. To start using this package users need to downloaded which RNA-seq dataset they are interesting based on the description of the experiment, by using such step the user will be able to download count tables which contains the number of count reads mapped to each transcript by pheno tables which contains whole description for each sample. This vignette is presented to guide researchers how to use this package.

UML Diagram of the RNAseqMVA package.

UML (Unified Modeling Language) was created as a result of the mess revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems and a results. In 1994-1996, the UML was developed by three software engineers working at rational software. It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only few updates.

Class diagram based on the UML with the purposes of visually representing our RNAseqMVA package along with its main objects, attributes, classes, subclasses, and properties beside that what is the relation between them that we have used it to give the researchers better understand, alter, maintain, or document information about the RNAseqMVA package.

We simply used UML to perform a modern approach to modeling and documenting our RNAseqMVA package. In fact, It's one of the most popular schema for visualizing object-oriented programming language. Notably, we used in our package S3 object-oriented in R programming language.

In **Figure 62**. We elucidate the make up a classes diagram, Briefly expose composition our analysis, it is requisite to have four classes which are studyCase, DataTableWithClasses, DataTableWithTrainTestSets and TrainTestResult. The sequel to, the structure of each class has 3 fields: the class name at the top, the class attributes right below the name, and the class operations/behaviors at the bottom. The relation between different classes (represented by a connecting line).

The summary of workflow of the analysis is initially you should have studycase object which is contain the recountID is the id is targeted in our analysis that mean the ID of the selected experiment from the recount2 warehouse, parameters that is vector which contains all required parameters in the analysis (e.g. directories, filtering, standardization, save.tables, save.image, classifiers, etc.), list of rawdata (countPerRuns, originalCounts) and dataSetForTest is list contains (filtered_count_table, Norm_count_table, log2_norm_count_table, log2_norm_PCs, log2_norm_DESeq_sorted, log2norm_edgeR_sorted, and log2_norm_Random_forst_sorted) this studyclase contains several operations and behaviors to name a few (load_count, Load_recount_experiment, filter_Data_Table, normalize_Samples, etc.). with classe of DataTableWithClasses it is inherited from the parent class Studycase which is have the same attribute for the studycase in addition to it has the special attributes related to class to name a few

(e.g. class_names, samples_per_class, class_Frequenceies, class_color, ect.). And DataTableWithClasses contains many operations / behaviors to name a few (build_attributes, iterate_Training_Testing, export_tables, etc.).

The third class is DataTableWithTrainTestsets. Which have their respective attribute is (Train_test_properties, iteration, training_properties. Train_size_per_class, etc.) with special operations / behaviors to name a few (build_Attributes, iterate_Training_Testing, print, etc.). the last class named TrainTestResults which have their respective attribute are (train_Index, test_Index, train_predicted_classes, test_predicted_classes, etc.).

