Universidade de Lisboa
Faculdade de Ciências
Departamento de Estatística e Investigação Operacional

LISBOA

UNIVERSIDADE
DE LISBOA

Study of the Role of SETD2 Mutations in clear cell Renal Cell Carcinoma (ccRCC)

Catarina Faria de Almeida

Trabalho de Projecto
Mestrado em Bioestatística

2013

Universidade de Lisboa
Faculdade de Ciências
Departamento de Estatística e Investigação Operacional

Study of the Role of SETD2 Mutations in clear cell Renal Cell Carcinoma (ccRCC)

Catarina Faria de Almeida

Trabalho de Projecto

Mestrado em Bioestatística

Trabalho de projecto orientado por Professora Doutora Lisete Maria Ribeiro de Sousa e por Professora Doutora Ana Rita Fialho Grosso

2013

To every uncle Tim.

Acknowledgements

To my great tutors, Professor Lisete Sousa and Professor Ana Rita Grosso, I have to express sincere gratitude, if there are people that can complain about their tutors, I was definitely not one of them, always teaching, always helpful, always patient, always supportive, always kind, even when working with me was not easy.

A special thank you to João, who was a source of stability that he can not possibly imagine, thanks for all *you've* done.

To Ana who is always ready to step in, always somebody to lean on. To my sister Mariana and to Sofia, always so close, no matter how far.

Last and most dear to my heart, I wish to thank my parents, who my nerves and anxiety tried to push away, and they never allowed for that to happen, *you've* been so good to me. I subscribe the sincere cliché that I owe them everything.

To all of these and to everyone who stood by my side and helped me get to where I am today, a sincere thank you.

Catarina Almeida, September 2013

Abstract

Clear cell Renal Cell Carcinoma, ccRCC, is the most common form of Renal Cancer, accounting for 90% of these cancers cases. It is well established that the majority of these cancers happen when both alleles of VHL (Von Hippel Lindau) tumour suppressor gene are mutated. It has also been observed that patients with this form of cancer present mutations on the SETD2 gene which applies its functions during transcription.

In the last few years the growth within sequencing technologies has been astonishing. Next Generation Sequencing technology, NGS, provides tools for assessing full genomes to a reference sequence in a matter of days, being extremely accurate, while also increasingly cost effective. One of its many applications is RNA-Sequencing, a method for transcriptome analysis.

Throughout this thesis we aimed at analysing RNA-Seq data from six samples: four with mutations on the SETD2 gene and two control samples. The main goal was to understand how the remaining genes on the transcriptome respond to these genes mutations.

In the first part of this work we aimed at analysing several forms to normalize the data, resorting to `R` software packages (`EDASeq`, `DESeq` and `edgeR`). Data normalization is a crucial step on NGS techniques, as these techniques have some inherent bias that need to be accounted for. `DESeq` proved to be the most selective, while `EDASeq` is not as stringent. The second part of this work aimed at identifying differentially expressed genes, to infer which genes behave in a significant way in the samples, packages `edgeR`, `DESeq` and `RankProd`. We identified six new genes as differentially expressed.

Keywords: Next Generation Sequencing (NGS), clear cell Renal Cell Carcinoma, normalization, differential expression, SETD2

Resumo

O carcinoma renal de células claras, ccRCC, é o tipo mais comum de cancro renal, sendo responsável por cerca de 70% destes tumores, com a mais alta taxa de mortalidade entre todos os tipos de cancro renal. A grande maioria destes tumores deve-se a mutações no gene VHL, um gene supressor de cancro. Não obstante, vários estudos de sequenciação do ccRCC acabaram por revelar a ocorrência de mutações somáticas no gene SETD2, uma *Histone methyltransferase* que trimetila a lisina 36 na histona H3 (H3K36me3). Este gene tem um papel fundamental na transcrição, um dos principais passos na expressão genética – processo pelo qual são geradas proteínas perfeitamente funcionais, permitindo que o ADN se desenrole e seja posteriormente transcrito. Está localizado no braço curto do cromossoma 3 e as mutações deste gene conduzem à perda de funções deste mesmo cromossoma.

O objectivo biológico da presente tese é avaliar as alterações do transcriptoma, induzidas pelas mutações no gene SETD2. Esta questão será abordada utilizando dados do transcriptoma completo, de linhas celulares mutadas no gene SETD2 e de linhas celulares não mutadas do gene *wild tipe* (WT), no ccRCC. Os dados desta análise consideraram 4 amostras biológicas de transcriptomas mutados do genoma SETD2 e 2 amostras *wild tipe*, dados estes que foram gerados pela unidade do investigador Sérgio Almeida, do Instituto de Medicina Molecular (IMM da FMUL).

Recentemente, o desenvolvimento de novas tecnologias de métodos de sequenciação, designadas por *Next Generation Sequencing (NGS)*, disponibilizou um novo método que, em simultâneo, executa o mapeamento e a quantificação de transcriptomas, chamado sequenciação de RNA (RNA-seq). Apesar de mais expendioso do que os estudos de *microarrays* e ainda com alguns problemas de análise de dados por resolver, a sequenciação do RNA pode avaliar o transcriptoma completo, disponibilizando a derradeira solução para a análise dos níveis e da estrutura de transcriptomas processados e não processados, sob diferentes condições. Esta técnica disponibiliza uma importante poupança de tempo (o genoma humano completo pode ser sequenciado em menos de uma semana, dependendo das opções do investigador) com qualidade, precisão de leitura (cerca de 98%) e poupanças de tempo. O transcriptoma completo de cada amostra é convertido em cRNA e separado em pequenos fragmentos (cerca de 200, 300nt), estes fragmentos são poste-

riormente utilizados como modelos no passo de sequenciação, onde só uma pequena sequência da parte final do fragmento irá ser sequenciada (chamada de *read*). Este processo gera milhões de *reads*, que podem ser depois alinhadas com o genoma e originar uma tabela de contagens para cada gene (número de *reads*, por gene) por amostra.

Algumas ferramentas bioinformáticas foram recentemente desenvolvidas, para analisar esta imensa informação gerada pela técnica RNA-seq. Estas ferramentas diferem entre si quanto à normalização e às técnicas estatísticas aplicadas, com impacto nos resultados finais. Assim, o objectivo da presente tese é explorar e comparar os diferentes métodos aplicados ao problema biológico acima mencionado. Para o efeito, a nossa análise recorreu ao Bioconductor. O Bioconductor é um software de utilização gratuita, não impondo quaisquer licenças de utilização, que disponibiliza código aberto para Bioinformática. Aqui, encontram-se *packages* que permitem processar a informação no que respeita aos 2 passos da análise: normalização e análise da expressão diferencial dos genes. O trabalho desta tese foi desenvolvido considerando a análise dividida nestes dois passos principiais: normalização e expressão diferencial. Os nossos estudos irão ser desenvolvidos em torno das diferentes formas de normalização dos dados, analisando posteriormente os diferentes resultados que se obtiveram na expressão diferencial dos dados.

A normalização é o passo pelo qual se consegue que uma base de dados com contagens profundamente discrepantes entre si possa ser comparável, aplicando a estes dados um denominador comum que toma em consideração os erros associados à utilização desta técnica. A expressão diferencial é o passo onde se identificaram os genes que revelaram significativas alterações da expressão estatística, entre duas amostras, tal como a mutação do gene SETD2 e as linhas das células do ccRCC *wild type*. No essencial, isto significa que estes genes revelam uma alteração significativa da sua expressão, das amostras mutadas para as *wild type*.

A nossa análise baseou-se na utilização de 4 *packages*, `EDASeq`, `edgeR`, `DESeq` e `RankProd`. Enquanto o primeiro foi desenhado apenas para realizar a normalização, o último foca-se apenas na análise da expressão diferencial. Ambos os *packages* `edgeR` e `DESeq` possuem abordagens próprias para realizar ambos os passos, podendo ao mesmo tempo receber contagens de dados normalizados pelo `EDASeq`. O método `RankProd` pode receber dados normalizados pelos métodos `EDASeq` e `DESeq`, e o método `edgeR` também pode receber dados normalizados obtidos pelo método `DESeq`.

A forma como estes dados são normalizados vai depender das premissas que cada método utiliza para normalizar os dados: o `edgeR` e o `DESeq` consideram abordagens diferentes na normalização *between lane*, enquanto o método `EDASeq` realiza uma normalização na própria *lane* relativamente ao conteúdo em GC, antes de proceder à normalização textitbetween lanes. Observámos ainda que todas estas abordagens disponibilizam bons níveis de normalização, bem correlacionados entre si (obtendo valores de 0.98/1.00 no coeficiente de correlação de Pearson) e no que respeita aos dados em bruto.

As combinações estudadas entre os vários métodos tiveram o objectivo de permitir uma comparação detalhada das metodologias, no que respeita aos seus próprios protocolos (normalização `DESeq` combinada com a análise de expressão diferencial com `DESeq` e normalização `edgeR` combinada com a análise da expressão diferencial com `edgeR`) e à junção de protocolos: nomeadamente as diferentes abordagens no passo da normalização pelos métodos - `EDASeq` ou `DESeq` – e ainda as diferentes abordagens no passo da expressão diferencial dos métodos – `edgeR`, `DESeq` ou `RankProd`.

No que respeita a protocolos próprios, observámos que o protocolo completo do método `edgeR` identificou muito mais genes que o método `DESeq`, independentemente dos níveis de significância considerados (1%, 5% e 10%). No que respeita à junção de protocolos, quando se juntou a normalização do `EDASeq` à expressão diferencial do `edgeR`, obteve-se um número significativamente maior de genes identificados quando comparado com quaisquer outros métodos.

O método `RankProd` acabou por apresentar uma nova perspectiva sobre os dados, tendo sido concebido para trabalhar os dados numa lógica de *microarrays*; contudo, ao assumir que cada *lane* dos nossos dados funciona como um *array*, propusemo-nos investigar se este *package* se podia ajustar aos nossos dados de NGS. Observámos que o `RankProd` se ajustava adequadamente ao receber dados normalizados oriundos do método `DESeq`, mas falhava quando recebia dados normalizados pelo método `EDASeq`.

Observámos que, globalmente, o método `DESeq`, quando utilizado como processo de normalização, conduz à identificação de um número menor de genes diferencialmente expressos que o `EDASeq`, para todos os níveis de significância considerados (1%, 5% e 10%). Por outro lado, quando considerados todos os métodos para a expressão diferencial, detectámos que o método `edgeR` identificou mais genes que o `RankProd` ou o `DESeq`, para os mesmos níveis de significância.

Observámos ainda que a maioria dos genes identificados com estes métodos (com excepção dos procedimentos de normalização do `EDASeq`) conduziram a um maior número de contagens para genes *up regulated* que para genes *down regulated*, o que significa que estes genes demonstram possuir maior expressão diferencial quando mutados do que nas amostras *wt*. O passo seguinte na análise foi perceber se os genes que estes métodos identificam como diferencialmente expressos são os mesmos. Considerando todos os métodos do R, identificámos 6 genes diferencialmente expressos comuns a todos eles (para

uma FDR=5%): "SLC2A10", "COL14A1", "GPR173", "LOC100506178", "EREG" e "ADAMTSL1". Uma vez excluído o `RankProd` desta comparação, foram identificados 27 genes como diferencialmente expressos, considerando o mesmo valor para a FDR.

Considerando investigações futuras, e no que respeita aos resultados obtidos pelo Bioconductor, a combinação entre técnicas que revelou um maior número de genes com expressão diferencial é entre o método `EDASeq` para a normalização com o método `edgeR` para a análise da expressão diferencial.

Adicionalmente, todas as combinações de técnicas onde o método `EDASeq` executa a normalização, claramente resultaram em amostras com um maior número de contagens.

No princípio deste projecto, propusemo-nos estudar 6 amostras de transcriptoma, obtidas com o RNA-Seq, duas correspondendo a amostras controlo, as outras quatro apresentando mutações no gene SETD2, o que leva à ocorrência do ccRCC. O desafio foi analisar os dados, procurando identificar genes que reagissem às mutações do SETD2, respondendo à questão de como este gene afecta os restantes genes no transcriptoma. Outro desafio a que nos propusemos, conforme indicado neste texto, foi a comparação de metodologias por forma a melhor aferir sobre o objectivo primário deste trabalho.

Este trabalho permitiu identificar 6 novos genes que respondem às mutações do SETD2.

Palavras cha*ve:* Next Generation Sequencing, clear cell Renal Cell Carcinoma, normalização, Expressão diferencial, SETD2

Contents

List of Figures

List of Tables

Preface

Clear cell Renal Cell Carcinoma, ccRCC, is the most common form of renal cancer, accounting for 7 out of 10 cases, having the highest mortality rate amongst renal cancer. The vast majority of these cancers are due to mutations on the VHL tumor suppressor gene. However, studies in ccRCC sequencing have revealed the occurrence of somatic mutations on SETD2 gene, an Histone methyltransferase that specifically trimethylates 'Lys-36' of histone H3 (H3K36me3). This histone modification is found in actively transcribed genes, revealing an important role of SETD2 in transcription, one of the main steps on gene expression. This gene is located in the short arm of chromosome 3 and mutations in it will lead to loss of function of the chromosome.

The biological goal of the present thesis is to assess the transcriptome alterations induced by SETD2 mutations. This question will be addressed using genome-wide transcriptomic data of ccRCC cell lines with mutated and wild-type SETD2. The data considered to our analysis considered four biological mutated replicates and two biological wid-type replicates. This data was produced by Sergio de Almeida Unit in Instituto de Medicina Molecular (IMM).

Recently, the development of novel high-throughput DNA sequencing methods, designated Next Generation Sequencing (NGS), has provided a new method for both mapping and quantifying transcriptomes, termed RNA sequencing (RNA-seq). Although more expensive than microarray studies and with some data analysis issues still to be solved, RNA sequencing can assess complete transcriptome coverage, providing the ultimate resolution to analyze the levels as well as the structures of both processed and unprocessed transcripts under different conditions. This technique provides both an effective time saving (the entire human genome can be sequenced in less than a week time, depending on the researchers choice) with read accuracy of about 98% while operating

under very reasonable cost & quality beneficts. The full transcriptome of each sample is converted to cDNA and then break into small fragments (around 200-300nt). These fragments are afterwards used as templates in the sequencing step, where only a short sequence of the fragment end will be sequenced (termed ”reads”). This process generates millions of reads that can be aligned to the genome and originate count table data for each gene (number of reads per gene) on each sample.

Several bioinformatics tools have been developed recently for analysis of the such a big amount of data originated from RNA-seq technology. These tools differ in the normalization and statistical methodologies applied, that can influence the final results. Thus, the present thesis aims to explore and compare the different methods, applying to the biological problem described above. we analysed it resorting to `R` Bioconductor packages. Bioconductor is a free open source software tool that provides ways to process the data regarding the two steps of the analysis involved: normalization and differential expression gene analysis.

This thesis workflow considers the analysis in two major steps: normalization and differentially expression analysis. Our studies will be performed in different ways for data normalization and report how this resulted in different outputs for the data expression analysis.

Normalization is the way we allow the very discrepant count data to be comparable to each other, rendering it to a common denominator, which accounts for bias associated with this technique. Differential expression gene analysis is the step where we identify the genes that show statistically significant expression alterations between two samples, such as SETD2 mutated and wild type ccRCC cell lines. Essentially, this means that these genes report a significant expression change from the mutated to the wild type samples.

Our analysis was based on four packages, `EDASeq`, `edgeR`, `DESeq` and `RankProd`. While the first is designed just to perform normalization, the latter focuses on differential expression analysis. Both `edgeR` and `DESeq` packages have their own approaches to perform both steps, at the same time being able to receive normalized count data from `EDASeq`. `RankProd` received normalized count data from both `EDASeq` and `DESeq`, and `edgeR` also received as input the normalized count data obtained with `DESeq`.

These count normalized data matrix will depend on the assump-

tions each method uses to normalize the data: `edgeR` and `DESeq` take different approaches normalizing between lanes, while `EDASeq` performs a GC content within lane normalization, before performing a between lane normalization. We observed that all of these approaches provided good normalized data, properly correlated (leading to 0.98 and 1.00 Pearson correlation coefficient) amongst them and regarding the raw data.

The combinations studied between the several methods were meant to provide a detailed comparison of the methodologies, regarding their self-protocols (`DESeq` normalization with `DESeq` differentially expression analysis and `edgeR` normalization with `edgeR` differentially expression analysis) and joined-protocols: namely inferring differences regarding different approaches on the normalization step – `EDASeq` or `DESeq` – and regarding different approaches on the differential analysis step – `edgeR`, `DESeq` or `RankProd`. Regarding self-protocols, we observed that `edgeR` full protocol identified far more genes then `DESeq` for all the significance levels considered (1%, 5% and 10%). Regarding joined-protocols, `EDASeq` normalization coupled with `edgeR` differential expression led to the greater number of identified genes amongst all the methodologies.

RankProd provided an interesting alternative perspective on the data. This package is designed to work with microarrays experiments; however, by assuming each lane of our experiment to function as an array, we proposed to see whether this package could adjust to our NGS data. We observed that `RankProd` seemed to adjust properly when provided with normalized data from `DESeq`, but it failed to adjust as good when provided with `EDASeq` normalized count data.

We observed that, globally, `DESeq` method, when used as a normalizing procedure, leads to lower identified differentially expressed genes than `EDASeq` for all the significance levels considered (1%, 5% and 10%). On the other hand, when considering all the methods for differential analysis, we detected that `edgeR` identified more genes than `RankProd` or `DESeq` to the same significance levels.

We also observed that the majority of genes identified with the methods (with exception to `EDASeq` normalization procedures) led to bigger counts for up regulated genes than for down regulated genes, which means that these genes show a higher expression in mutated then in *wt* samples.

The next step in the analysis was to observe whether these methodologies point to identifying the same differentially expressed

genes. Considering all `R` methods, we identified 6 genes as differentially expressed, in common for all methodologies (for a FDR of 5%): "SLC2A10", "COL14A1", "GPR173", "LOC100506178", "EREG" and "ADAMTSL1". When excluding `RankProd` from this compared analysis, 27 genes were identified as differentially expressed to the same FDR value considered.

Considering future researches and as per bioconductor results, the techniques combination that delivered a greater number of differentially expressed genes is `EDASeq` for normalization, combined with `edgeR` in differential expression analysis.

Furthermore, all technique combinations where `EDASeq` handles normalization clearly result in samples having a higher number of counts.

At the beginning of this project, we proposed to study 6 transcriptome samples obtained using RNA-Seq, two corresponded to control samples, the other four presented mutations on SETD2 gene, leading to ccRCC cancer expression. The challenge was to analyse this data, looking for genes that eventually reacted to SETD2 mutations, responding how this gene affects the remaining transcriptome. Another challenge we set ourselves to, as indicated in the text, was the comparison of methodologies in order to more accurately assess on the primary objective of this work.

Our studies led to the identification of six new genes responding to SETD2 mutations.

Chapter 1

Biological Background

This chapter is meant to give some background knowledge in order to fully understand the biological questions this thesis proposes to answer. It offers a description of the biological problem in study – renal cancer – and it also describes some important notions on genetics, which are necessary to understand the technique used to obtain the data – Next Generation Sequencing. After clearing all the theoretical concepts, the objectives established for this work are stated.

1.1 Genetics

Genes are segments of DNA involved in producing a protein. They include a region preceding and following the coding region, as well as intervening sequences (introns) between individual coding segments (exons). DNA is a long double stranded helix of molecules composed of sugar, phosphate and four different nucleotides, A for adenosine, T for thymidine, G for guanosine and C for cytidine. The nucleotides are arranged in triplets called codons, each one codifying a specific aminoacid which, in turn, will bind forming proteins.

Each nucleotide is constituted of a 5-carbon sugar that binds to one or more phosphate groups and to one nuclear base (A for adenine, T for thymine, G for guanine and C for cytosine), the nuclear base being the naming factor for the nucleotide. There are about 3 billion base pairs forming the genes that compose the double helix of human DNA, being the basis bound two by two – A pairs with T, G pairs with C – and this genetic information is copied to every single cell (Lewin, 2004).

