BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

TRANSCRIPTOMIC CHARACTERIZATION USING RNA-SEQ DATA ANALYSIS

TRANSKRIPTOMICKÁ CHARAKTERIZACE POMOCÍ ANALÝZY RNA-SEQ DAT

DOCTORAL THESIS

DIZERTAČNÍ PRÁCE

AUTHOR AUTOR PRÁCE	Layal Abo Khayal
SUPERVISOR ŠKOLITEL	prof. Ing. Ivo Provazník, Ph.D.

BRNO 2017

ABSTRACT

The high-throughputs sequence technologies produce a massive amount of data, that can reveal new genes, identify splice variants, and quantify gene expression genome-wide. However, the volume and the complexity of data from RNA-seq experiments necessitate a scalable, and mathematical analysis based on a robust statistical model. Therefore, it is challenging to design integrated workflow, that incorporates the various analysis procedures. Particularly, the comparative transcriptome analysis is complicated due to several sources of measurement variability and poses numerous statistical challenges. In this research, we performed an integrated transcriptional profiling pipeline, which generates novel reproducible codes to obtain biologically interpretable results. Starting with the annotation of RNA-seq data and quality assessment, we provided a set of codes to serve the quality assessment visualization needed for establishing the RNA-Seq data analysis experiment. Additionally, we performed comprehensive differential gene expression analysis, presenting descriptive methods to interpret the RNA-Seq data. For implementing alternative splicing and differential exons usage analysis, we improved the performance of the Bioconductor package DEXSeq by defining the open reading frame of the exonic regions, which are differentially used between biological conditions due to the alternative splicing of the transcripts. Furthermore, we present a new methodology to analyze the differentially expressed long non-coding RNA, by finding the functional correlation of the long non-coding RNA with neighboring differential expressed protein coding genes. Thus, we obtain a clearer view of the regulation mechanism, and give a hypothesis about the role of long non-coding RNA in gene expression regulation.

KEYWORDS:

RNA-Seq, Differential Gene Expression (DGE), Alternative splicing, Differential Exon Usage (DEU), long non-coding RNA (lncRNA).

ABSTRAKT

Vysoce výkonné sekvenční technologie produkují obrovské množství dat, která mohou odhalit nové geny, identifikovat splice varianty a kvantifikovat genovou expresi v celém genomu. Objem a složitost dat z RNA-seq experimentů vyžadují škálovatelné metody matematické analýzy založené na robustníchstatistických modelech. Je náročné navrhnout integrované pracovní postupy, které zahrnují různé postupy analýzy. Konkrétně jsou to srovnávací testy transkriptů, které jsou komplikovány několika zdroji variability měření a představují řadu statistických problémů. V tomto výzkumu byla sestavena integrovaná transkripční profilová pipeline k produkci nových reprodukovatelných kódů pro získání biologicky interpretovovatelných výsledků. Počínaje anotací údajů RNA-seq a hodnocení kvality je navržen soubor kódů, který slouží pro vizualizaci hodnocení kvality, potřebné pro zajištění RNA-Seq experimentu s analýzou dat. Dále je provedena komplexní diferenciální analýza genových expresí, která poskytuje popisné metody pro testované RNA-Seq data. Pro implementaci analýzy alternativního sestřihu a diferenciálních exonů jsme zlepšili výkon DEXSeq definováním otevřeného čtecího rámce exonového regionu, který se používá alternativně. Dále je popsána nová metodologie pro analýzu diferenciálně exprimované dlouhé nekódující RNA nalezením funkční korelace této RNA se sousedícími diferenciálně exprimovanými geny kódujícími proteiny. Takto je získán jasnější pohled na regulační mechanismus a poskytnuta hypotéza o úloze dlouhé nekódující RNA v regulaci genové exprese.

KLÍČOVÁ SLOVA

RNA-Seq, diferenciální genová exprese (DGE), alternativní splicing, diferenciální použití exonů (DEU), dlouhá nekódující RNA (lncRNA).