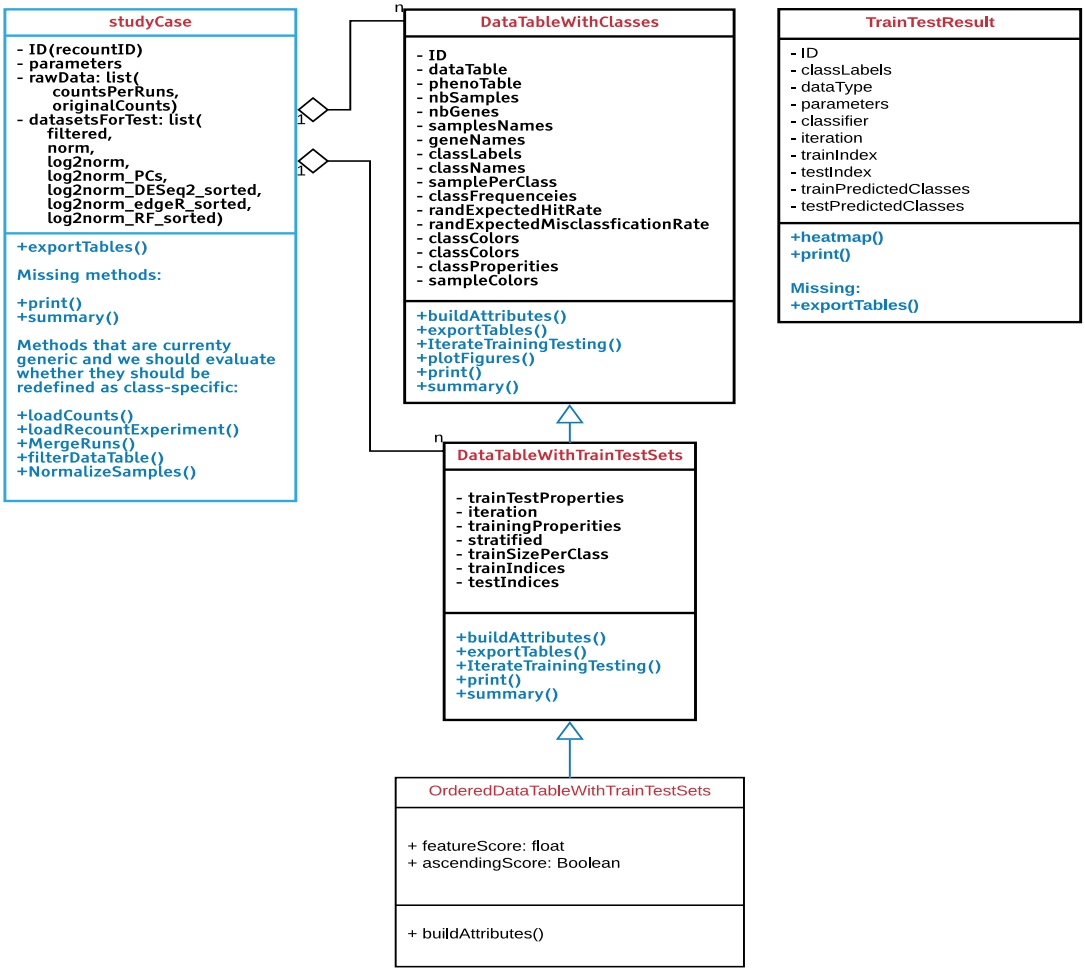


Figure 62 UML diagram to visuals our used objects and their classes and operations.

Availability of RNAseqMVA

The RNAseqMVA is available on github: https://github.com/elqumsan/RNAseqMVA Currently, the package can readily be downloaded, compiled and used to reproduce all the results presented in this PhD thesis. We are in the process of writing a Vignette which will provide detailed information about the use of the package, and enable its usage for analyses of custom datasets.