It is known that a base pair (bp) is 0.34 nanometers long (a meter corresponds to 1,000,000,000 nanometers) and that the typical cell size is around 10,000/100,000 nanometers long (e Silva *et al.* , 2008), so the question that arises is: how is it possible that so many basis can get into a single cell?

The answer to this question lies in the extreme level of packing of the DNA. The long molecules of DNA containing the genes are organized into chromosomes (figure 1.1) and different species have a different number of chromosomes, with a specific number and order for the nucleotides. Humans have 23 pairs of chromosomes and the entire set of 23 human chromosomes is called the human genome. This means that the 23 pairs of chromosome make up for 6 billion base pairs of DNA per cell, which translates in about 2 meters of DNA $[(0.34 \times 10^{-9}) \times (6 \times 10^{9})]$ per cell (Damaschun *et al.* , 1983). How is this packing achieved?

In human genetics, within the chromosomes, the DNA is packed into chromatin. Chromatin consists of DNA and structural proteins. Within the chromatin itself, the repeated unit is the nucleosome, these are constituted by a portion of DNA wrapped around proteins called histones. Nucleosomes contain 9 histone proteins, H1, H2A, H2B, H3 and H4. Two of each of H2A, H2B, H3 and H4 form the eight protein complex of the nucleosomes core. H1 is then added to the formation to maintain the DNA wrapped in place. Histones are responsible for maintaining chromatin shape and structure and, therefore, responsible for compacting the DNA into a smaller volume.

It is important to state that each gene, each fragment of DNA, encodes a specific protein that expresses a particular trade, which goes from hair color to risk for certain diseases. Most important, regarding this thesis, genes regulate the biochemical processes that occur in the body, it is therefore extremely important that these genes are properly expressed, so they can play their role.

The process by which the information that genes carry (regarding their function) is used to synthesize a protein, is called gene expression (figure 1.2). It has four main steps: transcription, splicing, translation and post-translational modifications. Transcription is the process by which individual genes are copied into RNA molecules: an mRNA (messenger RNA) is created from a DNA template, resorting to a series of transcription factors (an mRNA is a molecule of RNA whose sequence is complementary to the coding

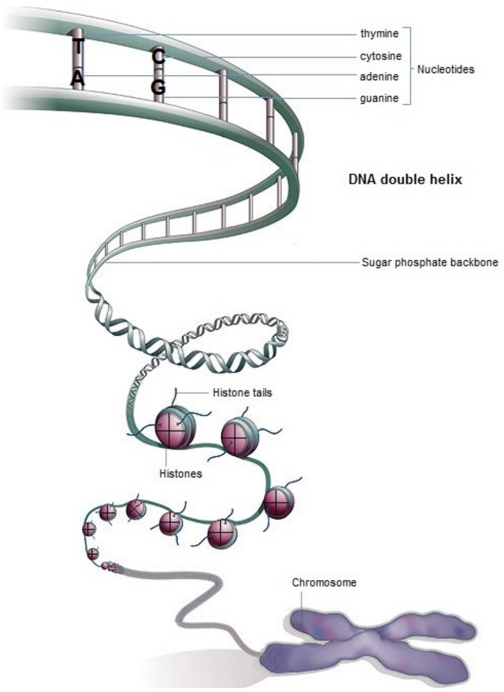

Figure 1.1: Zooming in on a chromosome. The image emphasises the great level of DNA coiling, the four nucleotides, bound together as described, form the DNA double helix that, resorting to histones, coils into chromosomes – figure adapted from (Cheng-Fu, 2013)

.

sequence of one of the strands of the DNA). The collection of these RNAs formed in transcription is called transcriptome. Following transcription, there is splicing, where the non-coding regions - introns - are removed and exons are joined together, leading to the finished product of mRNA. The next step is translation, where the mRNA is translated to polypeptide chains that then suffer post-translational modifications, before becoming the mature protein product (Lewin, 2004).

The chromatin is known to play a key role in the regulation of gene expression, with particular emphasis on transcription. On a closer look at this phase, transcription involves a copy of one of the DNA strands so, at a given point of gene expression, the two DNA strands are separated. This means that something must happen to break the tight bonds that coil the DNA around the histones. Histones will temporarily be removed, making the DNA accessible to transcription. The way the cell is designed to do this

is by modifying histones (Talbert *et al.* , 2012).

There are a few proteins that, given the correct signal from the cell, will trigger mechanisms which will lead to certain alterations on histones, making the chromatin more accessible. Some of those modifications can be acetylations, methylations, di-methylations, tri-methylations, phosphorilations or ubiquitinations, respectively corresponding to the addition of an acetyl group, $CHCO_3$, one, two or three methyl groups, CH_3, a phosphate group PO_4^{3-} or a protein called ubiquitin. The core histones (H2A, H2B, H3 and H4) have long proteic tails (mentioned in figure 1.1) and it is in different positions of these tails where the modifications will occur.

The vastness of modifications that can happen in all 5 types of histones, at different points of these proteins, led to the need of creating a proper nomenclature to identify them. Several authors postulated their views, Bryan Turner (Turner, 2005) proposed that histone modifications would be characterized first by stating the modified histone, then the residue that is altered and last the type of modification suffered. Exemplifying with the practical example of this work, H3K36me3 is regarding a 3 methylation (me3) on the aminoacid lysine(K) which is in position 36 of histone 3 (H3) tail.

1.2 clear cell Renal Cell Carcinoma (ccRCC)

Kidneys (figure 1.3A) are the main organ of the urinary tract system, being responsible for filtering the blood, reabsorbing main nutrients and water and eliminating what the body does not need, as urine. They play a very important role in maintaining, within normal values, the hormonal, the electrolytes and the blood pressure levels. Kidneys play their functions at its basic functional structure, the nephron (figure 1.3B), which is composed by the renal corpuscule (Bowman's capsule) and renal tubule. It is highly irrigated (figure 1.3C) in order to allow for reabsorption. The components meant to be reabsorbed follow through bloodstream, and those to be eliminated go through the collecting duct (Cohen & McGovern, 2005). These processes are based in physiological and biochemical tight regulation, and errors affecting them can have severe repercussions.

Cancerinogenesis is the process that leads to cancer generation. By definition, cancer occurs when there is growth of abnormal cells. This growth is generated when there are errors in vital regu-

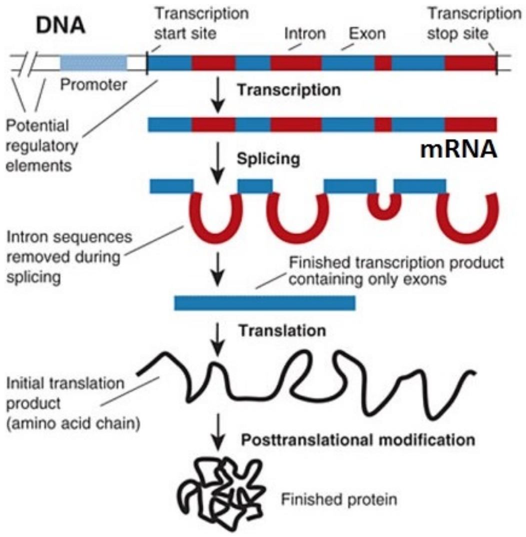
DNA
Transcription start site
Intron
Exon
Transcription stop site
Promoter
Potential regulatory elements
Transcription
mRNA
Splicing
Intron sequences removed during splicing
Finished transcription product containing only exons
Translation
Initial translation product (amino acid chain)
Posttranslational modification
Finished protein

Figure 1.2: Gene Expression mechanism. Several processes need to occur to obtain a finished protein: the DNA is first transcribed to mRNA, which then suffers splicing events to remove the non coding segments of the mRNA. Only then the protein is translated into polypeptide chain, these suffer posttranslational modifications that lead to a protein, image adapted from (TheMedicalNews, 2013)

.

latory pathways. In most situations, the accumulation of errors in the cellular machinery that lead to cancer happen by somatic mutations. These mutations happen after conception, meaning that they are not inherited, they occur only on the daughter cells. Cancers progress when multiple cycles of mutations happen. Various causes can lead to these errors on the pathways - many cancers are generated by loss rather than the increase of gene function (recessive and dominant genes respectively). In any case, altered DNA nucleotides (mutations) are the basis of cellular changes that cause cancer - this includes chemical alterations of individual nucleotides or the order in which nucleotides occur.

Several factors can lead to cancerinogenesis, like smoking, diet habits (fat, water, fibre and vitamins being key topics), sex hormones and family history. Age is also a factor: as people get older, the most likely it is that some mechanisms loose specificity, leading to malfunction and possibly shutting down (Latchman, 2007).

Renal Cell Carcinoma, RCC, occurs when malignant cells form in the proximal tubule, at the cortex. It accounts for 9 out of 10

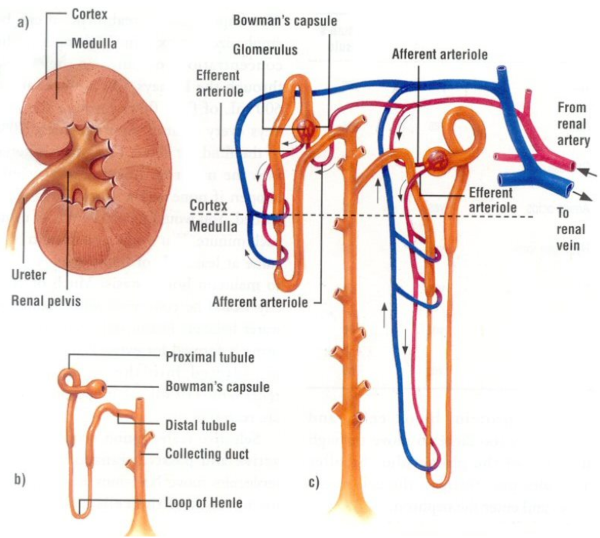

Figure 1.3: Schematic representation of the urinary tract system. (A) The picture goes from the kidney, (B)to its basic structure, the nephron, (C) emphasizing how irrigated it is, in order to better reabsorb and eliminate nutrients (Gallant, 2013)

.

kidney cancers. There are several subtypes of tumors of RCC, including clear cell RCC, ccRCC, papillary RCC (type I and type II), chromophobe RCC, collecting duct RCC, and unclassified RCC. ccRCC is the most common tumor in the kidney, 7 out of 10 kidney tumor cases are ccRCC (Singer *et al.* , 2012).

About 60% of ccRCC cases occur when both alleles of the VHL tumor suppressor gene are mutated (leading to its inactivation) – this gene is absent from most other cancers (Nabi *et al.* , 2010). However, recent studies on ccRCC sequencing have identified, amongst the genes that are mutated in ccRCC, somatic mutations in SETD2, a methyltransferase that mediates tri-methylation of lysine 36 of histone 3 (H3K36me3) during transcription, and also, that the majority of these mutations (82%) had an hypoxic pattern of expression, which means they occur induced by oxygen free conditions (Carvalho, 2012). As mentioned, histones are the key component in chromatin, which is responsible for DNA packing. So, it is clear that SETD2 is a chromatin-modifying gene, however its precise role on transcription is not yet clear (Dalgliesh *et al.* , 2010). Another role has been found for this gene, that is, SETD2 has a key part in activating the major regulator on DNA

Damage Response, DDR, regulating the accuracy of the DNA damage signalling and repair. But how the gene acts at the sites of DNA damage is also an unanswered question.

1.3 Next Generation Sequencing (NGS)

Sequencing a segment of DNA means determining the exact order of these base pairs in the DNA chain. DNA sequencing is a mile stone on understanding genetics and it can be used to determine sequences for single genes, for chromosomes or even an entire genome sequence. And that is exactly what a group of american scientists proposed to do, with the "Human Genome Project". The US Department of Energy and the US National Institute of Health set up the Office of Human Genome Research and, together with geneticists from all over the world, began, in 1990, the project of determining the sequence of the human DNA. The full human genome sequence was completed in 2003. Back then, they used Sanger's technique to determine the long human genome sequence. With this method, the bases of a small fragment of DNA are sequentially identified from signals emitted as each fragment is re-synthesized from a DNA template strand, using fluorescence or a radioactive element. It determines the order of the bases one at a time, and that is why it took 13 years. Approximately 3 billion dollars where needed to determine the sequence of the 3 billion chemical base pairs of the human DNA (Chial, 2008).

Another important step on DNA sequencing were the Microarrays. These are 2D arrays that sample several probes at the same time, for instances, if the goal of a certain study is to examine a specific region of the human genome of several individuals. What the arrays allow to do, relatively to Sanger's technique, is to test all of the samples from the several individuals in one single array, saving up a significant amount of time and money (Diamandis, 2000).

Sequencing has come a long way since Sanger, and today there are second and third generation sequencing. Nowadays, with the advent of new technologies, the human genome can be sequenced in a week time, and the cost is drastically lower, around 5,000 USD (ORNLaboratory, 2013). This is resorting to Next Generation Sequencing, NGS, the technology used in this thesis.

The principal in both techniques is the same: the bases of a

DNA fragment are sequentially identified from signals emitted, as fragments are re-synthesized from a DNA template strand. The big difference is that NGS techniques do this resorting to massive parallel sequencing platforms, which allows them to have up to millions of reads per DNA sample (Nowrousian, 2010).

The NGS technique used is this thesis, Illumina Solexa Genome Analyser, immobilizes sequencing templates on a flow cell. This flow cell is a solid platform that has eight lanes, the lanes constitute the basic unit of this technology, and in each one there is a sample in test, that will lead to an extreme amount of output. There are three main steps to this process: first there is the library preparation (figure 1.4A), then the cluster amplification (figure 1.4B) and then sequencing (figure 1.4C). When preparing the samples, the DNA is randomly fragmented and adapters are bound to these fragments, making up for the ligated DNA. The single stranded fragments will bind to primers on a solid surface of the flow cell, trough a process called bridge Polimerase Chain Reaction, bridge PCR. PCR is a technique used to amplify copies of DNA, generating copies of a particular read. It uses very small DNA sequences, the primers, that essentially operate as triggers for the fragments copy. In bridge PCR, these primers are linked to the solid-support surface, and they bind both ends of the sequence of interest adapters, forming a kind of an arch – hence bridge PCR – unlabelled nucleotides are then added for fragment amplification, generating clusters of unique DNA fragments (Berglund *et al.* , 2011). These PCR colonies will then be sequenced, through enzyme driven biochemistry processes and using fluorescent irreversible tagged dye terminators. They are sequenced to either be aligned with a reference genome or to be used to assemble a DNA sample – which is called *de novo* sequencing. The full set of aligned reads reveals the entire sequence in study (Voelkerding *et al.* , 2009).

Figure 1.4 is from a paired-end sequencing. To avoid regions on the sequence where no reads align to, NGS techniques can be used for paired-end sequencing. Here, the fragment is sequenced from both ends, providing two reads for each fragment, which translates into a superior alignment across regions that have repetitive sequences, while, at the same time, allowing to produce longer overlapping sequencing reads, filling the gaps. This results in a complete overall coverage. Furthermore, the distance between each paired read is identified, and alignment algorithms use this in-

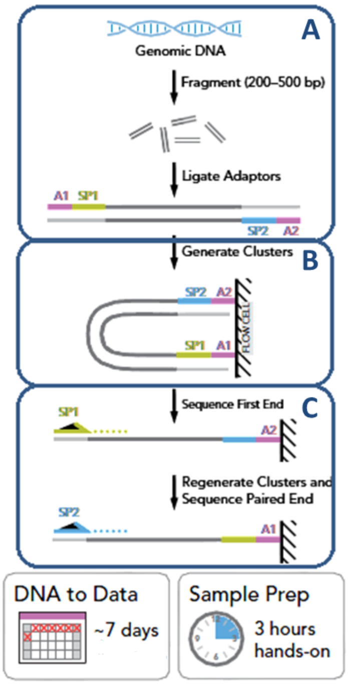

Figure 1.4: NGS - Genome analyzer workflow separated by its main steps, (A) first is the library preparation, (B) to which follows cluster generation (C) and sequencing. NGS techniques can sequence samples as big as the human genome in a matter of days with little sample preparation, image adapted from (Illumina, 2013)

.

formation to map the reads on the repetitive regions with more accuracy (Nowrousian, 2010).

In this thesis, the sequencing technology was applied in high-throughput mRNA sequencing, RNA-Seq, to analyse the transcriptome, estimating every transcripts abundance. The methodology follows the three steps mentioned before but, with RNA-Seq, the quantification of gene expression levels is done by first converting RNA transcripts into complementary DNA fragments, cDNA, and then these fragments proceed to being sequenced leading to the short reads (Oshlack *et al.* , 2010).

1.4 Objectives

This thesis aims at analysing NGS data of ccRCC samples. For that matter, in this project, transcriptome data from six different cellular lines was analysed: two wild type samples, wt, and four mutated samples on the SETD2 gene. As previously stated in this chapter, SETD2 encodes proteins that are involved in histone modifications, meaning that this gene plays its functions in DNA packing, being therefore extremely important its tight regulation, in order to have proper processes of gene transcription and consequent active tumor suppressor genes.

By comparing mutated cellular lines on this gene to non mutated cellular lines, we proposed to study how this genes mutations may or may not impact the remaining genes. So the main focus of this project is to identify variations between the two kinds of cellular lines, regarding the genes that compose their sequence, not only to identify those that are differentially expressed, but also to understand to what extend SETD2 influences the other genes that compose each sample.

The data to the six cellular lines were obtained resorting to the NGS Illumina/Solexa technique and were analysed with different Bioconductor packages: `edgeR`, `DESeq`, `EDASeq` and `RankProd`. The main question here was to assess how these methodologies (with different focus on the data analysis and alternative approaches to it) evaluate the data: do they identify the same genes? And if so, with what significance?

There are two steps on the analysis: normalization (to make the data comparable) and differential expression (to assess whether a gene is significantly expressed). While `EDASeq` can only perform the first and `RankProd` the second, `edgeR` and `DESeq` do both. And this leads to another question regarding the methodologies: given that each method has its own protocols for the two separate parts of the analysis, could there be a cross analysis between them that provides more interesting results? To this extend, separate studies regarding both phases of the analysis were performed, where the same normalization procedures were applied (EDASeq and `DESeq`) and the same differential analysis procedures were studied (`DESeq` and `edgeR`). RankProd is a package designed to identify genes that are differentially expressed in Microarray data. This package provides another objective to this thesis: it was studied here to infer if this specific method for microarray analysis can extend to NGS data.

Chapter 2

Methods for data Analysis

The present chapter contains a description of the methodologies applied on this thesis. It focuses first on the technique used to obtain the data (NGS) and then on the two step analysis of the gene counts obtained with NGS: different methods are applied to achieve normalized data, and different methods are used to determine which genes are differentially expressed (i.e. biologically significant). Both phases were analysed with `R` software.

Hereinafter, in order to unify the methodologies, Y_{gi} is considered as the number of reads (*i.e.* the coverage) from sample *i* mapped to gene *g*. The set of genewise counts (read counts) for sample *i* makes up the expression profile or library (i.e. sequencing depth) for that sample. The library size refers to the number of mapped short reads obtained from the libraries sequencing process.

2.1 NGS data analysis workflow

The NGS technique used in this work is the Illumina/Solexa Genome Analyzer, with the Illumina HiSeq2000 model. The technology is able to process up to three billion reads, each around 100 bp, resulting in a total amount of count data around 600 Gb, with an accuracy of base calling of about 99.5% (Zhang *et al.* , 2011).

The pipeline (figure 2.1) to process this immense quantity of output is based on the processing of different types of file formats (figure 2.2). NGS outputs a massive load of counts, since it measures how many reads align to what specific part of the sequence in study (Kadota *et al.* , 2012). The volume of information it returns makes it impossible to be managed on a normal PC. Users

must connect to a proper server and use, via command line, the processors for NGS data.