BIBLIOGRAPHIC CITATION

Abo Khayal, Layal. Transcriptomic characterization using RNA-Seq data analysis

Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, 2018. 125 p. Academic advisor: Prof. Ing. Ivo Provazník, Ph.D.

DECLARATION

I declare that I have written my doctoral thesis on the theme of “Transcriptomic characterization using RNA-Seq data analysis” independently, under the guidance of the doctoral thesis supervisor and using the technical literature and other sources of information which are all quoted in the thesis and detailed in the list of literature at the end of the thesis.

As the author of the doctoral thesis I furthermore declare that, as regards the creation of this doctoral thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone’s personal and/or ownership rights and I am fully aware of the consequences in the case of breaking Regulation §11 and the following of the Copyright Act No 121/2000 Sb., and of the rights related to intellectual property right and changes in some Acts (Intellectual Property Act) and formulated in later regulations, inclusive of the possible consequences resulting from the provisions of Criminal Act No 40/2009 Sb., Section 2, Head VI, Part 4.

Brno 18.10.2017

....
Ing. Layal Abo Khayal

ACKNOWLEDGEMENT

I would like to thank all the people who supported me during my doctoral study in the Department of Biomedical Engineering at Brno University of Technology, and during my research stay in Charité medical university, Germany. Especially, I thank my supervisor Prof. Ing. Ivo Provaznik, Ph.D. head of the Department of Biomedical Engineering at Brno University of Technology for his endless support. I would like also to express my gratitude and appreciation to my external supervisor Prof. Dr. Med. Peter N Robinson from Charité Berlin, for his professional guidance. Special thanks to my family in Syria and friends for their great patience, persistence, and inspiration. I dedicate my PhD thesis to my mother, and to my great mother and homeland Syria, wishing peace to return to its whole lands soon.

Brno, 18.10.2017

……………………………..

Ing. Layal Abo Khayal

TABLE OF CONTENTS

INTRODUCTION TO RNA SEQUENCING

RNA-seq can be identify as an assembly of experimental and computational methods to determine the identity and abundance of RNA sequences in biological samples. The experimental methods involve isolation of RNA from cell, tissue, or whole-animal samples, preparation of libraries that represent RNA species in the samples, actual chemical sequencing of the library, and subsequent bioinformatic data analysis. A critical distinction of RNA-seq from earlier methods, such as microarrays, is the incredibly high throughput of current RNA-seq platforms, the sensitivity afforded by newer technologies, and the ability to discover novel transcripts, gene models, and small noncoding RNA species.

RNA-seq methods are derived from generational changes in sequencing technology. First-generation high-throughput sequencing typically refers to Sanger sequencing. With capillary electrophoresis being utilized to deal with nucleic acid fragment lengths, a standard run might employ 96 capillaries and generate a sequence length in range of 600 to 1000 bases yielding approximately 100,000 bases of sequence. Second-generation sequencing, also known as next-generation sequencing (NGS), refers to methods using similar sequencing by synthesis chemistry of individual nucleotides, but performed in a massively parallel format, so that the number of sequencing reactions in a single run can be in millions. A typical NGS run could consist of 6000M sequencing reactions of 100 nucleotides yielding 600 billion bases of sequence information. Third-generation sequencing refers to methods that are also massively parallel and use sequence by synthesis chemistry but have as templates individual molecules of DNA or RNA. Third-generation sequencing platforms have fewer sequencing reactions per run, in the order of a few millions, but the length of sequence per reaction can be larger and can easily run into the 1500 nucleotide range [1].

Data obtained from an RNA-seq experiment can be substantially informative, ranging from the identification de novo protein coding transcripts in embryonic stem cells to characterization of gene regulation and alternative splicing. Questions that can be answered using RNA-seq data include: What are the differences in the levels of gene expression in normal and cancer cells? What happens to the gene expression levels in cell lines missing a tumor suppressor gene? Which genes are up-regulated during the development of brain? How is gene splicing changed during oxidative stress? What novel miRNAs can we discover in a human embryonic stem-cell sample?