Appendix C: full list of R packages used

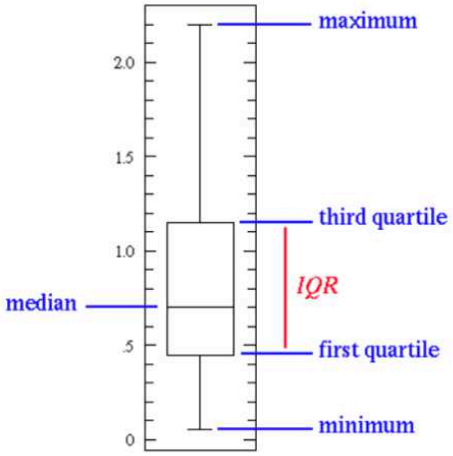
```
─ Session info ─────────────────────────────────────────────────────────────────
 setting  value
 version  R version 3.5.1 (2018-07-02)
 os       macOS  10.14.1
 system   x86_64, darwin15.6.0
 ui       RStudio
 language en_US.UTF-8
 collate  en_US.UTF-8
 ctype    en_US.UTF-8
 tz       Europe/Paris
 date     2018-12-03

─ Packages ─────────────────────────────────────────────────────────────────────
 package             * version    date       lib source
 abind                 1.4-5      2016-07-21 [1] CRAN (R 3.5.0)
 acepack               1.4.1      2016-10-29 [1] CRAN (R 3.5.0)
 affy                  1.58.0     2018-05-01 [1] Bioconductor
 affyio                1.50.0     2018-05-01 [1] Bioconductor
 annotate              1.58.0     2018-05-01 [1] Bioconductor
 AnnotationDbi         1.42.1     2018-05-08 [1] Bioconductor
 assertthat            0.2.0      2017-04-11 [1] CRAN (R 3.5.0)
 backports             1.1.2      2017-12-13 [1] CRAN (R 3.5.0)
 base64enc             0.1-3      2015-07-28 [1] CRAN (R 3.5.0)
 bibtex                0.4.2      2017-06-30 [1] CRAN (R 3.5.0)
 bindr                 0.1.1      2018-03-13 [1] CRAN (R 3.5.0)
 bindrcpp              0.2.2      2018-03-29 [1] CRAN (R 3.5.0)
 Biobase             * 2.40.0     2018-05-01 [1] Bioconductor
 BiocGenerics        * 0.26.0     2018-05-01 [1] Bioconductor
 BiocInstaller         1.30.0     2018-05-01 [1] Bioconductor
 BiocParallel        * 1.14.2     2018-07-08 [1] Bioconductor
 biomaRt               2.36.1     2018-05-24 [1] Bioconductor
 Biostrings            2.48.0     2018-05-01 [1] Bioconductor
 bit                   1.1-14     2018-05-29 [1] CRAN (R 3.5.0)
 bit64                 0.9-7      2017-05-08 [1] CRAN (R 3.5.0)
 bitops              * 1.0-6      2013-08-17 [1] CRAN (R 3.5.0)
 blob                  1.1.1      2018-03-25 [1] CRAN (R 3.5.0)
 broom               * 0.5.0      2018-07-17 [1] CRAN (R 3.5.0)
 BSgenome              1.48.0     2018-05-01 [1] Bioconductor
 bumphunter            1.22.0     2018-05-01 [1] Bioconductor
 callr                 3.0.0      2018-08-24 [1] CRAN (R 3.5.0)
 caret               * 6.0-80     2018-05-26 [1] CRAN (R 3.5.0)
 checkmate             1.8.5      2017-10-24 [1] CRAN (R 3.5.0)
 class               * 7.3-14     2015-08-30 [2] CRAN (R 3.5.1)
 cli                   1.0.1      2018-09-25 [1] CRAN (R 3.5.0)
 cluster               2.0.7-1    2018-04-13 [2] CRAN (R 3.5.1)
 codetools             0.2-15     2016-10-05 [2] CRAN (R 3.5.1)
 colorspace            1.3-2      2016-12-14 [1] CRAN (R 3.5.0)
 commonmark            1.6        2018-09-30 [1] CRAN (R 3.5.0)
 crayon                1.3.4      2017-09-16 [1] CRAN (R 3.5.0)
 CVST                  0.2-2      2018-05-26 [1] CRAN (R 3.5.0)
 data.table            1.11.8     2018-09-30 [1] CRAN (R 3.5.0)
 DBI                   1.0.0      2018-05-02 [1] CRAN (R 3.5.0)
 ddalpha               1.3.4      2018-06-23 [1] CRAN (R 3.5.0)
 DelayedArray        * 0.6.6      2018-09-11 [1] Bioconductor
 DEoptimR              1.0-8      2016-11-19 [1] CRAN (R 3.5.0)
 derfinder           * 1.14.0     2018-05-01 [1] Bioconductor
 derfinderHelper       1.14.0     2018-05-01 [1] Bioconductor
 desc                  1.2.0      2018-05-01 [1] CRAN (R 3.5.0)
 DESeq2              * 1.20.0     2018-05-01 [1] Bioconductor
 devtools            * 2.0.1      2018-10-26 [1] CRAN (R 3.5.1)
```

```
digest                0.6.18     2018-10-10 [1] CRAN (R 3.5.0)
dimRed                0.2.2      2018-11-13 [1] CRAN (R 3.5.1)
doMC                * 1.3.5      2017-12-12 [1] CRAN (R 3.5.0)
doParallel          * 1.0.14     2018-09-24 [1] CRAN (R 3.5.0)