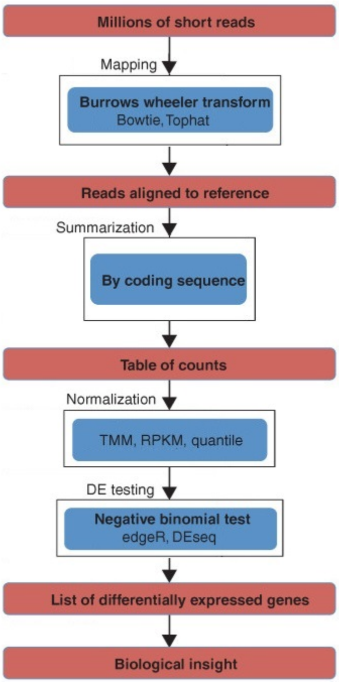

Figure 2.1: Methodology applied in RNA-Sequencing. The steps are marked on red boxes and the methods used are marked with the blue boxes. The reads are mapped to a reference genome resorting to Bowtie. These reads are then summarized into a table of counts which shows how many genes were aligned to the sample under analysis. These counts are normalized resorting to different methodologies and only then is applied statistical testing to determine genes that are DE (image adapted from (Oshlack *et al.* , 2010)

).

With this in mind, the first step is to assess data quality. This can be done resorting to several tools, such as FastQC, PRINSEQ or FASTX. Focusing on FastQC, this tool reads *.fastq* files into *.fastaqc* files. These files are obtained by processing the short reads sequence and attributing a quality score to every base that constitutes it. Hence, these files contain information regarding the read quality. This provides a plateau on the analysis: provided the reads are good the analysis proceeds, if they are not (in other words, if they exhibit unusual qualities, which is a synonym of poor sequencing quality itself, or sequencing contamination), then

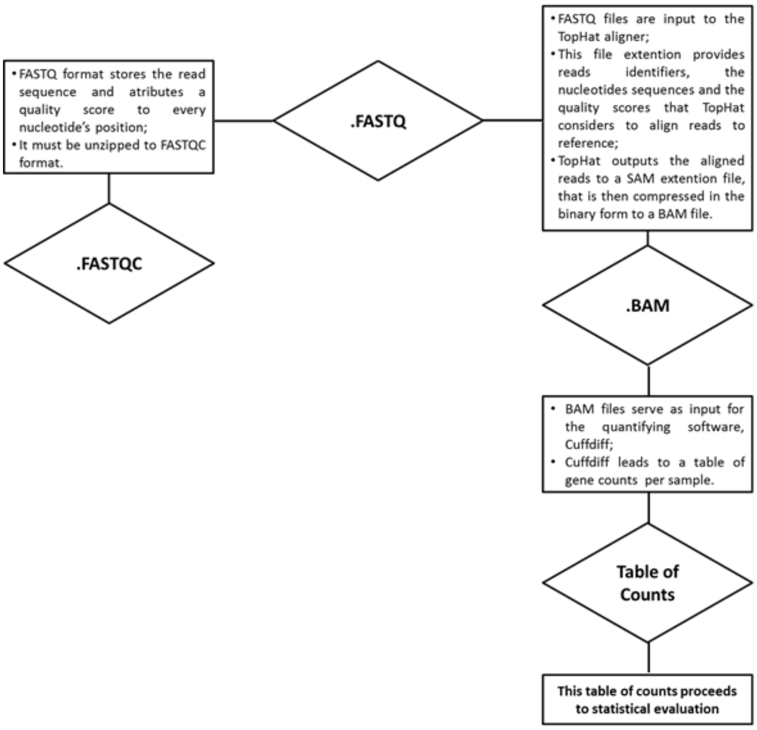
• FASTQ format stores the read sequence and atributes a quality score to every nucleotide's position;
• It must be unzipped to FASTQC format.
.FASTQ
• FASTQ files are input to the TopHat aligner;
• This file extention provides reads identifiers, the nucleotides sequences and the quality scores that TopHat considers to align reads to reference;
• TopHat outputs the aligned reads to a SAM extention file, that is then compressed in the binary form to a BAM file.
.FASTQC
.BAM
• BAM files serve as input for the quantifying software, Cuffdiff;
• Cuffdiff leads to a table of gene counts per sample.
Table of Counts
This table of counts proceeds to statistical evaluation

Figure 2.2: Different file extensions involved on the NGS data analysis processing.

some reparametrization on the NGS technique is required (CBCB, 2013).

The millions of short reads, after quality assessment, will be aligned to a reference resorting to the TopHat mapper. TopHat is based on the short reader aligner Bowtie, and uses it to align reads to a reference. The difference between these two softwares is that TopHat allows gap alignment as well. TopHat will align the reads to the reference using the *.fastq* files as input (which have the gene coordinates) and return the aligned reads, and the reads that overlap, onto the file extension *.sam*. These files will be comprised into *.bam* files, which store the same information but in binary code (CBCB, 2013). The mapping is done with the Borrows-Wheeler algorithm (which is ideal for short reads that have to be aligned to a big reference sequence) and with reads that align only one time, so the obtained reads belong to that gene and that gene only (Oshlack *et al.* , 2010).

The mapped reads are assembled into an expression summary. For this study, we will consider gene-level counts, which means that the reads are assembled to a gene-level expression summary *i.e.*, the number of reads that fall into a given gene, which is done resorting to Cuffdiff. This software receives the *.bam* files (which are the compressed version of the *.sam* files) with the aligned reads obtained per gene and then counts them, outputting the information on a table of counts that then proceeds to statistical evaluation.

An example of a simple table of counts obtained with this methodology is as follows in table 2.1, where the *g* genes are indicated on the rows, and the columns indicate the *i* samples or libraries. Each cell will correspond to the number of reads aligned to the gene in that sample.

gene	mut_1	control_1	control_2	mut_2
1	154	298	120	35
2	16	831	4	273
3	16	155	1	35
4	218	9	63	39
5	982	385	325	163
6	14	5	1	1

Table 2.1: Gene expression matrix example.

When proceeding to the data evaluation, there are two very important steps involved: normalization (chapter 2.2) and differential expression gene analysis (chapter 2.3). These two steps will lead to a list of differentially expressed genes, which constitute the biological insight of this project.

2.2 Methods for Normalization

This is the key step leading to the analysis of gene's Differential Expression, DE. To better understand why, Robinson & Oshlack (2010) propose a scenario where a large number of genes are uniquely aligned to one of the two experimental conditions in study (in this thesis, mutated or wild type). What happens is that the real sequencing state available for the other genes is shortened. If a large amount of sequencing is dedicated to a specific experimental condition, then there is less sequencing available for the remaining genes. This must be adjusted, otherwise it will lead to a skewed analysis towards one of the experimental conditions (Robinson & Oshlack, 2010),(Oshlack & Wakefield, 2009).

It is therefore imperative to make this data comparable, in a step called normalization. The purpose of this step is to weigh-in systematic technical effects that occur in the data and remove them, to ensure that systematic bias have minimal impact on the results. A few sources of systematic variation in RNA-seq can be referred. For instance, larger library sizes will typically lead to higher counts for the entire sample, which is described as the overdispersion problem. Another example is that read counts are generally proportional to the gene length. It is important to understand that normalization is necessary in order to only consider sample-specific effects on the analysis, hence removing systematic variations (Robinson *et al.* , 2010).

During the last 4 years, some normalization approaches on how to treat RNA-seq data have been studied, resulting in several available methodologies, each one considering different assumptions, algorithms, approximations and aspects of the data, to adjust for different bias. But, despite all these studies, there is still not a consensus towards how the choice of normalization influences the downstream analysis (Dillies *et al.* , 2012).

Bioconductor is an open source software that uses `R` statistical programming language. This software provides tools for the high-throughput genomic data analysis, conveying numerous software packages designed for the analysis of: DNA microarray, DNA sequence, a process responsible for genetic variability called Single

Nucleotide Polimorfism (SNP) and other data, such as RNA-seq (Bioconductor, 2013). Some of these packages are designed only for data normalization, while some perform both normalization and DE analysis (chapter 2.3).

Considering the methods that have their own protocols for normalization and for the identification of differentially expressed genes, DEG, two methods are described in this thesis: `edgeR` and `DESeq` (with Bioconductor). Regarding the packages that focuses specifically on normalization, we used the package `EDASeq`.

2.2.1 `edgeR`

This package was created in 2008 by Robinson, McCarthy, Chen, Lun and Smyth. Although scaling to library size makes sense, and there are ways to do this (like computing the proportion of every genes reads rendering them to the total number of reads and then comparing it across samples) this might not be enough as a normalization choice. The number of reads that align to a gene is also function of the composition of the RNA population that is being sampled: different experimental conditions express different RNA repertoires and the proportion might not always be directly comparable, which can lead to an over or under sampling effect, misleading DE calls.

With this in mind a new method for normalization was presented by Robinson and Oshlack: the trimmed mean of M-values normalization method, TMM (Robinson & Oshlack, 2010).

Let us consider that the expected value of Y_{gi} is a function of: the true, yet unknown, expression levels (number of reads), μ_{gi}; of the length of the gene L_g; and of the total RNA output of a sample N_i, as follows:

$$E[Y_{gi}] = \frac{\mu_{gi} L_g}{S_i} N_i \tag{2.1}$$

$$where \quad S_i = \sum_{g=1}^{G} \mu_{gi} L_g \tag{2.2}$$

What equation 2.2 means is that S_i is the sum of the number of reads obtained for every single gene in all libraries, times the length of that library – it represents the total RNA output of a sample, which is unknown and, as mentioned before, can vary a lot according to the RNA population. S_i can not be estimated directly because there is no way of knowing for sure μ_{gi} or L_g for every gene.

On the other hand, the relative RNA production of two samples, represented by $f_i = S_i/S_{i'}$, can (Robinson & Oshlack, 2010).

Based on this quantity, Robinson and Oshlack assembled a method that equates the overall expression levels of genes between samples, under the premise that most genes are not DE (Oshlack & Wakefield, 2009). What they did was to consider a trimmed mean of the data, where both the log fold-change, log FC, between library *i* and library *r* (M^r_{gi}) and the absolute intensity (A_g) were cut in order to level the estimates of RNA production, defining them as:

$$M^r_{gi} = log_2 \frac{Y_{gi}/N_i}{Y_{gi'}/N_{i'}} \tag{2.3}$$

$$A_g = \frac{1}{2} log_2 (Y_{gi}/N_i \cdot Y_{gi'}/N_{i'}) \tag{2.4}$$

The authors then calculated normalization factors by selecting one sample as reference, and calculating TMM factors for the other samples relatively to this reference, so that for every tested sample, TMM is computed as the weighted mean of log ratios between the test and the reference sample, excluding the most expressed genes and the genes with the highest log ratios. Given the hypothesis that the majority of genes are not DE – the null hypothesis – this TMM should be close to one. If it is not, then the value obtained provides the correction factor to be applied to the library sizes in order to be in agreement with H_0. In `R`, the function `calcNormFactors()` calculates these normalization factors. To obtain the normalized read counts, the software must consider these normalization factors and re-scale them by the mean of the normalized library sizes. The normalized read counts *per se* are retrieved by dividing the raw read counts by these re-scaled normalization factors (Dillies *et al.* , 2012). This method provides a robust way to weight in relative RNA production levels, with the normalization factors proceeding directly to DE analysis.

2.2.2 `DESeq`

`DESeq` was developed by Anders and Huber (2010). `DESeq` and `edgeR` both base their normalization method on the hypothesis that most genes are not DE, although they adopt different approaches. Being that said, Anders and Huber (Anders & Huber, 2010) developed a method of normalization where, along with Y_{gi}, they introduce,

s_i, the size factor which stands for the effective size of library i. In the normalization step, m size factors, s_i, are estimated from the count data, resorting to the median of the ratios of observed counts for all genes, as follows:

$$\widehat{s}_i = median_i \frac{k_{gi}}{(\prod_{v=1}^{m} k_{gv})^{1/m}} \tag{2.5}$$

The denominator of equation 2.5 is interpreted as a reference sample, to which every size factor is compared (Anders & Huber, 2012).

The authors created a method that, like `edgeR`, takes into account that the total number of reads might not be a factor good enough to normalize the data. They admit that there might be some highly differentially expressed genes that can have a big influence on the total read counts, and this will lead to a biased DE analysis, if not normalized. They then devised a method where each sample – column – is divided by the geometric mean of the rows – genes – and the median of these ratios is the sizing factor for the sample in question (Rapaport *et al.* , 2013).

In `R` the normalization is reached with the functions `estimateSizeFactors()` and `sizeFactors()`, where the size factors are computed for each sample, and the counts are divided by the factor associated with that sample (Anders, 2010).

2.2.3 EDASeq

Thus far, it has been stated in this chapter, how some fragments characteristics make them preferentially detected with RNA Seq techniques (Hansen *et al.* , 2012). An example that has already been stated, is that longer genes tend to bias the analysis, as they typically tend to have more reads aligned to them. Another strong example is the effect of GC content, which has been shown to affect DNA related measurements, such as RNA Seq (Pickrell *et al.* , 2010). Pickrell *et al* have demonstrated GC-content effect can change from sample to sample and Benjamini & Speed (2007) have demonstrated that both genes with high content GC and low content GC reveal this sample specific effect (Benjamini & Speed, 2012).

With this knowledge regarding GC-content effect, Risso and Dudoit created `EDASeq` package, 2010. These authors took a different approach from the two methods described before: they consider that there are two types of effects on read counts, within-lane gene-specific effects and between-lane distributional differences effects. These two types of effects lead to a two step normalization process: within lane normalization accounts for GC content or gene length biases, while between lane normalization focuses on normalization for the sequencing depth (Risso, 2011). In essence, the approach this authors made is different from `edgeR` and `DESeq`, because the `EDASeq` package has this dual approach, by first considering a lane-specific normalization – GC content or gene length (within normalization) – and only then accounting for the sequencing depth (between normalization)(Bullard *et al.* , 2010). The normalization is achieved with `withinLaneNormalization()` and `betweenLaneNormalization()` functions.

The authors consider several options for either within and between normalization. Within-lane normalization has four approaches to adjust for GC content or to gene length effect on the sample: loess robust local regression, global-scaling, using the median or the upper quantile, and full-quantile normalization. Loess regression will perform a regression on the data, according to the gene effect of interest (either GC content or gene length) (Risso *et al.* , 2011). The three approaches that use quantiles will be function of a defined number of equally sized bins. These bins divide the data according to GC content in several stratus: global scaling using the median will scale the data to have the same median for each bin, global scaling using the upper quantile scales the data to have the same upper quantile and full-quantile normalization will take the several bins quantiles and pair them in order to obtain the median for every quantile. This is an approach similar to microarrays where, for each lane, the distribution of read counts is matched to a reference distribution, that is defined according to the median counts of the sorted lane (Bullard *et al.* , 2010).

Between-lane normalization adjusts for lane sequencing depth, *i.e.*, by the number of total read counts per lane *i*. This normalization aims at rendering lane differences, making the samples comparable. The authors postulated three different types of normalization procedures, the same above referred: global-scaling normalization using upper quantile, global-scaling normalization using the median and full quantile normalization. The way these normalizations process the quantiles is the same as described for within normalization but applied to the lanes in study. Hence, global scaling using the median will force the median of each lane to be the same, global scaling using the upper quantile will force the upper quantile of each lane to be the same and full quantile normalization will take the quantiles of each lane and pair the median of every quantile (Bullard *et al.* , 2010).

This package focuses specifically in data normalization, creating an `EDASeq` object of normalized data count, to be used for DE analysis by `DESeq` and `edgeR`.

2.3 Methods for Differential Expression Gene analysis

The previous normalization step will lead to normalized count data, on the form of a matrix, where each cell will correspond to the number of reads aligned to gene g in sample i. This count data matrix will constitute the input for a differential expression gene analysis. DE analysis is used to measure differences in expression levels between two conditions, allowing the analyst to infer about the genomic structure under study, either a known sequence, or a *de novo* sequencing.

As normalization takes into account different considerations that lead to different normalized count data, DE analysis also has methods based on distinct premises, that lead to alternative ways to achieve differential expression.

Being that said, different statistical methods can be assumed for DE analysis. The selected methods to be presented in this thesis are: `edgeR`, `DESeq` and `RankProd`. `RankProd` has the characteristics of not only considering a non-parametric approach – while all others assume parametric assumptions – but also being meant for microarray analysis. Other methods examples are NOISeq, DEXseq, DEGseq, which were not considered in our analysis, since these are not as used in the literature.

The considered methods are all based on the null hypothesis that genes are not differentially expressed against the alternative hypothesis that they are differentially expressed. The five methods provide statistical elements that allow the user to infer about differential expression analysis and gene regulation. To be mentioned, and of key importance, are the log fold-change, log FC, and the false discovery rate, FDR (Benjamini & Hochberg, 1995).

The log fold-change calculation, logFC, is computed between

two experimental conditions. It is calculated by dividing two values, A and B, that reflect gene expression measured under two different experimental conditions, typically between mutated and control (Robinson *et al.* , 2010).

FDR is a measure used in multiple testing: it controls the number of false discoveries in tests that result in a positive result (*i.e.* significant result). Multiple testing corrections will adjust p-values derived from multiple statistical tests, in order to correct for occurrence of false positives. This is better explained resorting to an example: a p-value of 0.05 would imply that 5% of all tests result in false positives. However, when considering an adjusted p-value of 0.05, what is being considered is that that 5% of significant tests will result in false positives. Methods for multiple testing whose aim is to decrease FDR (like Benfamini-Hochberg) imply smaller adjusted p-values (Benjamini & Hochberg, 1995). This is a measure typically used in microarrays or NGS data, given the amount of tests performed in such gene-expression analysis.

2.3.1 `edgeR`

In order to perform statistical testing, `edgeR` fits the data to a negative binomial distribution, NB. NB distribution has different mean and variance. This gives more reliability than the Poisson distribution (which is by definition the model used for count data), given this distribution assumes the same value for the mean and the variance – unlike Poisson, over dispersion is accounted for within NB distribution (Robinson & Smyth, 2007). This being said, the NB distribution allows the possibility of gene-specific variability (some genes may show different biological variability from one another) and this is accounted for.

The gene counts are thus modelled as follows:

$$Y_{gij} \sim \mathbf{NB}(M_i p_{gj}; \phi_g) \tag{2.6}$$

$$\text{with } \mathbf{E}(Y_{gij}) = \mu_{ij} = M_i p_{gj} \tag{2.7}$$

$$\text{and } \mathbf{var}(Y_{gij}) = \mu_{ij}(1 + \mu_{ij}\phi_g) \tag{2.8}$$

with Y_{gij} being the number of counts for gene *g* in library *i* and replicate *j*, M_i the library size for library *i*, p_{gi} the proportion of reads for gene *g* in library *i* and ϕ_g the overdisperson parameter for gene *g*(Robinson & Smyth, 2007).

`edgeR` devises two levels of variation: biological and technical variation. The package considers the dispersion parameter, ϕ_g, as the square of the coefficient of variation, CV (ϕ_g = CV^2). CV is given, by definition, by the sum of the square of the biological coefficient of variation, BCV and the square of the technical CV, as follows (with BCV being the dominant source of variation) (McCarthy *et al.* , 2012):

$$CV^2 = BCV^2 + TechnicalCV^2 \tag{2.9}$$

The first step towards differential expression gene analysis is to model the data dispersion parameter, which determines how to model the variance for each gene. This is done by first estimating the common dispersion parameter for the genes, with `estimateCommonDisp()` function. With this consideration, all the genes are admitted to have the same value for dispersion when modelling the variance. The authors assume an extension to this tactics, given by `estimateTagwiseDisp()` function, where genes assume their own dispersion value, while also rendering it to the common dispersion estimate obtained with `estimateCommonDisp()` function. This adjustment is done with a quantile adjusted conditional maximum likelihood test, qCML, which searches for an equilibrium between common and tagwise dispersion (Robinson & Smyth, 2008).

The method these authors built introduces this weighted conditional likelihood estimator that considers tagwise dispersion when estimating the common dispersion. This shrinkage considers an approximate empirical Bayes rule, which adapts the similarity of the dispersions, considering sample sizes, scores and informations (Robinson & Smyth, 2007).

To test the difference between expression levels under two conditions, these authors use an exact test, analogous to Fisher?s exact test or the likelihood ratio test, LRT. For both, the quantile adjustment considered for qCML is used to adjust the tag counts to a common library size (Robinson & Smyth, 2007). The exact test was developed for experimental data with single factor, while LRT, which is in fact a LRT test for a Generalized Linear Model, GLM, was design mainly for experiments with multiple factor design. However, this is outside the context of this project, where we worked with single factor design experiment.