New data derived from RNA-seq platforms showed a vast diversity for gene structure, identified novel unknown genes, and shed light on noncoding transcripts of both small and long lengths [2]. The following General scheme for RNA-seq experiments illustrates the workflow from tissue to data in the RNA-seq method. It is shown with alternatives for CLIP-seq, miRNA-seq, and general RNA-seq. The isolation is performed for RNAs and RNA- protein complexes, then a size selection is applied to separate the libraries construction. (Figure 0.1)

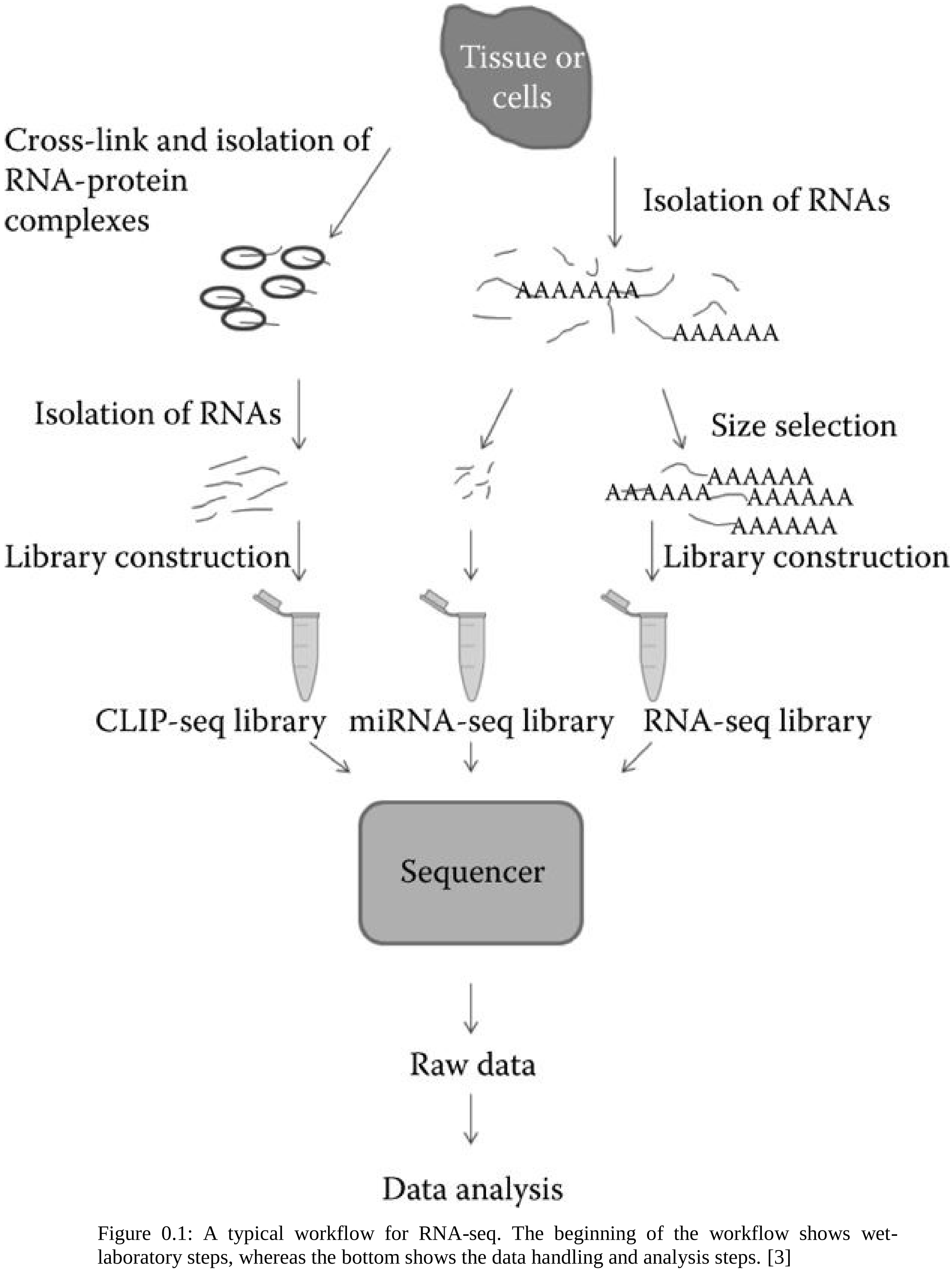

Figure 0.1: A typical workflow for RNA-seq. The beginning of the workflow shows wet-laboratory steps, whereas the bottom shows the data handling and analysis steps. [3]

1. THE THEORETICAL REVIEW

1.1. ISOLATION OF RNAs

RNAs are typically isolated from freshly dissected or frozen cells or tissue samples using commercially available kits. High-throughput RNA isolation systems also exist that relies mainly on RNA attached to magnetic particles which facilitate their washing and isolation. It is also possible, although not ideal, to isolate RNA from formalin-fixed, paraffin-embedded tissues. To prevent degradation of RNA, samples can be immersed in RNA storage reagents such as RNAlater (Ambion), or processed partially and stored as a phenolic emulsion (Trizol). At this stage, RNA samples can also be enriched for size-specific classes such as small RNAs using column systems (miRVana; Ambion). Alternatively, samples can be isolated initially as total RNA and then size selected by polyacrylamide gel electrophoresis.[4]

In almost all cases of total RNA isolation, genomic DNA will contaminate the sample. This is unavoidable, and even if the contamination is minor, the sensitivity and throughput of RNA-seq will eventually capture these contaminants. Therefore, it is common practice that total RNA-isolated samples are treated with DNase, to digest contaminating DNA prior to library preparation. Most DNase kits provide reagents for inactivating DNase once the contaminating DNA has been removed. The amount of total RNA required for RNA-seq library preparation varies. Standard library protocols require 0.1–10 μg of total RNA, and high-sensitivity protocols can produce libraries from as little as 10pg of RNA. It is becoming common that RNA from single cells is isolated and specific kits for these applications are becoming available. [3]