doRNG                 1.7.1      2018-06-22 [1] CRAN (R 3.5.0)
downloader            0.4        2015-07-09 [1] CRAN (R 3.5.0)
dplyr               * 0.7.8      2018-11-10 [1] CRAN (R 3.5.0)
DRR                   0.0.3      2018-01-06 [1] CRAN (R 3.5.0)
e1071               * 1.7-0      2018-07-28 [1] CRAN (R 3.5.0)
edgeR               * 3.22.5     2018-10-02 [1] Bioconductor
foreach             * 1.4.4      2017-12-12 [1] CRAN (R 3.5.0)
foreign               0.8-71     2018-07-20 [2] CRAN (R 3.5.0)
Formula               1.2-3      2018-05-03 [1] CRAN (R 3.5.0)
fs                    1.2.6      2018-08-23 [1] CRAN (R 3.5.0)
genefilter            1.62.0     2018-05-01 [1] Bioconductor
geneplotter           1.58.0     2018-05-01 [1] Bioconductor
GenomeInfoDb        * 1.16.0     2018-05-01 [1] Bioconductor
GenomeInfoDbData      1.1.0      2018-10-29 [1] Bioconductor
GenomicAlignments     1.16.0     2018-05-01 [1] Bioconductor
GenomicFeatures       1.32.3     2018-10-04 [1] Bioconductor
GenomicFiles          1.16.0     2018-05-01 [1] Bioconductor
GenomicRanges       * 1.32.7     2018-09-20 [1] Bioconductor
geometry              0.3-6      2015-09-09 [1] CRAN (R 3.5.0)
GEOquery              2.48.0     2018-05-01 [1] Bioconductor
ggplot2             * 3.1.0      2018-10-25 [1] CRAN (R 3.5.0)
glue                  1.3.0      2018-07-17 [1] CRAN (R 3.5.0)
gower                 0.1.2      2017-02-23 [1] CRAN (R 3.5.0)
gridExtra             2.3        2017-09-09 [1] CRAN (R 3.5.0)
gtable                0.2.0      2016-02-26 [1] CRAN (R 3.5.0)
Hmisc                 4.1-1      2018-01-03 [1] CRAN (R 3.5.0)
hms                   0.4.2      2018-03-10 [1] CRAN (R 3.5.0)
htmlTable             1.12       2018-05-26 [1] CRAN (R 3.5.0)
htmltools             0.3.6      2017-04-28 [1] CRAN (R 3.5.0)
htmlwidgets           1.3        2018-09-30 [1] CRAN (R 3.5.0)
httr                  1.3.1      2017-08-20 [1] CRAN (R 3.5.0)
ipred                 0.9-8      2018-11-05 [1] CRAN (R 3.5.0)
IRanges             * 2.14.12    2018-09-20 [1] Bioconductor
iterators           * 1.0.10     2018-07-13 [1] CRAN (R 3.5.0)
jsonlite              1.5        2017-06-01 [1] CRAN (R 3.5.0)
kernlab               0.9-27     2018-08-10 [1] CRAN (R 3.5.0)
knitr               * 1.20       2018-02-20 [1] CRAN (R 3.5.0)
lattice             * 0.20-38    2018-11-04 [2] CRAN (R 3.5.0)
latticeExtra          0.6-28     2016-02-09 [1] CRAN (R 3.5.0)
lava                  1.6.3      2018-08-10 [1] CRAN (R 3.5.0)
lazyeval              0.2.1      2017-10-29 [1] CRAN (R 3.5.0)
limma               * 3.36.5     2018-09-20 [1] Bioconductor
locfit                1.5-9.1    2013-04-20 [1] CRAN (R 3.5.0)
lubridate             1.7.4      2018-04-11 [1] CRAN (R 3.5.0)
magic                 1.5-9      2018-09-17 [1] CRAN (R 3.5.0)
magrittr              1.5        2014-11-22 [1] CRAN (R 3.5.0)
MASS                  7.3-51.1   2018-11-01 [2] CRAN (R 3.5.0)
Matrix                1.2-15     2018-11-01 [2] CRAN (R 3.5.0)
matrixStats         * 0.54.0     2018-07-23 [1] CRAN (R 3.5.0)
memoise               1.1.0      2017-04-21 [1] CRAN (R 3.5.0)
ModelMetrics          1.2.2      2018-11-03 [1] CRAN (R 3.5.0)
munsell               0.5.0      2018-06-12 [1] CRAN (R 3.5.0)
nlme                  3.1-137    2018-04-07 [2] CRAN (R 3.5.1)
nnet                  7.3-12     2016-02-02 [2] CRAN (R 3.5.1)
pheatmap            * 1.0.10     2018-05-19 [1] CRAN (R 3.5.0)
pillar                1.3.0      2018-07-14 [1] CRAN (R 3.5.0)
pkgbuild              1.0.2      2018-10-16 [1] CRAN (R 3.5.0)
pkgconfig             2.0.2      2018-08-16 [1] CRAN (R 3.5.0)
pkgload               1.0.2      2018-10-29 [1] CRAN (R 3.5.1)
```

```