2.3.2 DESeq

As in edgeR, DESeq package also assumes data to follow a NB distribution, thus addressesing this "over dispersion" problem by modelling the number of read counts for gene *g* in sample *i* (Y_{gi}) resorting to the Negative Binomial distribution, whose parameters are determined by μ_{gi} (mean) and σ^2_{gi} (variance) (Anders & Huber, 2012).

$$Y_{gi} \sim \mathbf{NB}(\mu_{gi}; \sigma^2_{gi}) \tag{2.10}$$

The mean is expressed as the product of $q_{g,\rho(i)}$, and s_i, as follows:

$$\mu_{gi} = q_{g,\rho(i)} s_i \tag{2.11}$$

where $\rho(i)$ is the experimental condition of sample *i* and $q_{g,\rho(i)}$ is therefore a condition value for a given experimental condition on a specific sample *i* on gene *g*; $q_{g,\rho(i)}$ is proportional to the expected value of the true (but unknown) concentration of fragments from gene *g*, under condition *i*; s_i is the size factor for sample *i*. This is particularly important when estimating the library size.

The variance: σ^2_{gi}, reflects the dispersions to each gene, it is the sum of the shot noise and the raw variance.

$$\sigma^2_{gi} = \mu_{gi} + s_i^2 \upsilon_{g,\rho(i)} \tag{2.12}$$

The shot noise, μ_{gi}, is the name given to the uncertainty in measuring a concentration by counting reads. It is dominating in lowly expressed genes. The raw variance, $s_i^2 \upsilon_{g,\rho(i)}$, is the sample-to-sample variation, this term traduces the effective variance in the counts and it is dominating in highly expressed genes (Anders & Huber, 2010). The shot noise is a function of q_g and $\rho(i)$:

$$\upsilon_{g,\rho(i)} = \upsilon_\rho(q_{g,\rho(i)}) \tag{2.13}$$

When fitting the model, the first step is to define a table of counts of sequencing reads. These counts cannot be rounded, nor can they be counts of covered based pairs, DESeq is designed to work with raw counts.

Then, the package must be provided with a data.frame that stores information about the samples and their features, each row being a sample and each column being a feature about the sample, such as type of library or sample conditions. It can store size sample annotations, conditions and size factors – this is called metadata (Anders, 2010).

The model requires some parameters to be set, the first being the estimation of the effective library size (this is the normalization step) and it is imperative, because the number of reads is not necessarily a good way to assess differential expression for other than the highly expressed genes. Different samples may come from different sequencing depths, which must be put in terms of comparison. In this step, *m* size factor vectors, s_j, are estimated from the count data, resorting to the median of the ratios of observed counts along all genes:

$$\widehat{s}_i = \text{median}_g \frac{k_{gi}}{(\prod_{v=1}^{m} k_{gv})^{1/m}} \tag{2.14}$$

Then, the variance is estimated. For this, we first need to estimate, for each experimental condition ρ, *n* expression strength parameters, $q_{g,\rho(i)}$, resorting to an averaging of the counts scaled to the size vectors, according to the formula:

$$\widehat{q}_{j\rho} = \frac{1}{m_\rho} \sum_{i:\rho(i)=\rho} \frac{k_{gi}}{\widehat{s}_i} \tag{2.15}$$

where m_ρ is the number of replicates for each condition ρ and k_{gi} is the number of counts for gene *g* in sample *i*.

Afterwards, the sample variances, $w_{g\rho}$, are calculated on a common scale, for each gene *g* on condition ρ:

$$w_{g\rho} = \frac{1}{m_\rho - 1} \sum_{i:\rho(i)=\rho} \left(\frac{Y_{gi}}{\widehat{s}_i} - \widehat{q}_{g\rho} \right)^2 \tag{2.16}$$

A mean scaling factor, $z_{g\rho}$, is also defined for each gene *g* on condition ρ:

$$z_{g\rho} = \frac{\widehat{q}_{g\rho}}{m_\rho} \sum_{i:\rho(i)=\rho} \frac{1}{\widehat{s}_i} \tag{2.17}$$

By default, `DESeq` then uses a parametric fit for statistical inference on the variance, resorting to a GLM family. This regression may however lead to bad results, in which case, a local regression is fitted to the data points $w_{g\rho}$ under condition $q_{g,\rho}$, to obtain a smooth function for the estimates of the raw-variance, as follows (Anders & Huber, 2010):

$$\widehat{v}_\rho(\widehat{q}_{g\rho}) = w_\rho(\widehat{q}_{g\rho}) - z_{g\rho} \tag{2.18}$$

The test for differential expression with this package is based on the null hypothesis that the expression strength parameter for the samples of experimental condition A is the same as the expression strength parameter for the samples of experimental condition B: $\mathbf{H}_0 : q_a = q_b$ against the alternative hypothesis of $\mathbf{H}_1 : q_a \neq q_b$. To test this, the authors defined two test statistics, K_{gA}, K_{gB}, as the total counts in each condition, and their sum, K_{gS}, as:

$$K_{gA} = \sum_{i:\rho(i)=A} K_{gi}, \quad K_{gB} = \sum_{i:\rho(i)=B} K_{gi}, \quad K_{gS} = K_{gA} + K_{gB} \tag{2.19}$$

and computed the probabilities of the events $K_{gA} = a$ and $K_{gB} = b$ has $p(a,b)$. To the pair of observed counts (K_{gA}, K_{gB}) a p-value was adjusted as:

$$p_g = \frac{\sum_{\substack{a+b=k_{gS} \\ p(a,b) \leq p(k_{gA},k_{gB})}} p(a,b)}{\sum_{a+b=k_{gS}} p(a,b)} \tag{2.20}$$

This approach is similar to Robinson and Smyth and is analogous to to other conditioned tests, such as fisher's exact test (Robinson & Smyth, 2008).

2.3.3 RankProd

RankProd package was developed by Breitling *et al.* (2004). Essentially, it is a non-parametric approach on the data, that provides tools to determine the significance levels for each gene, identifying differentially expressed genes (based on the estimated percentage of false predictions, pfp – also know as the FDR) associated in the rank products's calculation, ***RP***. The method uses this rank system to classify genes among replicates and identify those that are consistently highly ranked (either strongly up regulated, or strongly down regulated) as differentially expressed (Hong, 2011).

This package presents a different perspective on the data: it assumes that the approximation of a distribution for the count data might be too much of an assumption. This can potentially not be the best way to process the data as (from the n genes in study) possibly not all of the genes can be assumed to follow the same distribution. By admitting a non parametric approach, these authors start their analysis in a non restricted way.

Breitling *et al.* (2004) assume not only that most genes are not differentially expressed but also that their variance measure is equal. Regarding the replicate arrays, the authors assume that the measurements are independent between them (Breitling *et al.* , 2004).

RankProd package considers that, under the null hypothesis, the order/rank of all genes is random and that the probability of finding a specific gene among the top r of n genes in a replicate is given by:

$$p = r/n \tag{2.21}$$

Multiplying these probabilities allows the calculation of the corresponding combined probability, as a rank product:

$$RP = \prod_j r_j/n_j \tag{2.22}$$

where r_j is the position of a specific gene in the j-th replicate and n_j is the total number of genes in the j-th replicate sorted by increasing/decreasing values (up regulated/down regulated).

High ***RP*** values reflect a bigger certainty (higher probability) that the observed position of the gene in study is where the method estimated it to be, hence, high ***RP*** values concern significant genes to the analysis (Hong *et al.* , 2006) In order to check how likely it is to observe a certain ***RP*** value, the authors propose a permutation-based procedure. This resampling method allows sample rearrangement, producing different sets of the same data, to more accurately quantify estimates and perform significance tests. In practice, one has to count how many simulated ***RP*** values are smaller or equal than the observed ***RP***, $\#(RP)$, over a large number, m, of permutations on the experimental data. From here, an estimate of the ***RP*** expected value, $\mathrm{E}(RP)$, is given by $\#(RP)/m$.

Now, it is possible to calculate an estimate of the FDR for gene g and all the genes with ***RP*** values smaller or equal than the ***RP*** of gene g, RP_g:

$$FDR_g = \frac{\mathbf{E}(RP_g)}{r(g)}$$

where $r(g)$ represents the position of gene g in the list of genes ordered by increasing value of ***RP***.

The problem of multiple testing resulting from the simultaneous analysis of thousands of genes, is automatically resolved by this procedure since ***RP*** values are converted into expected values, which allows the direct calculation of the FDR.

The main disadvantage of ***RP*** method is that there is a significant loss of performance, when the equal-variance assumption is

seriously violated and the number of replicates is higher than three (Breitling & Herzyk, 2005).

Chapter 3

Results

This chapter is meant to show the results obtained thorough the course of the project, with particular emphasis on the differentially expression phase of the analysis, exploring all the methods used and the very different results they lead to. The data in origin of the project follows the defined Y_{gi} for the number of read counts the align to gene g (g = 1, 2, 3, 4, 5, ..., 23207) in sample i (i = 1,2,3,4,5,6).

The tests performed in this chapter, to assess differential expression, were performed to a FDR of 1%, 5% and 10%, although ultimately we chose to consider a FDR of 5%, in order to achieve a reasonable number of genes.

3.1 Samples

At a genetic level, ccRCC occurs by loss of a portion of chromosome 3. This can be due to several alterations in different positions of that specific DNA portion, all of them leading to the inactivation of gene SETD2 in this type of cancer (Duns *et al.* , 2010). Duns *et al*, 2010, studied 10 ccRCC mutated cellular lines that had deletions in specific places of this chromosome portion. Their studies led to the conclusion that the mutations in four of those samples, RCC_AB, RCC_ER, RCC_FG2 and RCC_MF, will all have a strong effect on SETD2.

The authors verified (through a protein detection technique, Western Blot) that these mutations led to the inactivation of H3K36me3, as seen in figure 3.1. In this figure, six samples were tested for the presence of proteins H3K36me3 and H3K36me2. A

black spot indicates the presence of those proteins on the sample tested, with fuller spots, in principle, being synonym of a bigger amount of protein detected. The lack of a spot means that the method did not detect the protein under analysis on the sample. We can see that the four samples tested reveal absence of H3K36me3 protein.

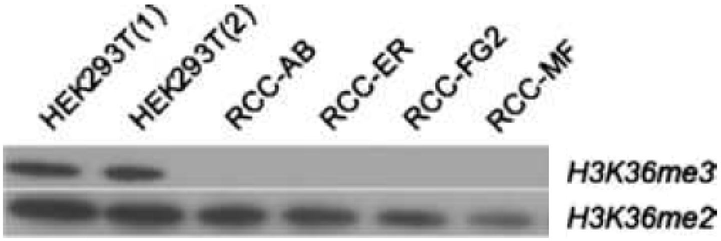

Figure 3.1: Global histone methylation levels in ccRCC mutated cellular lines. H3K36me3 and H3K36me2 levels in ccRCC cellular lines were detected by Western blot analysis on the four mutated samples and also on two HEK293T samples, which were included as positive controls for all protein tests – image adapted from (Duns *et al.* , 2010).

According to these authors results, we chose to analyse the 4 above mentioned samples. RCC_AB, RCC_ER, RCC_MF and RCC_FG2 had, respectively, 1 bp deletion, a mutation, a 9 bp deletion and a lack of 3 terminal exons of the chromosome sequence. For our analysis, two other samples served as controls (or *wt* samples): RCC1_Caki1 and RCC1_Caki2.

The 6 samples (or libraries) were analysed resorting to Illumina/Solexa technique analysis with HiSeq 2000 model, forming paired-end reads. After read quality assessment, all sequenced libraries were mapped to the human genome data base of NCBI group (hg19) using TopHat (v.2.0.3).

The mapped reads were then assembled into a table of counts, showing how many read counts are there to each gene, and 23207 genes from the entire human genome were tested. This table of counts served as input for every normalization technique applied to the data. The counts for six of those genes are represented in table 3.1 and the total counts for the raw data are represented in figure 3.4. The same six genes were represented for every normalization procedure studied in this thesis, and the total count data, normalized for each technique, was plotted in figure 3.4, to em-

phasize the differences of the several considerations taken by the normalization techniques.

	RCC_AB	RCC_Caki1	RCC_Caki2	RCC_ER	RCC_FG2	RCC_MF
WASH7P	2546	1431	2682	2083	2063	3274
NOC2L	10101	20421	33962	18615	16438	6718
SLC2A10	76	7504	6716	1	22	2
LOC729737	316	536	303	552	1915	1073
AMIGO2	7107	43401	71248	5196	11981	5469
SETD2	4419	8916	10555	1533	1138	6055

Table 3.1: Raw counts for the 6 ccRCC cellular lines.

3.2 Normalization

3.2.1 edgeR

This package also receives as input the raw count matrix, which serves to create edgeR object, DGEList. This object is part of the DGEList-class and stores the read counts and a data.frame. And this data.frame has the library sizes and the experimental conditions for each sample.

After creating this object, the first step was to apply a filter to the analysis. The criteria applied here is to filter out lowly expressed genes, the interest being in keeping tags that are expressed in at least one of the two experimental conditions in study (mutated or *wt* samples). As proposed by the authors, genes having less then one count per million on either of the groups were removed. Thus, the filtered data is composed by 14247 genes out of the initial 23207, reducing it to about 61%.

After straining the data, the analysis proceeded with the estimation of the size factors, applying calcNormFactors() function. This function aims at determining normalization factors to scale the raw library sizes. The obtained normalized factors, as well as the library sizes and experimental groups, are stored on the DGEList object and are represented in table 3.2.

Note that all library sizes seem equally sized and that the norm.factors column also appears to give relatively similar sized factors. RCC_AB and RCC_FG2 present the highest normalization factor, which means that these should be the smallest libraries from the bunch. Likewise, RCC_MF has the lowest normalization

	group	lib.size	norm.factors
RCC_AB	mutated	157228170	1.1331755
RCC_Caki1	wt	153131637	0.9470461
RCC_Caki2	wt	159440222	1.0287259
RCC_ER	mutated	121392837	0.9439515
RCC_FG2	mutated	142518192	1.0601597
RCC_MF	mutated	118509054	0.9051302

Table 3.2: Size factors for `edgeR`

factor, implying that this is the sample with the highest number of reads aligned to.

Like in `EDASeq`, the normalized count data obtained resorting to this package is stored on a matrix to be used for DE analysis. The counts for six of the genes analysed are represented in table 3.3 and the distribution of the full normalized count data for `edgeR`, is represented in figure 3.4.

	RCC_AB	RCC_Caki1	RCC_Caki2	RCC_ER	RCC_FG2	RCC_MF
WASH7P	2015.36	1391.67	2306.22	2563.81	1925.69	4304.62
NOC2L	7995.89	19859.83	29203.21	22911.44	15344.12	8833.20
SLC2A10	60.53	7297.79	5774.92	1.35	20.53	2.80
LOC729737	250.02	521.28	260.49	679.50	1787.61	1410.73
AMIGO2	5625.85	42208.34	61264.64	6395.35	11183.74	7190.89
SETD2	3498.07	8670.99	9076.01	1886.94	1062.23	7960.96

Table 3.3: Counts normalized for `edgeR`

With `edgeR`, both the `lib.size` and the `norm.factors` are multiplied together and act as effective library size factor.

3.2.2 DESeq

The input to this package is the matrix of read count data obtained with NGS for gene g in every sample i.

After reading the count data matrix, a `data.frame` was defined for the samples, specifying their experimental condition and the type of read they were assigned to (all of which paired-end read).

We then provided this `data.frame` to `DESeq` `newCountDataSet()` function, from the `CountDataSet` class. This function creates a `CountDataSet` object which stores information regarding the aligned read counts and the samples in study, as defined by the `data.frame`. This object is not yet fully defined, as it still lacks information about the size factors and the dispersions, the first being the key

step at this point of the analysis (normalization), the latter to be determined later, on the DE analysis (section 3.3.2).

The function `estimateSizeFactors()` **was used for the normalization step, to estimate the effective library size based on each sample count data. The function estimates the size factors, storing them to the CountDataSet object, as obtained by table 3.4.**

	sizeFactor	condition
RCC_AB	1.32	mutated
RCC_Caki1	1.02	unmutated
RCC_Caki2	1.14	unmutated
RCC_ER	0.85	mutated
RCC_FG2	1.11	mutated
RCC_MF	0.77	mutated

Table 3.4: Size factors for `DESeq`

Two things must be noted here. The first is that the values for the size factors of each library are not very discrepant (they all revolve around the value one). The second is that samples from RCC_ER and RCC_MF (both mutated samples) are the two samples that have, in comparison, the higher number of counts, given that their size factor values are the lowest, the table also indicates that RCC_AB has the highest size factor, indicating that this sample has less counts than the others.

Like in the two previous methods, the normalized counts obtained for `DESeq` **are kept in a matrix that will serve as input for DE analysis that. Like** `EDASeq`**, DE analysis will proceed with** `DESeq` **(section 3.3.2), with** `edgeR` **(section 3.3.1) and with** `RankProd` **(section 3.3.3). Table 3.5 represents the normalized counts of the same six out of the 23207 genes (notice that, unlike the previous methods – that filter the data prior to estimating normalization factors –** `DESeq` **does not perform this step here, it performs it after estimating variance, in section 3.3.2). The full effect of** `DESeq` **normalization procedure on the data, compared to raw counts, is represented by the first line of figure 3.4.**

3.2.3 EDASeq

EDASeq package is a two step normalization procedure: it first normalizes within lane (making the count data within every library comparable to one another) and then between lane (allowing the data between the different lanes to be comparable).

	RCC_AB	RCC_Caki1	RCC_Caki2	RCC_ER	RCC_FG2	RCC_MF
WASH7P	1924.17	1407.73	2342.55	2441.41	1851.64	4236.08
NOC2L	7633.96	20088.87	29663.56	21817.99	14753.86	8692.12
SLC2A10	57.44	7381.95	5865.98	1.17	19.75	2.59
LOC729737	238.82	527.28	264.65	646.98	1718.80	1388.31
AMIGO2	5371.20	42695.11	62230.42	6090.05	10753.50	7076.09
SETD2	3339.71	8770.99	9219.10	1796.78	1021.41	7834.29

Table 3.5: Counts normalized for `DESeq`

The analysis with this package starts with a file containing the gene level counts for the six samples. The first thing done was to apply a filter to this raw data. The criteria used, as advised by the authors, was that genes that had a total amount of ten average read counts in the six samples are excluded: this made our data go from 23207 to 17293 genes, about 75%. After this, a `data.frame` was created, storing gene level features information – gene length and GC content – for every gene in study. The `data.frame` served as input to define the `newSeqExpressionSet()` function of `EDASeq` `SeqExpressionSet()` class. This function stores not only read counts, but also feature data for every gene and information about the samples.

EDASeq can normalize the data according to both stored features: GC content or gene length. Our analysis proceeded with normalization for GC content, given this is the most used and most described approach in papers. The results were plotted in figure 3.2. The figure represents, on the first column, the loess regression of the counts on GC content. Column two has the same data on the axis, but considering within lane normalized counts for GC content, with function `withinLaneNormalization()`. Column three accounts for the between lane normalization, achieved with `betweenLaneNormalization()`. From this picture, it seems clear that GC content is a good input for this package normalization in our count data, leading to a stable variation of it – as seen in the upper panel three of figure 3.2.

With these authors approach, when normalizing within lane for the GC content – second panel in figure 3.2 – the effect this factor introduces to the count data is removed. When normalizing between lane, the effect on sequencing depth is also removed. As a result, these authors made available a method with a more specific normalization process, leading to normalized data, as seen in panel three from figure 3.2.

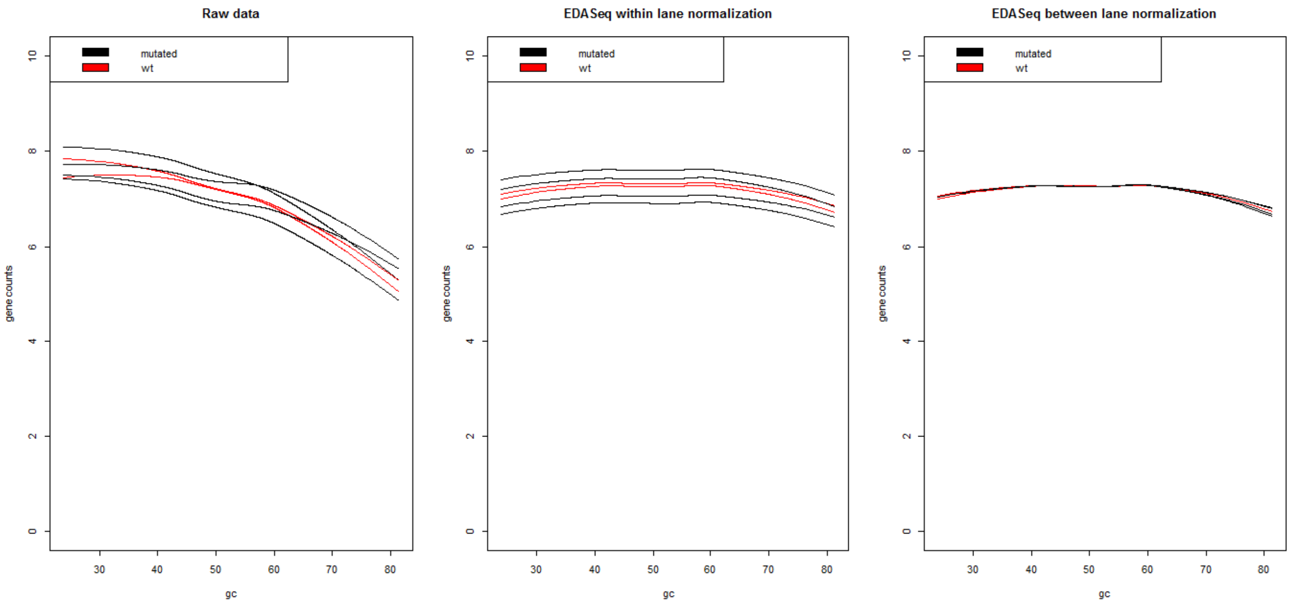

Figure 3.2: Biasplots for `EDASeq` normalization. The first line represents the three panels for data normalization for GC content, the second line represents the data normalized for gene length. The first column plots raw counts regarding the variable in study (GC content and gene length), the second and third columns stand for within and between lane normalization, for each factor.

An extra parameter in the `newSeqExpressionSet()` function is the `offset`, which is a matrix with the same size as the count data. It is used to store a normalization offset to be supplied to the model in differential expression analysis. This parameter allows this packages users to keep the count data unchanged, saving an offset argument to the data, which, if not specified, will constitute a matrix of zeros. A visual representation of the `offset` parameter is provided with figure 3.3. The figure plots the GC content data, followed by its between lane normalization and then between lane normalization with an offset. The first and third panel appear to be the same, although they are produced with different criteria.

The normalized counts obtained with `EDASeq` are stored in a matrix to be used for differential expression analysis by `DESeq` (section 3.3.2), by `edgeR` (section 3.3.1) and by `RankProd` (section 3.3.3). Table 3.6 represents the normalized counts of six of the 17293 genes. The full effect of `EDASeq` normalization procedure on the data, compared to raw counts, is represented by the first column of figure

3.4.

	RCC_AB	RCC_Caki1	RCC_Caki2	RCC_ER	RCC_FG2	RCC_MF
WASH7P	2920	2058	3068	3013	2172	5058
NOC2L	11155	21230	33032	23735	18058	9417
SLC2A10	52	7100	5741	2	14	6
LOC729737	447	930	646	1098	1993	2201
AMIGO2	5432	31616	56718	6624	11100	6466
SETD2	2954	8167	8354	1648	698	7263

Table 3.6: Counts normalized for `EDASeq`.

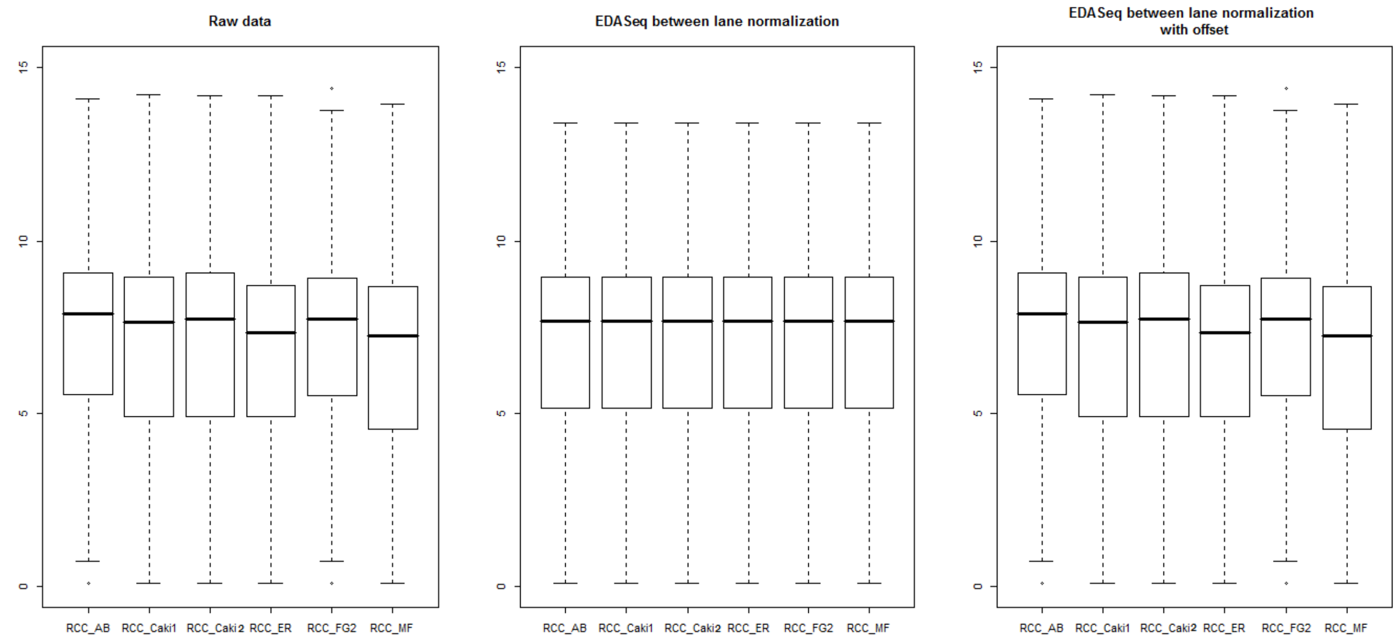

Figure 3.3: The effect of `offset`. Panel one represents the raw data for GC content; panel two represents the between lane normalization for GC content; panel three stands for the between lane normalization, with the offset to GC content.

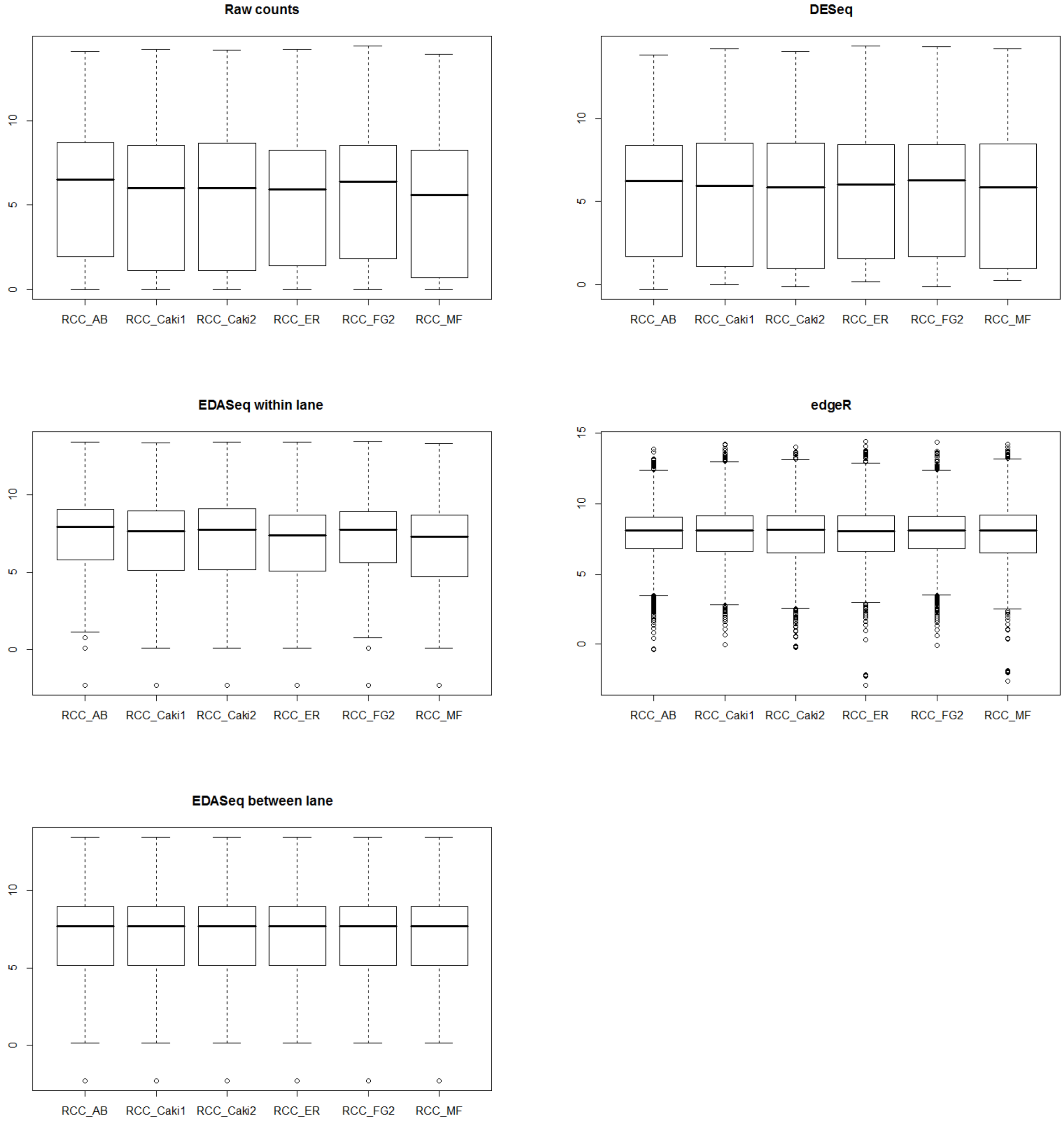

Figure 3.4: Boxplots for normalized data counts on log scale. The picture represents the effects of the several normalization procedures on the data: the first column corresponds to the raw data, followed by `EDASeq` within lane and between lane normalization; the second column shows data normalized for `DESeq`, followed by `edgeR`.

3.3 Differential Expression Analysis

3.3.1 `edgeR`

`edgeR` with `edgeR` normalized matrix

With the fully defined `DGEList` object in section 3.2.1, and before proceeding to DE analysis, we did a Multidimensional scaling plot of the gene expression profiles, with the `plotMDS()` function, represented in figure 3.5. This function calculates (internally) the distance between each pair of samples as the root of the mean square deviation (Euclidean distances), and plots them. It provides an interesting way to perceive the data on a 2D scale, showing the sample relations in a multidimensional scaling mode.

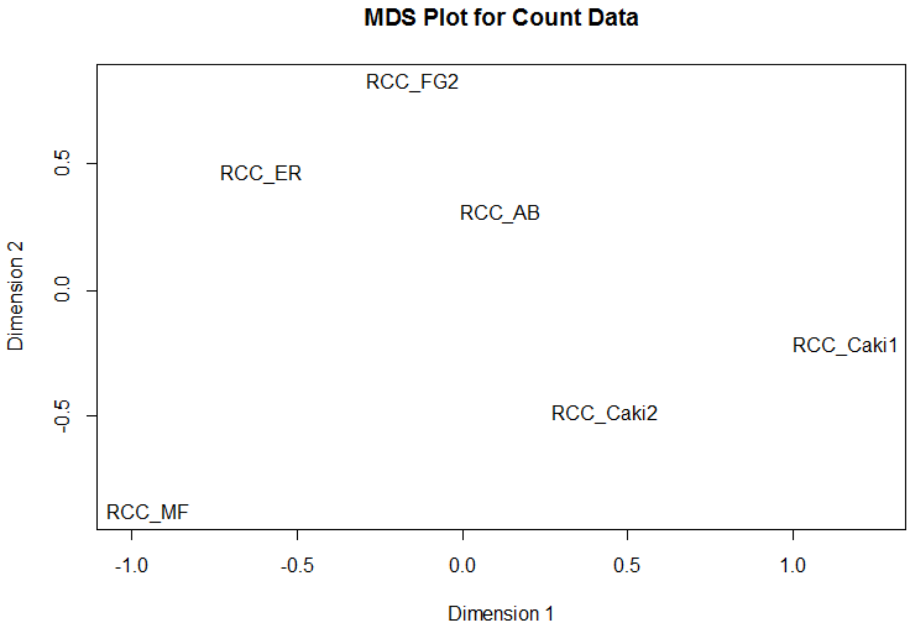

Figure 3.5: Multidimensional scaling plot for the data –normalization with `edgeR` and DE analysis with `edgeR` – showing the relations between the samples in two dimensions, with dimension 1 separating mutated and *wt* samples.

We can see from this figure that dimension 1 separates the mutated from the *wt* samples. This evidences that the replicates from each condition are reasonably similar to each other, which should lead to differences in DE analysis.

We carried through the analysis by estimating the dispersion. This was done by first estimating the common variance parameter for the reads, with estimateCommonDisp() function. This function receives as input the DGEList and returns a value for the dispersion, considering all genes. The value obtained was 0.5260142, with which we get to the coefficient of biological variation of 0.7252683. This value means that the true abundance for each gene can vary up or down by 73% between replicates, which points to a high variation.

The analysis progressed with the estimateTagwiseDisp(), which estimates an individual dispersion parameter for each gene, while moderating it to the common dispersion parameter. This way, estimates that are bigger then the common value are made smaller and *vice versa.* This offers an improved statistical inference by sharing information, which is important because the common dispersion parameter determines the same unique value for the variance for all genes – and that can be considered a rather risky statement. The way we set this squeezing effect was by defining the prior.df parameter in the estimateTagwiseDisp() function. In this analysis, we defined it to be 4 degrees of freedom (given that we were working with 6 samples for two groups, $6 - 2 = 4$), this coerces edgeR to restraint the tagwise dispersion to the common value.

A plotBCV() was designed in figure 3.6, representing both the biological coefficient variation (obtained with the common dispersion) and the estimated dispersions (obtained with tagwise dispersion):

This led to the final stage of the analysis, the differential expression, which is achieved with the exactTest() and the topTags() functions. The first receives as input the DGEList and a vector, pair, which names the group comparison to be done. The latter received as input the exactTest() statistics, and adjusted it with the Benjamini-Hochberg procedure to control FDR. The top six genes the method reported as differentially expressed, are given in table 3.7.

In this table, tests for each genes were done, based on the null hypothesis that is defined in exactTest() (and that is based on the qCML method): evaluating the null hypothesis, that there is no difference between mutated and *wt* samples (non differentially expressed genes) *versus* the alternative hypothesis that there is. The table reports logFC values, logCPM values (which considers the log of the counts per million obtained for that gene), the PValue

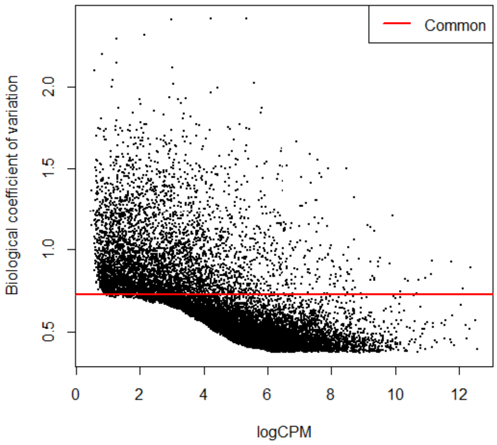

Figure 3.6: Normalization with `edgeR` and DE analysis with `edgeR` – plot for the dispersion estimates obtained with tagwise dispersion (black dots) and common dispersion, 0.5260142, (represented with the red line).

Comparison of groups: mutated-wt				
	logFC	logCPM	PValue	FDR
MLLT11	-3,960785	5,734351	1,04557E-13	1,49047E-09
PLAC8	-7,346652	4,589403	2,08922E-11	1,48909E-07
C8orf4	-7,796926	1,722237	1,11116E-10	4,72846E-07
COL5A1	7,762157	5,694833	1,32682E-10	4,72846E-07
ACOT7	-3,067926	5,897775	6,0447E-10	1,72334E-06
AFAP1-AS1	-9,447595	5,712523	9,62531E-10	1,9709E-06

Table 3.7: Normalization with `edgeR` and DE analysis with `edgeR` – top 6 identified genes.

and the `FDR` (which refers to the adjusted *p-value*) obtained for this tests.

A final function was added to the analysis, `decideTestsDGE()`. This function received as input the `DGEList` results from the `exactTest()` function (again, adjusting the Benjamini-Hochberg procedure to control FDR, by settling it to 5%). This analysis identified 307 genes as differentially expressed, 183 up regulated and the remaining 124 down regulated (these results can be visualized resorting to the `plotSmear()` in figure 3.7):

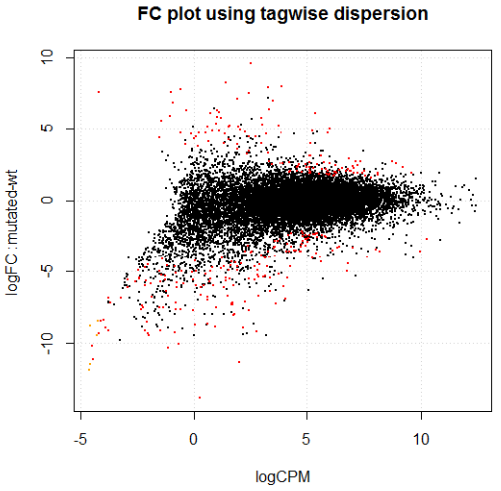

Figure 3.7: Normalization with `edgeR` and DE analysis with `edgeR` – plot of all 307 genes identified as DE (red dots).

`edgeR` with `EDASeq` normalized matrix

The `EDASeq` package estimates correction factors that adjust for sample specific GC content effects. These are compatible with `edgeR`, when passed to this package with the original counts and an offset (to be supplied to a generalized linear model).

Thus, the original counts and the offset obtained with `EDASeq` in section 3.2.3 was used to create an `edgeR` `DGEList` object, which served as input for the `plotMDS()` function, plotting the relative similarities of the six samples in study. This is represented by figure 3.8.

From this figure, we can see that dimension 1 seems to separate the mutated from the *wt* samples into two different regions, which allows us to infer that the replicates are analogous within their experimental group. This should translate in differences in DE analysis.