pkgmaker               0.27       2018-05-25 [1] CRAN (R 3.5.0)
pls                    2.7-0      2018-08-21 [1] CRAN (R 3.5.0)
plyr                   1.8.4      2016-06-08 [1] CRAN (R 3.5.0)
preprocessCore         1.42.0     2018-05-01 [1] Bioconductor
prettyunits            1.0.2      2015-07-13 [1] CRAN (R 3.5.0)
processx               3.2.0      2018-08-16 [1] CRAN (R 3.5.0)
prodlim                2018.04.18 2018-04-18 [1] CRAN (R 3.5.0)
progress               1.2.0      2018-06-14 [1] CRAN (R 3.5.0)
ps                     1.2.1      2018-11-06 [1] CRAN (R 3.5.0)
purrr                  0.2.5      2018-05-29 [1] CRAN (R 3.5.0)
qvalue                 2.12.0     2018-05-01 [1] Bioconductor
R6                     2.3.0      2018-10-04 [1] CRAN (R 3.5.0)
randomForest         * 4.6-14     2018-03-25 [1] CRAN (R 3.5.0)
RColorBrewer           1.1-2      2014-12-07 [1] CRAN (R 3.5.0)
Rcpp                 * 1.0.0      2018-11-07 [1] CRAN (R 3.5.0)
RcppRoll               0.3.0      2018-06-05 [1] CRAN (R 3.5.0)
RCurl                * 1.95-4.11  2018-07-15 [1] CRAN (R 3.5.0)
readr                  1.1.1      2017-05-16 [1] CRAN (R 3.5.0)
recipes                0.1.3      2018-06-16 [1] CRAN (R 3.5.0)
recount              * 1.6.3      2018-07-28 [1] Bioconductor
registry               0.5        2017-12-03 [1] CRAN (R 3.5.0)
remotes                2.0.2      2018-10-30 [1] CRAN (R 3.5.1)
rentrez                1.2.1      2018-03-05 [1] CRAN (R 3.5.0)
reshape2               1.4.3      2017-12-11 [1] CRAN (R 3.5.0)
rlang                  0.3.0.1    2018-10-25 [1] CRAN (R 3.5.0)
RNAseqMVA            * 0.5.1      2018-12-03 [1] local
rngtools               1.3.1      2018-05-15 [1] CRAN (R 3.5.0)
robustbase             0.93-3     2018-09-21 [1] CRAN (R 3.5.0)
roxygen2             * 6.1.1      2018-11-07 [1] CRAN (R 3.5.0)
rpart                  4.1-13     2018-02-23 [2] CRAN (R 3.5.1)
rprojroot              1.3-2      2018-01-03 [1] CRAN (R 3.5.0)
Rsamtools              1.32.3     2018-08-22 [1] Bioconductor
RSQLite                2.1.1      2018-05-06 [1] CRAN (R 3.5.0)
rstudioapi             0.8        2018-10-02 [1] CRAN (R 3.5.0)
rtracklayer            1.40.6     2018-09-04 [1] Bioconductor
S4Vectors            * 0.18.3     2018-06-08 [1] Bioconductor
scales                 1.0.0      2018-08-09 [1] CRAN (R 3.5.0)
scatterplot3d        * 0.3-41     2018-03-14 [1] CRAN (R 3.5.0)
sessioninfo            1.1.1      2018-11-05 [1] CRAN (R 3.5.0)
sfsmisc                1.1-2      2018-03-05 [1] CRAN (R 3.5.0)
stringi                1.2.4      2018-07-20 [1] CRAN (R 3.5.0)
stringr                1.3.1      2018-05-10 [1] CRAN (R 3.5.0)
SummarizedExperiment * 1.10.1     2018-05-11 [1] Bioconductor
survival               2.43-1     2018-10-29 [2] CRAN (R 3.5.1)
testthat               2.0.1      2018-10-13 [1] CRAN (R 3.5.0)
tibble                 1.4.2      2018-01-22 [1] CRAN (R 3.5.0)
tidyr                  0.8.2      2018-10-28 [1] CRAN (R 3.5.0)
tidyselect             0.2.5      2018-10-11 [1] CRAN (R 3.5.0)
timeDate               3043.102   2018-02-21 [1] CRAN (R 3.5.0)
usethis              * 1.4.0      2018-08-14 [1] CRAN (R 3.5.0)
VariantAnnotation      1.26.1     2018-07-04 [1] Bioconductor
vsn                  * 3.48.1     2018-05-24 [1] Bioconductor
withr                  2.1.2      2018-03-15 [1] CRAN (R 3.5.0)
XML                  * 3.98-1.16  2018-08-19 [1] CRAN (R 3.5.0)
xml2                   1.2.0      2018-01-24 [1] CRAN (R 3.5.0)
xtable                 1.8-3      2018-08-29 [1] CRAN (R 3.5.0)
XVector                0.20.0     2018-05-01 [1] Bioconductor
yaml                 * 2.2.0      2018-07-25 [1] CRAN (R 3.5.0)
zlibbioc               1.26.0     2018-05-01 [1] Bioconductor
```