The following steps for this analysis are in essence very similar to the previous topic, except that for this joint analysis, a design matrix is defined, in order to create a generalized linear model, which was done resorting to the `model.matrix` function. This function creates a design matrix for the given factors (in this case, a vector where the value 1 was given to the mutated samples, and the value 2 was given to the *wt* samples), using the original data,

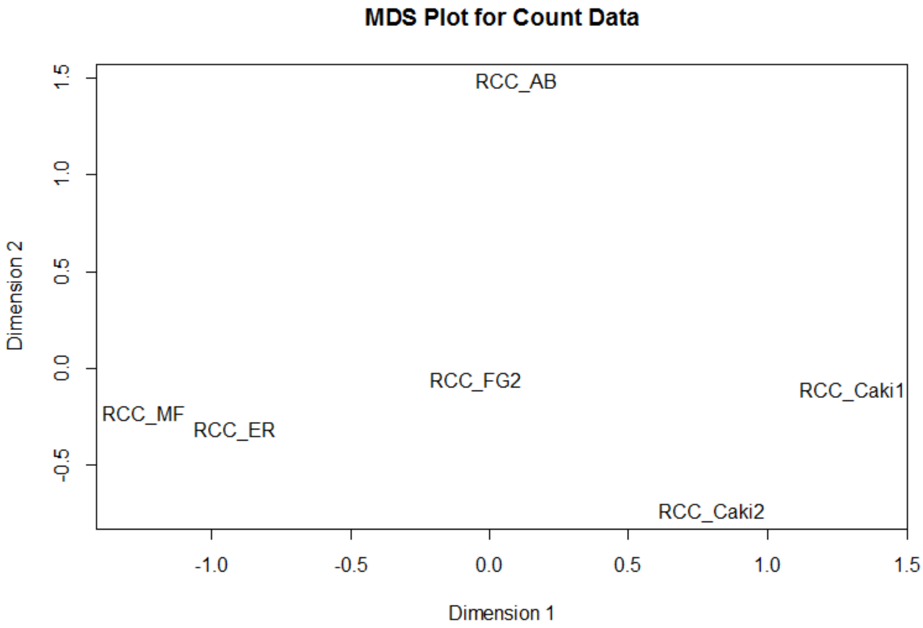

Figure 3.8: Multidimensional scaling plot for the data –normalization with `EDASeq` and DE analysis with `edgeR` – showing the relations between the samples in two dimensions, with dimension 1 separating mutated and *wt* samples.

as defined by `EDASeq` with an offset.

Next, the dispersion was estimated, but using the appropriate functions to a GLM model, the `estimateGLMCommonDisp()` function and the `estimateGLM` `TagwiseDisp()` function. These were applied under the same premises as before, but they are supplied not only with the `DGEList` object, but also with the `design` matrix. The obtained value for the `estimateGLMCommonDisp()` was 0.6485521, meaning that the coefficient of biological variation value was 0.8053273. This points to a high variation between replicates.

The `estimateGLMTagwiseDisp()` was enforced to the data and then a `plotBCV()` function was created, to plot how the two estimate dispersions functions are associated, shown in figure 3.9.

Afterwords, we assessed for differentially expression, resorting first to functions `glmFit()` (in which the offset is supplied) and `glmLRT()`. These functions fit a negative binomial model, implemented through generalized linear models methods, followed by

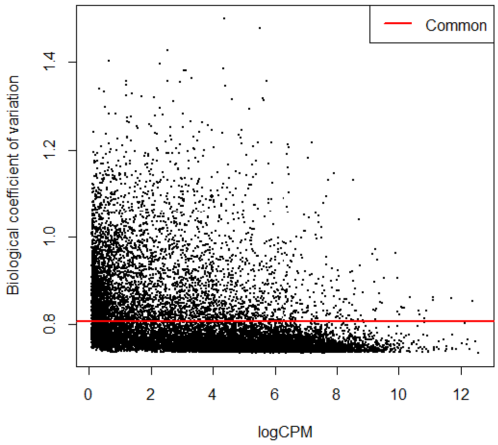

Figure 3.9: Normalization with `edgeR` and DE analysis with `edgeR` – plot for the dispersion estimates obtained with tagwise dispersion (black dots) and common dispersion, 0.6485521, (represented with the red line).

likelihood ratio tests for the coefficients in the linear model.

Finally, the `topTags()` function was applied to the data: it receives as input the `glmLRT()` statistics and adjusts the test p-values resorting to Benjamini-Hochberg procedure. Table 3.8 shows the top six genes this joint methodology returned as differentially expressed:

Coefficient: groups: mutated-wt					
	logFC	logCPM	LR	PValue	FDR
GPR173	10,681581	29,03964	58,64665	1,88689E-14	3,262998e-10
ISL2	9,759915	26,46001	53,61301	2,44138E-13	1,459297e-09
FAM25A	10,620449	25,57775	53,36073	2,77591E-13	1,459297e-09
EREG	10,271295	34,14963	52,97660	3,37546E-13	1,459297e-09
PLAC8	8,289833	31,61084	51,32342	7,83356E-13	2,709314e-09
CST4	11,769234	26,81185	48,61243	3,11886E-12	8,989077e-09

Table 3.8: Normalization with `EDASeq` and DE analysis with `edgeR` – top 6 identified genes

This method also used `decideTestsDGE()` function and `plotSmear()` to determine and better visualize the results obtained with this methodology: the analysis determined 866 genes as differentially expressed, 464 being up regulated, 402 down regulated.

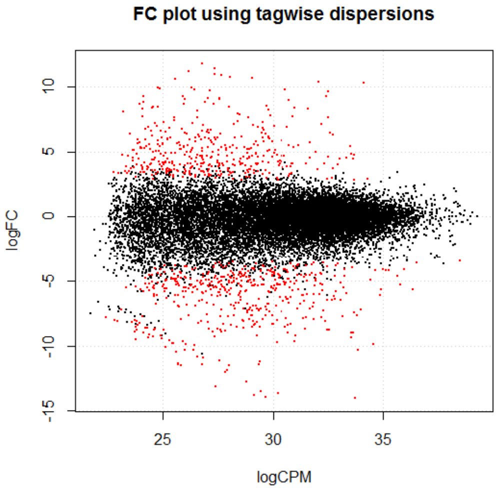

Figure 3.10: Normalization with `edgeR` and DE analysis with `edgeR` – plot of all 866 genes identified as DE (red dots).

`edgeR` with `DESeq` normalized matrix

A final approach was taken regarding `edgeR` differential expression achievements. To that extend, in this part of the analysis, we provided `edgeR` the normalized count data obtained from `DESeq` normalization as in section 3.2.2.

The same graphical representation of Euclidean distances was determined and plotted in figure 3.11.

And, once again, the samples were devised based on their experimental group, leading to a good prediction for the differential expression analysis.

The next step was to determine variance dispersion, `estimateCommonDisp()` led to 0.6659756, which translates into a coefficient of biological variation of 0.8160733. The `estimateTagwiseDisp()` function was applied to the data, for the same `prior.df` value of 4 degrees of freedom (6 samples − 2 experimental groups = 4) and the `plotBCV()` was designed, representing with a red line the common dispersion value statistics obtained for this analysis and, with the black dots, the estimates achieved with tagwise dispersion (figure 3.12).

This led to final stage of the analysis, identifying differentially expressed genes. This was achieved with the `exactTest()` and `topTags()` functions, with the same inputs as previously described

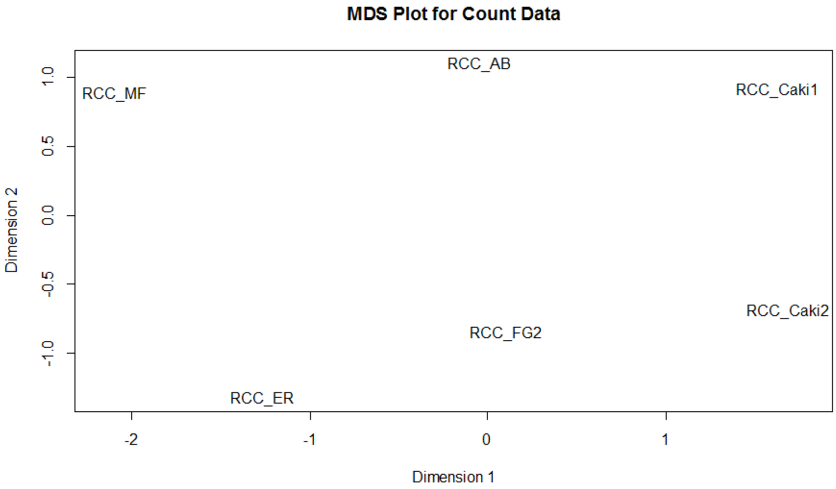

Figure 3.11: Multidimensional scaling plot for the data –normalization with`DESeq` and DE analysis with `edgeR` – showing the relations between the samples in two dimensions, with dimension 1 separating mutated and *wt* samples.

in section 3.3.1, under "`edgeR` with `edgeR` normalized matrix". The top six genes this joint methodology identified as differentially expressed are represented in table 3.9.

Comparison of groups: mutated-wt				
	logFC	logCPM	PValue	FDR
MLLT11	-3,96587	5,717642	1,92E-12	4,45E-08
C8orf4	-7,74723	1,665067	1,01E-10	1,17E-06
PLAC8	-7,50675	4,530177	2,49E-10	1,92E-06
COL5A1	7,721449	5,740842	5,15E-10	2,36E-06
INSR	7,340207	4,698672	5,49E-10	2,36E-06
RASSF6	5,315024	5,179019	6,09E-10	2,36E-06

Table 3.9: Normalization with`DESeq` and DE analysis with `edgeR` – top 6 identified genes

The `decideTestsDGE()` function revealed 254 significant genes for a FDR of 5%: 136 were up regulated, 118 were down regulated. A graphical representation of this results is given by figure 3.13. We can observe that the plot returns orange dots on the yy axis. These dots represent the genes whose counts were zero in all samples of one of the groups (mutated or *wt*).

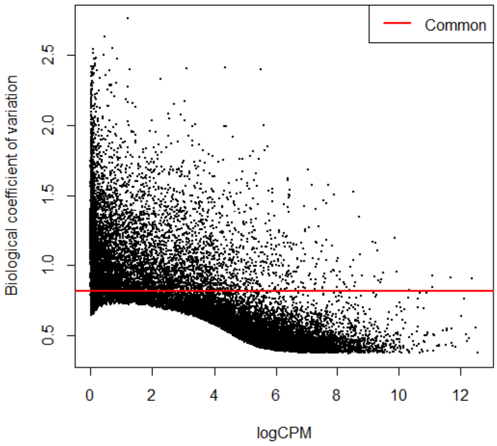

Figure 3.12: Normalization with `edgeR` and DE analysis with `edgeR` – plot for the dispersion estimates obtained with tagwise dispersion (black dots) and common dispersion, 0.6659756, (represented with the red line).

3.3.2 DESeq

DESeq with`DESeq` normalized matrix

To complete `CountDataSet` object, after having estimated the size factors vector (section 3.2.2), the analysis proceeded with the variance estimation, by use of the `estimateDispersions()` function. With the results achieved with this function, a graphic of empirical dispersion values and local regression dispersion values was plotted against the mean of normalized counts, obtained as from the `estimateSizeFactors()` function, resorting to the `plotDispEsts()` function (figure 3.14).

The variance estimation, achieved with the `estimateDispersions()` function, depends on estimating a dispersion value for each gene, fitting a local regression curve through the estimates and then assigning a dispersion value to each gene.

DESeq most recent version uses a parametric fit for the regression. This way, the regression line models not only the underlying dispersions, but also the true underlying variance between different genes, which is a more conservative approach. Upon estimating the dispersion for each gene, what`DESeq` does is attribute a value to the per gene estimation (every black dot on figure 3.14). Should this point estimate be below the parametric regression line, the

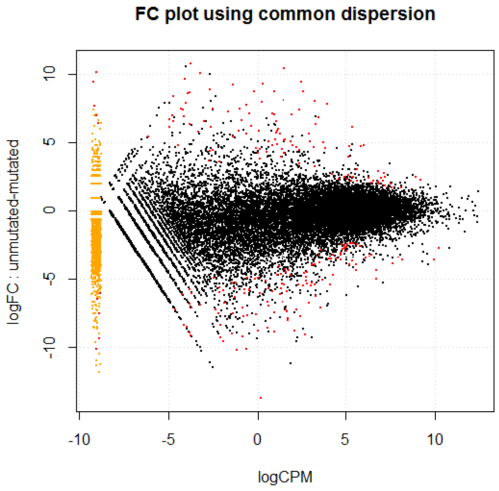

Figure 3.13: Normalization with `edgeR` and DE analysis with `edgeR` – plot of all 254 genes identified as DE (red dots).

method assigns to this point the estimate obtained from the regression line. If the point estimate lies above the regression line, the method assumes this value as it is.

However, this parametric fit gave very poor results so, as advised by the package authors, the analysis proceeded with a local fit (Anders & Huber, 2010). With a local fit, the assumption is that the red regression line in figure 3.14 stands for the true underlying dispersions, and that the variation of the point estimates around it reflects the sampling variance.

Like mentioned in section 3.2.2, Anders and Huber recommend that the filtering step should be done after estimating the variance. They hypothesize that, by filtering the data at this stage, the raw p-values should be the same as without filtering, but the adjusted p values might get better (Anders & Huber, 2012).

In order to do this, a `data.frame` with a new variable was created, `filterstat`, which is defined as the average number of reads for each gene across all samples. The `data.frame` also has a `p-value` associated with testing the null hypothesis of the equality of the mean counts in both experimental conditions, and the `row.names` from the genes in study. With this `data.frame` a scatterplot was devised, plotting `filterstat` *versus* the log of the `p-value`. This is represented in figure 3.15A.

From figure 3.15A we can point that it seems that 40% of the

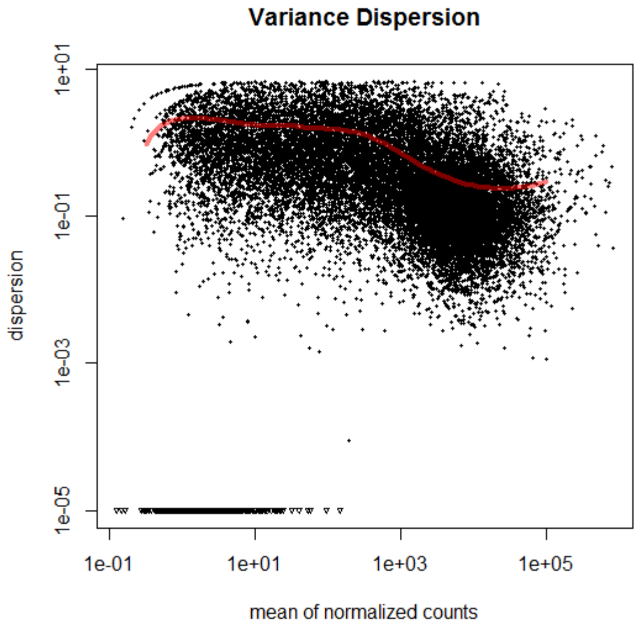

Figure 3.14: Plot for empirical dispersions forDESeq normalization andDESeq differential expression analysis. The empirical dispersions values for EDASeq normalization andDESeq differential expression were plotted with a local fit no better adjust to the data.

genes revolve around 2.5 in the log scale, which translates in a p-value around 0.003 ($10^{-2.5}$). This means that in 40% of the genes with lower counts (with exception of a few points that do not seem to follow this trend) almost none seem to achieve that p-value. The authors propose those low count genes to be excluded from the analysis, which reduces the data to 13924 genes (scatterplot presented in figure 3.15B).

The function `nbinomTest()` was then used to test the differences between the base means of the mutated and *wt* samples. Table 3.10 shows the results for the first six tested genes, which are organized in a `data.frame`. The `baseMean` stands for the mean normalized counts, `baseMeanA` and `baseMeanB` stand for the mean of normalized counts for the mutated and *wt* samples, respectively, the `log`$_2$`FoldChange` is the logarithm of the `foldChange` which is calculated from the mutated to the *wt* samples.

The null hypothesis tested is $q_{iA} = q_{iB}$, which raises the question if the expression strength parameters from gene *i* on sample

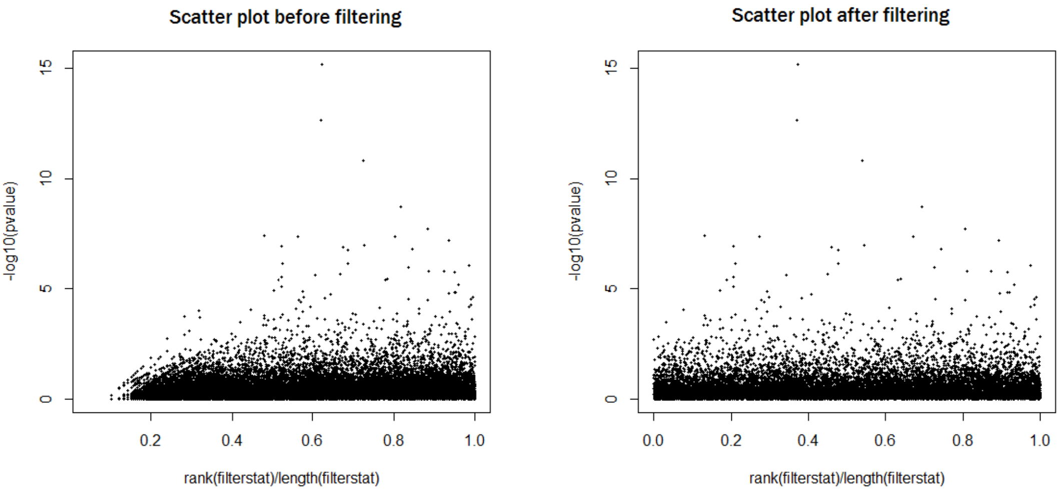

Figure 3.15: Normalization with`DESeq` and DE analysis with`DESeq` – the figure represents a scatterplot of the raw data (A) and the filtered data (B).

id	baseMean	baseMeanA	baseMeanB	FC	log2FC	pval	padj
SLC2A10	2221,48	20,23593	6623,967	327,3369	8,354632	6,41E-16	1,35E-11
COL14A1	2160,478	43,12196	6395,189	148,3047	7,21242	2,16E-13	2,28E-09
GLUL	5017,09	7510,402	30,46438	0,004056	-7,94562	1,53E-11	1,08E-07
MLLT11	7568,358	1213,797	20277,48	16,70582	4,062279	1,77E-09	9,35E-06
LOC100422737	13563,7	20275,1	140,9121	0,00695	-7,16877	1,81E-08	7,62E-05
COL5A1	6994,693	10465,58	52,92484	0,005057	-7,62749	4,06E-08	0,000107

Table 3.10: Normalization with`DESeq` and DE analysis with`DESeq` – top 6 identified genes

A **are the same for sample *B*. This translates in assessing, in the test statistics, the counts for these samples on each gene. The null hypothesis corresponds to the non differentially expressed genes and the alternative hypothesis is that the counts obtained for the mutated and the *wt* samples are different, corresponding to the differentially expressed genes. With R, this is evaluated considering the fold-change obtained from the mutated to the wild type samples: the `pval` is the *p-value* for the statistical significance of this change and the `padj` is the adjusted *p-value* for multiple testing, using the Benjamini-Hochberg procedure, which controls FDR (Anders & Huber, 2012).**

The results for these multiple tests are represented in figure 3.16, with the differentially expressed genes at a FDR of 0.05 rep-

resented in red. This figure is a `plotMA`, which, by definition, plots the log of an intensity ratio (M-values) against averages (A-values). Figure 3.16 represents the log$_2$`FoldChange` against the mean of normalized counts, `baseMean`. The figure has a clear tilt in the negative *yy* axis, which is due to the fact that there are only two *wt* samples versus four mutated samples.

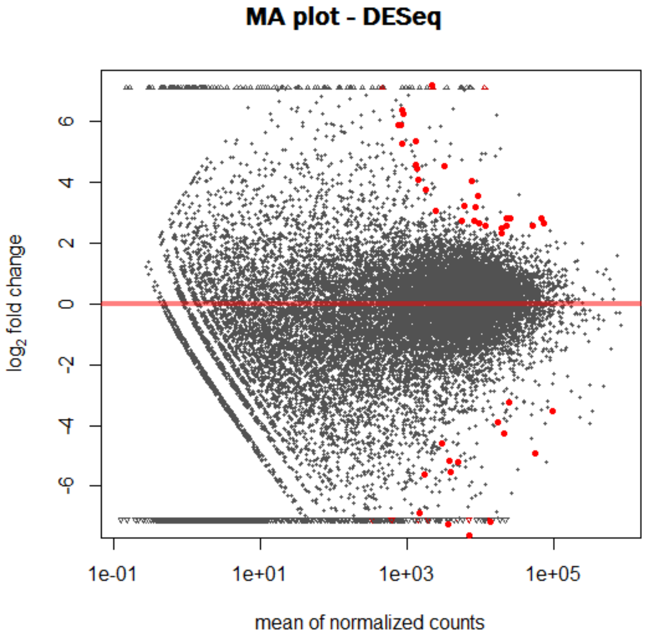

Figure 3.16: Normalization with`DESeq` and DEG analysis with`DESeq` – plot MA. This plot represents the mean of the normalized counts against the log_2 FC, representing the differentially expressed genes in red.

From this plot (figure 3.16) a few things can be noted: we can see that the point estimates are relatively uniformly distributed and appear to show a symmetry around the *yy* axis - marked with the red line - which is an indicator that the differential expression testing gave reliable results. We can also observe that this methodology seems to identify more up regulated genes (genes with a positive log FC) then down regulated genes.

This analysis, with normalized data with`DESeq` and differential expression analysis with`DESeq` identified 53 genes at a 5% FDR, 34 up regulated and 19 down regulated.

DESeq with EDASeq normalized matrix

The matrix of normalized counts obtained with EDASeq in section 3.2.3 serves as input to determine differential expression withDESeq. The first step regarding this analysis was to define the normalized count matrix as aDESeq object, to make it accessible to this package. This was done resorting to the as function (this function receives as input the count matrix and forces it into the CountDataSet class ofDESeq).

However, this CountDataSet is defined differently from the previous section: since the data it receives is already normalized for between-lane, it defines the sizeFactor vector as a vector composed by six values of one, as observed by the following table 3.11.

	sizeFactor	condition
RCC_AB	1	mutated
RCC_Caki1	1	unmutated
RCC_Caki2	1	unmutated
RCC_ER	1	mutated
RCC_FG2	1	mutated
RCC_MF	1	mutated

Table 3.11: Effect on size factors forDESeq when normalized with EDASeq

The analysis proceeded with the estimateDispersions() function, to which, as in the previous section, a local regression was applied, and the values obtained were represented with plotDispEsts() in figure 3.17.

The next step in the analysis was to assess for differentially expressed genes, by applying the nBinomTest() to the data.

id	baseMean	baseMeanA	baseMeanB	FC	log2FC	pval	padj
EREG	19057,83	57082,5	45,5	0,000797	-10,293	1,20E-27	2,08E-23
CYTL1	4602,333	13787	10	0,000725	-10,4291	8,16E-25	7,05E-21
COLEC10	6099	18249,5	23,75	0,001301	-9,58571	2,16E-23	1,24E-19
AFAP1-AS1	5819,833	17402,5	28,5	0,001638	-9,25412	2,60E-22	1,12E-18
GSTA1	14222,67	1,5	21333,25	14222,17	13,79585	2,56E-19	8,86E-16
KRT79	1617,833	4843	5,25	0,001084	-9,84937	2,27E-18	6,53E-15

Table 3.12: Normalization with EDASeq and DE analysis withDESeq – top 6 identified genes

The joint analysis with data normalized with EDASeq and differential analysis withDESeq led to the identification of a very considerable number of genes, when compared with the full analysis

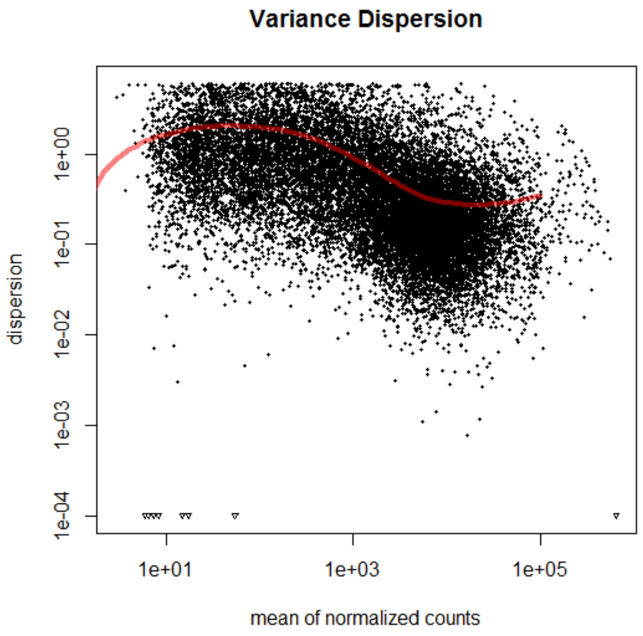

Figure 3.17: Plot for empirical dispersions for `EDASeq` normalization and`DESeq` differential expression analysis. The empirical dispersions values for `EDASeq` normalization and`DESeq` differential expression were plotted with a local fit no better adjust to the data.

with`DESeq`. It identified, at a 5% FDR, 542 genes as differentially expressed, 204 up regulated, 338 down regulated.

3.3.3 RankProd

Even though `RankProd` premises do not link directly to the technique in study (NGS) – as they are applied in microarrays – this method is designed to operate meta-analysis and its DE analysis assessment starts, as in Hong 2011, by being supplied with a normalized matrix of the gene expression data to be analysed (Hong, 2011). Thus far, to our knowledge, there are no studies pointing at discovering `RankProd` applicability to NGS data but, by assuming our data's six lanes as arrays, we proposed to study whether it applies to this RNA-Seq data set.

RankProd with EDASeq normalized matrix

The first step in the analysis was to use `RP()` function. This function applies the rank product method to identify differentially expressed genes. It takes as input a count data matrix, a vector with the class labels of the samples and, an extra parameter stating the number of permutations. The first argument was provided from the normalized counts obtained with `EDASeq` in section 3.2.3; the second is defined by the number of columns, with labels 0 and 1, for *wt* and mutated samples respectively; the third was set to $m = 100$ permutations.

The results obtained with this function served as input for the `plotRP()` function, which plots the estimated pfp values (or FDR) *versus* the number of identified genes, to a certain cutoff (in this analysis considered to be 0.05), as in figure 3.18.

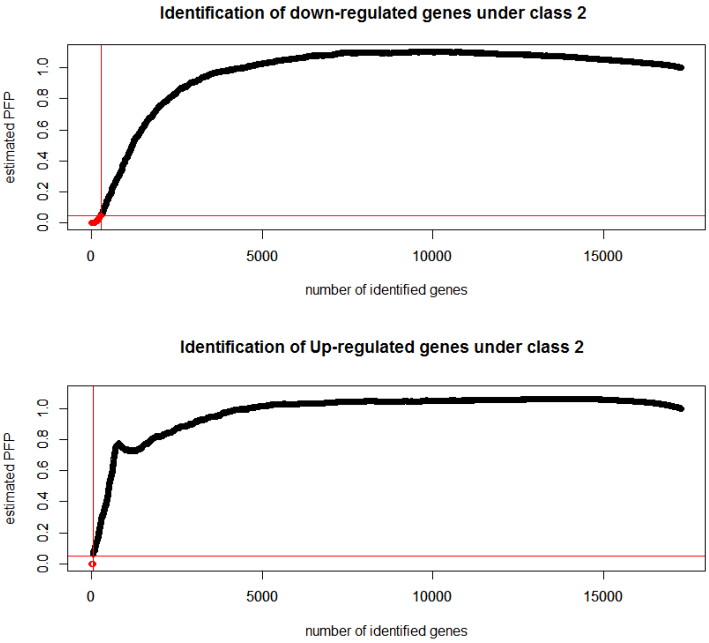

Figure 3.18: Normalization with `EDASeq` and DE analysis with `RankProd` – FDR representation for genes ordered by RP, red dots corresponding to DEG.

The following step in the analysis was to apply `topGene()` function to the data, to obtain the FDR values associated to every test. The top six genes identified by this methodology as differentially

expressed (340 genes for a FDR of 0.05) are represented in table 3.13 (with `FC` standing for fold-change and `regulation` designating the regulation of the gene).

id	RP	FC	FDR	p.val	regulation
CCRL2	134,1432	0,0009	0,0000	0,0000	down
CST4	140,0818	0,0013	0,0000	0,0000	down
FAM25A	143,6501	0,0024	0,0000	0,0000	down
GRAP2	155,9459	0,0006	0,0000	0,0000	down
HENMT1	159,5171	0,0033	0,0000	0,0000	down
CCDC140	161,6660	0,0048	0,0000	0,0000	down

Table 3.13: Normalization with `EDASeq` and DE analysis with `RankProd` – top 6 identified genes.

This methodology outcome led to the identification of 340 differentially expressed genes for a FDR of 5%. From these, 285 were down regulated and 55 were up regulated. The "EDASeq+RP" plot for down regulated genes provides a visual observation to these results not only representing the genes (in red), but also because it complies with typical behaviour of `RankProd`, plotting the genes as an increasing curve. However, for the up regulated genes, around the 500th gene position according to RP value, a distortion is presented in the curve, which does not provide the usual behaviour of the FDR.

RankProd with`DESeq` normalized matrix

The same methodology, with the steps described in the previous topic, was applied to the normalized`DESeq` count matrix, which was imported from section 3.2.2.

The obtained results for the `plotRP()` function, for the same cutoff, are represented in figure 3.19, with the top six identified genes by this methodology represented in table 3.14.

The results of this method seem to induce far more optimism than the prior combination. They provide good plots, as expected by this method: where there is a rapid growth in the number of identified genes for the smallest considered FDR values. Also, analysing the full table output from these tests, we can see that it provides fair values and that the FDR increases with increasing RP values.

This joint methodology identified 149 genes as differentially expressed for a FDR of 5%. From these genes, 41 were down regu-

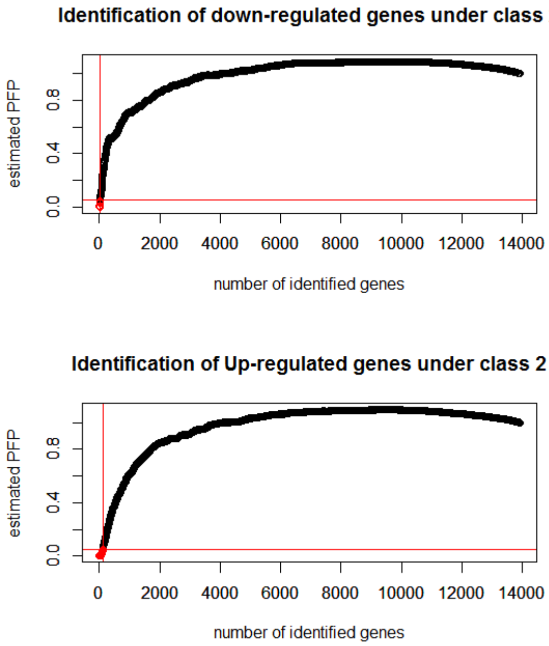

Figure 3.19: Normalization with`DESeq` and DE analysis with `RankProd` – FDR representation for genes ordered by RP, red dots corresponding to DEG.

lated, the other 108 were identified as up regulated.

3.4 Comparing methodologies

Although the focus of this thesis is on inferring differentially expressed genes between mutated and *wt* samples, it is pertinent to have a closer look on the normalization techniques applied and their effects on the samples.

Thus, it seemed interesting to see how the normalization effects correlate within the several methodologies applied. An interesting way to represent this is through `plot()`, for any of the samples tested. An over view of these relationships between normalization techniques was provided with `pair()` function, reflected in figure 3.20 for RCC_AB. This figure represents (on the log scale) the counts obtained for this sample, regarding the methods considered for normalization, and determines the correlation between them resorting to the Pearson correlation coefficient.

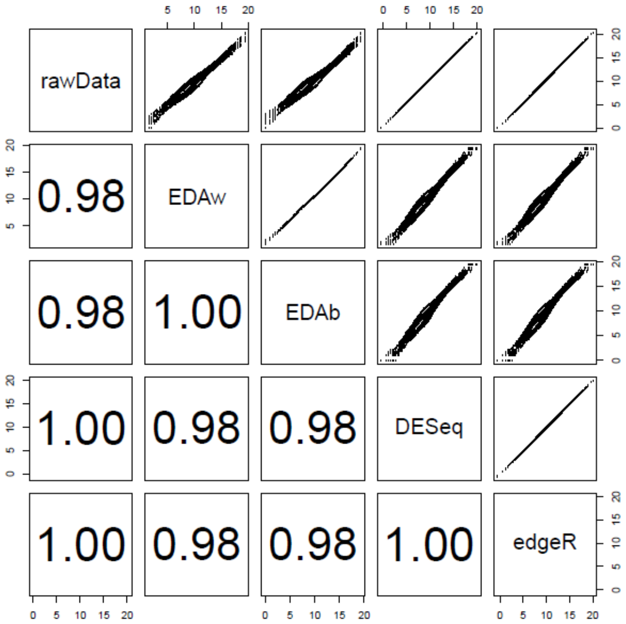

Figure 3.20: Pair plot for normalization for RCC_AB. This pair plot represents the several approaches taken on normalization and returns the correlation values associated with all comparisons.

id	RP/Rsum	Fold Change	FDR	P.value	Regulation
GPR173	20,2256	0,0016	0	0	down
EREG	22,3198	0,0037	0	0	down
IGFBP5	25,5953	2494,0709	0	0	up
NDN	30,6378	0,0242	0	0	down
SLC2A10	31,4153	0,0031	0	0	down
LOC100216001	43,7642	0,012	0	0	down

Table 3.14: Normalization withDESeq and DE analysis withDESeq – top 6 identified genes.

These results comply with the boxplots represented in figure 3.4, as they both reflect the smoothing effect provided with normalization on the data. The plot suggests that the genes each method selected to normalize are actually properly normalized, with high correlations, thus providing good input for differential analysis.

With reference to the normalization step, at this point, it is important to understand that as good as normalization step ends, the better DE analysis starts. And all the methods we have been studying ensure that normalization is robust and reliable, therefore supplying good raw material for the ultimate step, DE analysis.

As mentioned in the beginning of chapter 3, to assess differential expression, and to compare the results obtained with the several methodologies, we considered three cutoff values for FDR: 1%, 5% and 10%. The results obtained with this different considerations were summarized in figure 3.21.

From this figure we can observe that, when usingDESeq method for normalization, less genes are identified as DE, regardless of the method used for differential analysis (first three columns). EDASeq, on the other hand, seems to output the opposite result, leading to a higher number of DE (last three columns).

To be noted here is that edgeR normalized count matrix did not serve as input for the other methods. This is due to the fact that edgeR does not provide ways to export its count normalized data in a format that the other methods considered can read. This method served only to assess DE analysis regarding its own normalized count data.

Comparing the three given approaches in terms of normalization we can see thatDESeq, globally, reports the lowest counts – figure 3.21, first three columns – followed by edgeR – figure 3.21, forth column – and then EDASeq – figure 3.21, last three columns.

With the same figure, regarding DE analysis with different forms

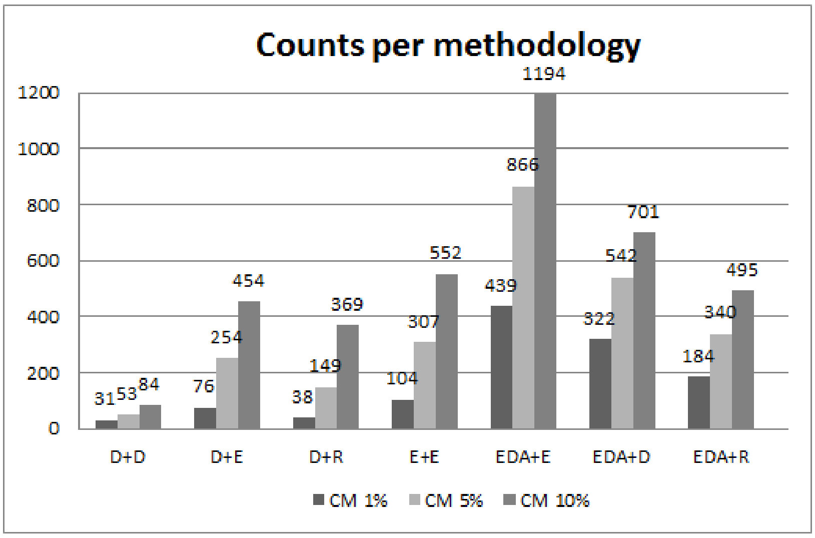

Figure 3.21: The figure represents the counts per methodology (CM) obtained for the three cutoff values of FDR (1%, 5% and 10%) for the several approaches taken, where D stands for`DESeq`, E stands for `edgeR`, R stands for `RankProd` and EDA stands for `EDASeq`. The + sign separates the two phases of the analysis, normalization and differential analysis.

of normalization, we can observe that it seems to perform best for `edgeR`, reporting a higher number of genes as differentially expressed (column 2 *versus* column 4 *versus* column 5) than for`DESeq` (column 1 *versus* column 6) or `RankProd` (column 3 *versus* column 7).

Another understanding from this figure is that, the higher the FDR considered, the higher the number of genes identified as differentially expressed. This because FDR will adjust for a higher p-value, allowing for more positive results. As mentioned, FDR is a statistical method used to correct for multiple comparisons: by choosing a higher cutoff criteria, we allow for a widen range of valid considered p-values and this means that more genes may be (falsely) identified as positive results. While a FDR of 10% can lead to bigger false positives, 1% may be too much of a stringent value. Thus, we proceeded the analysis with a FDR of 5%, clearing the previous figure 3.21 to figure 3.22.

Figure 3.22 shows the counts obtained for each methodology, while also evidencing the differences in the fold-change for the identified differentially expressed genes. We observed that the various combinations of normalization and DE analysis result in different

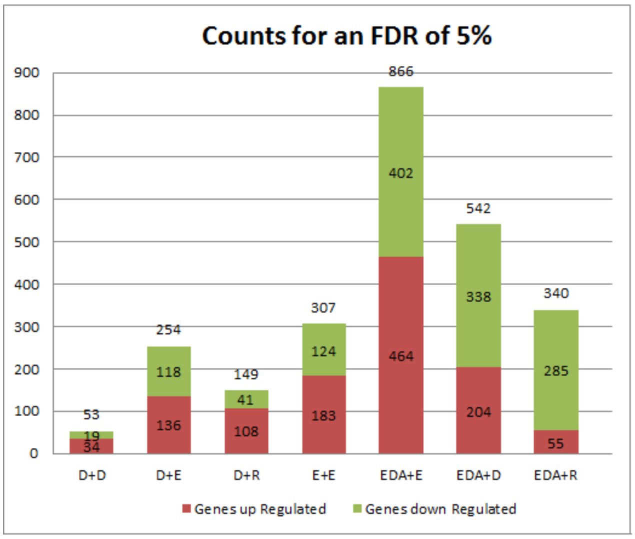

Figure 3.22: The figure represents the counts per methodology obtained for a FDR of 5%.

proportion of genes differentially identified as up or down regulated genes. With D+R leading to the lowest number of down regulated genes (41/149 = 28%), followed by D+D(36%), E+E(40%), EDA+E(46%), D+E(47%), EDA+D(62%) and EDA+R(84%) (with the highest number of identified down regulated genes). This result seems to point at EDASeq as a normalization method that leads to a better balance of differentially expressed genes.

Our analysis followed trough with comparing these methodologies among them. We know how many genes are detected as DE by the methodologies in study, but, regarding the main goal of this project, it is of key importance to learn if the methods identify the same genes as differentially expressed. In order to assess this, we first focused the analysis on determining how many genes these methodologies had in common for 5% FDR. and for the top x genes identified, how many were there in common. The results obtained were represented in table 3.15.

From table 3.15, on a first sight, we can immediately observe how EDASeq package regularly identifies a bigger number of differentially expressed genes. To a more extensive evaluation, and to simplify reader's comprehension, the table will be analysed first top/down. Thus, the first line compares self protocol methodologies, D+D *versus* E+E, leading to most genes identified with D+D to also being identified with E+E (43 out of the 53 genes reported

Methodology A	Methodology B	CM5% (A *vs* B)	top100	top500	top1000
D+D	E+E	43 (53 *vs* 307)	50	299	665
D+D	D+E	36 (53 *vs* 254)	49	263	649
D+D	D+R	23 (53 *vs* 149)	26	185	432
D+E	D+R	71 (254 *vs* 149)	28	240	524
D+E	E+E	200 (254 *vs* 307)	77	373	750
EDA+D	EDA+E	363 (542 *vs* 866)	32	246	502
EDA+D	EDA+R	106 (542 *vs* 340)	15	138	343
EDA+E	EDA+R	259 (866 *vs* 340)	39	293	616
EDA+E	E+E	202 (866 *vs* 307)	32	197	410
EDA+E	D+E	197 (866 *vs* 254)	41	275	561
EDA+D	D+D	48 (542 *vs* 53)	19	187	497
EDA+R	D+R	51 (340 *vs* 149)	20	240	563

Table 3.15: Methodology A and B stand for the two analysis in comparison as described in figure 3.21, at 5% FDR. The table also represents the total of genes identified as DE for the top 100, 500 and 1000 genes for both A and B.

as DE by D+D are also accounted with E+E). Furthermore, we can observe that these two methodologies seem to report that for a cutoff of X genes, more than 50% are identified by both. This points to an understanding that both methodologies have a similar internal computing process, as reported by Anders *et al* (Anders & Huber, 2012).