Appendix D: Glossary

Alpha (α)	An α- level test statistic would reject H_0 if $p \leq \alpha$.
An alternative hypothesis (H1)	Is a contrasting assertion about the population that can be tested against the null hypothesis.
ANOVA	Analysis of variance usually refers to an analysis of a contiuous dependent variable where all the predictor variables are categorical. One-way ANOVA, where there is only one predictor variable, is a generalization of the 2-sample t-test. ANOVA with 2 group is identical to the t-test. Two-way ANOVA refers to two predictors, and if the two are allowed to interact in the model, two-way ANOVA involves cross-classification of observations simultaneously by both factors.
Bootstrapping	Is the process of dividing the dataset into multiple subsets, with replacement. Each subset is of the same size of the dataset. These samples are called bootstrap samples.
Box Plot	It displays the full range of variation (from min to max), the likely range of variation (the interquartile range), and a typical value (the median). Below is a visualization of a box plot



Some of the inferences that can be made from a box plot:

- **Median**: middle quartile marks the median.
- **Middle box** represents the 50% of the data.
- **First quartile:** 25% of data falls below these line.
- **Third quartile:** 75% of data falls below these line.

Cross-validation	This technique involves leaving out m individuals at time, fitting a model on the remaining $n - m$ individuals, and obtaining an unbiased evaluation of predictive accuracy on the m individuals. The estimates are averaged over $\geq n/m$ repetition.
E-value	Expected value is statistics the sum or integral of all possible values of a random variable, or any given function of it, multiplied by the respective probabilities of the values of the variable.
Estimate	A statistical estimate of a parameter based on the data. See parameter, Examples include the sample mean, sample median, etc.
FDR ~ q-value	Control the proportion of false positive among rejected hypothesis.
FWER	Control the probability to have at least one false positive among rejected hypothesis.
Hypothesis testing	Is a common method of drawing inferences about a population based on statistical evidence from a sample.
Jack-knife	A statistical method of numerical resampling based on n samples of size $n - 1$ used to calculate the variance of an estimate from an original of size n
Multivariate model	A model that simultaneously predicts more than one dependent variable.
Null hypothesis (H0)	Customarily but not necessarily a hypothesis of no effect, the null hypothesis labeled H_0, is often used in the frequentist branch of statistical inference; classical statistics often assumes what one hopes doesn't happen (no effect of a treatment) and attempts to gather evidence against that assumption (i.e., tries to reject H_0).
P-value	The probability of getting a result of a test statistic as or more extreme than observed statistic had H_0 been true.
Quartiles	The 25^{th} and 75^{th} percentiles and the median. The three values divide a variables distributions into four intervals containing equal numbers of individuals.
False Negative Risk (FNR)	It is probability to getting sum of the false negative among the accepted hypothesis.
False Positive Risk (FPR)	It is the probability to getting sum of the false positive among the rejected hypothesis

Type I error rate	It is False positive rate: the probability of rejecting H_0 when the null hypothesis is in fact true. The type I error is often called α.
Type II error rate	Failing to detect an effect that is real. The type II error is referred to as β, which is one minus the power of the test. That mean the power of test is $1-\beta$

LIST OF FIGURES

LIST OF TABLES