Proceeding to the next three lines (regarding protocols for`DESeq` **normalizations) we notice that these point to consistent values, with D+D *versus* D+R, reporting the lowest counts.**

The 5th line is comparing the effect of DE analysis for D+E *versus* E+E, thus assessing `edgeR` **response. This test reported consistently higher figures both for a FDR of 5% and fot top X genes. Which leads us to infer how robust** `edgeR` **package performance is (Oshlack *et al.* , 2010).**

EDASeq normalization – lines 6, 7 and 8 – reported the highest counts from all methodologies comparisons, with EDA+R presenting the lowest count values amongst them, even though they still report higher counts then all the other approaches considered on the data (figure 3.22)

When comparing the `EDASeq` **normalization effect against the**`DESeq` **normalization effect for the given DE analysis methods (last 3 lines) we can report that it consistently returns higher values for** `EDASeq`**. Given that** `EDASeq` **performs a two step normalization – within and**

between lane – whileDESeq focuses on accessing between lane's only, this can mean that EDASeq is accounting for genes that escapeDESeq scope.

A critical analysis proceeded focusing on the two steps of this methodology: normalization and DE analysis. Therefore, we determined how many DE genes were identified in common when the same normalization strategy was set:DESeq and EDASeq (figure 3.23). We then assessed how many DE were identified in common when the DE strategy was under test, edgeR,DESeq and RankProd (figure 3.24). A final consideration was appraised regarding self normalization and DE analysis, with edgeR andDESeq (figure 3.25). All these combinations were best represented trough Venn Diagrams representations, as follows.

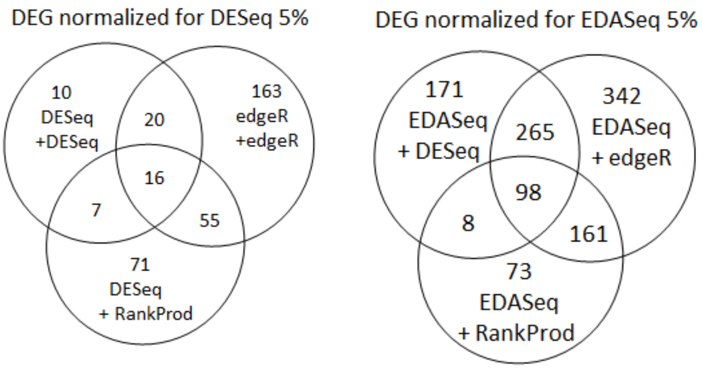

Figure 3.23: Venn diagram representing how many genes were identified as DE when the same normalization method was applied (i) normalized data withDESeq (identified 16 genes in common between the three DE strategies considered), (ii) normalized data with EDASeq (identified 98 genes in common between the three DE strategies).

From figure 3.23 we can observe that EDASeq normalization procedure leads to a higher number of genes identified as differentially expressed, which seems to point that EDASeq can be a less stringent normalization procedure thanDESeq.

Figure 3.24 enhances DE analysis results, regarding the methods chosen to normalize. The plot reverts edgeR as the method that leads to a higher number of genes identified has differentially expressed, while observing thatDESeq and RankProd appeared to be less selective.

Regarding self protocol methodologies, we can see that almost all the genes identified withDESeq full procedure (both normaliza-

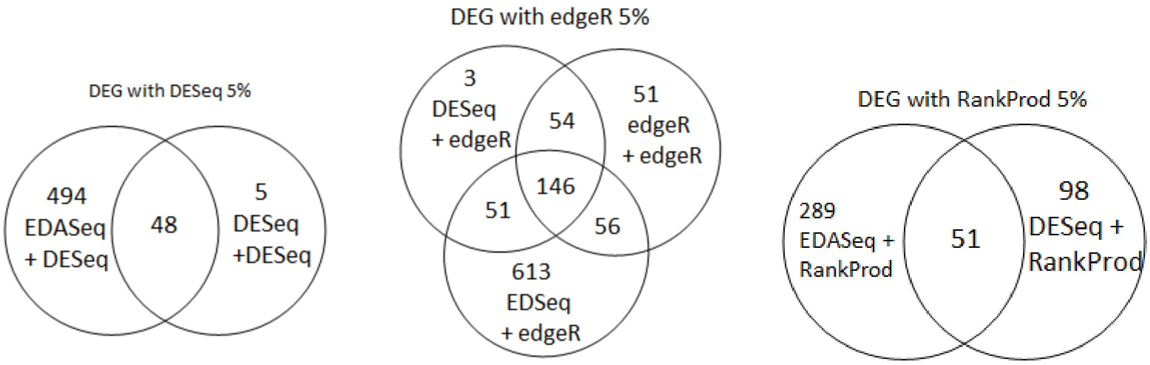

Figure 3.24: Venn diagram representing how many genes were identified as DE when the same DE analysis method was applied (i) DE data withDESeq (identified 48 genes in common between the two normalized strategies considered), (ii) DE with **edgeR** (identified 146 genes in common between the three normalized strategies considered) and (iii) DE with RankProd (identified 51 genes in common between the two normalized strategies considered).

tion and differentially expression gene analysis) were also identified as differentially expressed when edgeR self protocol was in study.

As a final inference we proposed to see how many genes were identified as differentially expressed among the 7 methodologies considered, resorting to intrinsic matching functions in R, and obtained a list of 6 genes for a 5% FDR:

```
[1] "SLC2A10"      "COL14A1"      "GPR173"       "LOC100506178"
[5] "EREG"         "ADAMTSL1"
```

Having observed that RankProd did not provide the most consistent results, particularly when provided with an EDASeq normalized count matrix (chapter 3.3.3), we were curious towards its influence in the final analysis and thus tested the same matching functions code in R, which led to 27 genes identified as differentially expressed for a 5%FDR:

```
[1]  "SLC2A10"      "COL14A1"      "GLUL"         "MLLT11"
[5]  "LOC100422737" "COL5A1"       "GPR173"       "BCAM"
[9]  "RASSF6"       "UNC5CL"       "CXADR"        "LOC100506178"
[13] "GPX3"         "ACOT7"        "CFI"          "EREG"
[17] "INSR"         "EPHA7"        "ADAMTSL1"     "ADAMTS16"
[21] "LOC440173"    "RAB11FIP4"    "GPC6"         "MYO15B"
[25] "TXNIP"        "TBX15"        "COBL"
```

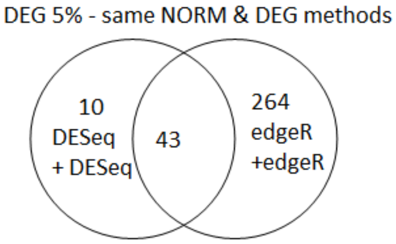

Figure 3.25: Venn diagram representing how many genes were identified as DE when the same method was applied for the two steps of the analysis: normalization and DE analysis (43).

Chapter 4

Discussion

This thesis main goal was to respond to a biological question on the influence of SETD2 mutations on ccRCC. To that extend, we analysed the obtained transcriptome for four mutated and two wild-type samples resorting to RNA-Seq technique.

This thesis workflow considered the analysis divided into two major steps: normalization and differential expression analysis. We aimed at studying different forms of data normalization and reported how this resulted in different outputs for the data expression analysis.

To a first extension, we aimed at comparing how different forms of normalization operate the data, all of these provided properly normalized count data: seen in figures 3.4 with the boxplots and 3.15 with the pair plots of Pearson correlations. These results of proper normalized data induce to a good prediction, since this is the raw material for DE analysis (Dillies *et al.* , 2012).

In order to accurately infer to our biological goal, we proposed to study several statistical ways to analyse the same data, thus aiming at obtaining a statistical support to our conclusions, regarding trully differentially expressed genes. This led to a complex set of combined analysis, where four methods (EDASeq, edgeR,`DESeq` and `RankProd`) were considered and all the possible logical combinations between them were followed trough, as follows: `EDASeq`+DESeq, `EDASeq`+edgeR, `EDASeq`+RankProd, `DESeq`+DESeq,`DESeq`+edgeR, DESerq+RankProd and edgeR+edgeR.

We observed that`DESeq` normalized procedures (with differential expression analysis resorting to`DESeq`, edgeR and `RankProd`) led to the lowest counts of differentially expressed genes: 53, 254 and 149 genes (respectively) were identified for a FDR of 5% – values taken from tables 3.21 and 3.22 – while EDASeq identified (for the same differential expression analysis approaches, and the same FDR) 542, 866 and 340 genes. We can see thatDESeq seems to offer a stricter normalization method, while EDASeq seems to have less stringent inherent methodologies to calculate normalizing factors. We know that EDASeq normalizes for both within and between samples, whileDESeq normalizes just between samples, which seem to point that either this method leaves relevant information out of the analysis, or it could be too selective. Another justification is that, given EDASeq normalizes for both, it benefits from a wider range of applicability.

When comparing methods to assess differentially expressed genes (to a FDR of 5%) regarding the different normalization techniques applied to it, edgeR revealed to be a more flexible method, identifying 254, 307 and 542 genes as differentially expressed forDESeq, edgeR and EDASeq normalization matrix. DESeq however, as a differentially expressed method, did not show a consistent outcome, identifying only 53 genes with self normalization and 542 when using EDASeq normalizing factors.

With RankProd, we achieved peculiar results: EDASeq plots are not conclusive, namely in what concerned up regulation, figure 3.18. The proportion of up regulated genes is lower than the proportion obtained by other methods (DESeq and edgeR, with the same normalization). On the other hand, the revealed boss could be an indicator that the method might not be appropriate for this kind of data – NGS. A possible, yet remote explanation for this, is that our data constitutes an unbalanced study, with four mutated samples to two *wt* samples.

Finally, the conjoint analysis for all the methods referred throughout this project, led to an identification of six differentially expressed genes with the analysis: "SLC2A10", "COL14A1", "GPR173", "LOC100506178", "EREG" and "ADAMTSL1".

Back to main goal of our project, the answer is that the above 6 genes respond to SETD2 mutations.

Many scientific questions remains unanswered for many years. NGS technique just started its contributions. Being so new, is expected to give more and more answers to the scientific challenges in the years to come.

Hopefully, these conclusions may lead to further investigations.

References

Anders, Simon. 2010. Analysing RNA-Seq data with the DESeq package. *Mol Biol*, 1–17.

Anders, Simon, & Huber, Wolfgang. 2010. Differential expression analysis for sequence count data. *Genome Biol*, 11(10), R106.

Anders, Simon, & Huber, Wolfgang. 2012. Differential expression of RNA-Seq data at the gene level -the DESeq package.

Benjamini, Yoav, & Hochberg, Yosef. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 289–300.

Benjamini, Yuval, & Speed, Terence P. 2012. Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic acids research*, 40(10), e72–e72.

Berglund, Eva C, Kiialainen, Anna, Syvänen, Ann-Christine, *et al.* . 2011. Next-generation sequencing technologies and applications for human genetic history and forensics. *Investigative genetics*, 2(1), 23.

Bioconductor. 2013. *Bioconductor: open source software for bioinformatics.* [http://www.bioconductor.org/; accessed 2013-September-10].

Breitling, Rainer, & Herzyk, Pawel. 2005. Rank-based methods as a non-parametric alternative of the T-statistic for the analysis of biological microarray data. *Journal of bioinformatics and computational biology*, 3(05), 1171–1189.

Breitling, Rainer, Armengaud, Patrick, Amtmann, Anna, & Herzyk, Pawel. 2004. Rank products: a simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments. ***FEBS letters***, 573(1), 83–92.

Bullard, James, Purdom, Elizabeth, Hansen, Kasper, & Dudoit, Sandrine. 2010. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. ***BMC bioinformatics***, 11(1), 94.

Carvalho, Sílvia. 2012. ***The role of SETD2 in transcription and DNA damage response.*** M.Phil. thesis, Faculty of Sciences, University of Lisbon.

CBCB. 2013. ***Center for Bioinformatics and Computational Biology – Software — CBCB.*** [`http://www.cbcb.umd.edu/software`; accessed 2013-September-21].

Cheng-Fu. 2013. ***Transcription Regulation Chromatin Structure and Dynamics.*** [`http://idv.sinica.edu.tw/ckao/Research%20interest.html`; accessed 2013-September-24].

Chial, H. 2008. DNA sequencing technologies key to the Human Genome Project. ***Nature Education***, 1(1).

Cohen, Herbert T, & McGovern, Francis J. 2005. Renal-cell carcinoma. ***New England Journal of Medicine***, 353(23), 2477–2490.

Dalgliesh, Gillian L, Furge, Kyle, Greenman, Chris, Chen, Lina, Bignell, Graham, Butler, Adam, Davies, Helen, Edkins, Sarah, Hardy, Claire, Latimer, Calli, ***et al.*** . 2010. Systematic sequencing of renal carcinoma reveals inactivation of histone modifying genes. ***Nature***, 463(7279), 360–363.

Damaschun, G, Damaschun, H, Misselwitz, R, Pospelov, VA, Zalenskaya, IA, Zirwer, D, Müller, JJ, & Vorobev, VI. 1983. How many base-pairs per turn does DNA have in solution and in chromatin? An answer from wide-angle X-ray scattering. ***Biomedica biochimica acta***, 42(6), 697.

Diamandis, Eleftherios P. 2000. Sequencing with microarray technology – a powerful new tool for molecular diagnostics. ***Clinical Chemistry***, 46(10), 1523–1525.

Dillies, Marie-Agnès, Rau, Andrea, Aubert, Julie, Hennequet-Antier, Christelle, Jeanmougin, Marine, Servant, Nicolas, Keime, Céline, Marot, Guillemette, Castel, David, Estelle, Jordi, *et al.* . 2012. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics*.

Duns, Gerben, van den Berg, Eva, van Duivenbode, Inge, Osinga, Jan, Hollema, Harry, Hofstra, Robert MW, & Kok, Klaas. 2010. Histone methyltransferase gene SETD2 is a novel tumor suppressor gene in clear cell renal cell carcinoma. *Cancer research*, 70(11), 4287–4291.

e Silva, Alexandra Maia, Quintas, Alexandre, Halpern, Ana Ponces, Ascenso, Carla, Videira, Arnaldo, Azevedo, Carlos, & de Oliveira, Carlos Resende. 2008. *Bioquímica: organização molecular da vida*.

Gallant. 2013. *Gallant's Biology Stuff*. [`http://kvhs.nbed.nb.ca/gallant/biology/nephron_structure.html`; accessed 2013-June-16].

Hansen, Kasper D, Irizarry, Rafael A, & Zhijin, WU. 2012. Removing technical variability in RNA-seq data using conditional quantile normalization. *Biostatistics*, 13(2), 204–216.

Hong, Fangxin. 2011. Bioconductor RankProd Package Vignette.

Hong, Fangxin, Breitling, Rainer, McEntee, Connor W, Wittner, Ben S, Nemhauser, Jennifer L, & Chory, Joanne. 2006. RankProd: a bioconductor package for detecting differentially expressed genes in meta-analysis. *Bioinformatics*, 22(22), 2825–2827.

Illumina. 2013. *Genomic Sequencing*. [`http://res.illumina.com/documents/products/datasheets/datasheet_genomic_sequence.pdf`; accessed 2013-September-9].

Kadota, Koji, Nishiyama, Tomoaki, Shimizu, Kentaro, *et al.* . 2012. A normalization strategy for comparing tag count data. *Algorithms for Molecular Biology*, 7(5).

Latchman, David. 2007. *Gene regulation*. Psychology Press.

Lewin, Benjamin. 2004. *Genes Viii.* Pearson Prentice Hall Upper Saddle River.

McCarthy, Davis J, Chen, Yunshun, & Smyth, Gordon K. 2012. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic acids research*, 40(10), 4288–4297.

Nabi, Ghulam, Cleves, Anne, & Shelley, Mike. 2010. Surgical management of localised renal cell carcinoma. *Cochrane Database Syst Rev*, 3.

Nowrousian, Minou. 2010. Next-generation sequencing techniques for eukaryotic microorganisms: sequencing-based solutions to biological problems. *Eukaryotic cell*, 9(9), 1300–1310.

ORNLaboratory. 2013. *The Human Genome Project Information Archive.* [`web.ornl.gov/sci/techresources/Human_Genome/index.url`; accessed 2013-September-7].

Oshlack, Alicia, & Wakefield, Matthew J. 2009. Transcript length bias in RNA-seq data confounds systems biology. *Biol Direct*, 4(1), 14.

Oshlack, Alicia, Robinson, Mark D, Young, Matthew D, *et al.* . 2010. From RNA-seq reads to differential expression results. *Genome Biol*, 11(12), 220.

Pickrell, Joseph K, Marioni, John C, Pai, Athma A, Degner, Jacob F, Engelhardt, Barbara E, Nkadori, Everlyne, Veyrieras, Jean-Baptiste, Stephens, Matthew, Gilad, Yoav, & Pritchard, Jonathan K. 2010. Understanding mechanisms underlying human gene expression variation with RNA sequencing. *Nature*, 464(7289), 768–772.

Rapaport, Franck, Khanin, Raya, Liang, Yupu, Krek, Azra, Zumbo, Paul, Mason, Christopher E, Socci, Nicholas D, & Betel, Doron. 2013. Comprehensive evaluation of differential expression analysis methods for RNA-seq data. *arXiv preprint arXiv:1301.5277.*

Risso, Davide. 2011. EDASeq: Exploratory Data Analysis and Normalization for RNA-Seq. *R package version*, 1(0).

Risso, Davide, Schwartz, Katja, Sherlock, Gavin, & Dudoit, Sandrine. 2011. GC-content normalization for RNA-Seq data. *BMC bioinformatics*, 12(1), 480.

Robinson, Mark, McCarthy, Davis, Chen, Yunshun, & Smyth, Gordon K. 2010. edgeR: differential expression analysis of digital gene expression data User's Guide.

Robinson, Mark D, & Oshlack, Alicia. 2010. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol*, 11(3), R25.

Robinson, Mark D, & Smyth, Gordon K. 2007. Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, 23(21), 2881–2887.

Robinson, Mark D, & Smyth, Gordon K. 2008. Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*, 9(2), 321–332.

Singer, Eric A, Gupta, Gopal N, & Srinivasan, Ramaprasad. 2012. Targeted therapeutic strategies for the management of renal cell carcinoma. *Current opinion in oncology*, 24(3), 284.

Talbert, Paul B, Ahmad, Kami, Almouzni, Geneviève, Ausió, Juan, Berger, Frederic, Bhalla, Prem L, Bonner, William M, Cande, W Zacheus, Chadwick, Brian P, Chan, Simon WL, *et al.* . 2012. A unified phylogeny-based nomenclature for histone variants. *Epigenetics & chromatin*, 5(1), 1–19.

TheMedicalNews. 2013. *What is Gene Expression?* [http://www.news-medical.net/health/What-is-Gene-Expression.aspx; accessed 2013-September-5].

Turner, Bryan M. 2005. Reading signals on the nucleosome with a new nomenclature for modified histones. *Nature structural & molecular biology*, 12(2), 110–112.

Voelkerding, Karl V, Dames, Shale A, & Durtschi, Jacob D. 2009. Next-generation sequencing: from basic research to diagnostics. *Clinical chemistry*, 55(4), 641–658.

Zhang, Jun, Chiodini, Rod, Badr, Ahmed, & Zhang, Genfa. 2011. The impact of next-generation sequencing on genomics. *Journal of Genetics and Genomics*, 38(3), 95–109.