1.2. QUALITY CONTROL OF RNA

It is required that RNAs are quality checked for degradation, purity, and quantity prior to library preparation. Nanodrop and similar devices measure the fluorescent absorbance of nucleic acid samples typically at 260 and 280nm. As the device measures absorbance of the sample, it is not able to distinguish between RNA and DNA, and therefore cannot indicate whether the RNA sample is contaminated with DNA. Moreover, degraded RNA will give similar readings as intact RNA, and therefore we cannot know about the quality of the sample. The 260/280 absorbance ratio will, however, provide some information about contamination by proteins.[5]

Agilent Bioanalyzer is a microfluidics capillary electrophoresis-based system to measure nucleic acids. The Agilent Bioanalyzer offers advantages of sensitivity and accuracy for performing RNA separation, detection, and quantitation, coupled with a rapid, automated system. The electrophoresis being used for sizing nucleic acid samples. When size standards are run, the sizing and quantitation of RNAs in the sample provides critical information not only on the concentration, but also on the quality of nucleic acid. Degraded RNAs will appear as a smear at low-molecular weights, whereas intact total RNA will show sharp 28S and 18S peaks. The Bioanalyzer system contains a microchip that is loaded with size controls and space for up to 12 samples at a time. Samples are mixed with a polymer and a fluorescent dye, which are then loaded and measured through capillary electrophoretic movement. The integrated data analysis pipeline on the instrument will also render the electrophoretic data into a gel-like picture for users more accustomed to traditional gel electrophoresis. The RNA profile of each sample is automatically displayed as individual electropherograms. A comparison of the results from the gel electrophoresis to the Agilent 2100 Bioanalyzer digital gel (Figure 1.1a) and to the corresponding electropherograms (Figure 1.1 b).[6]

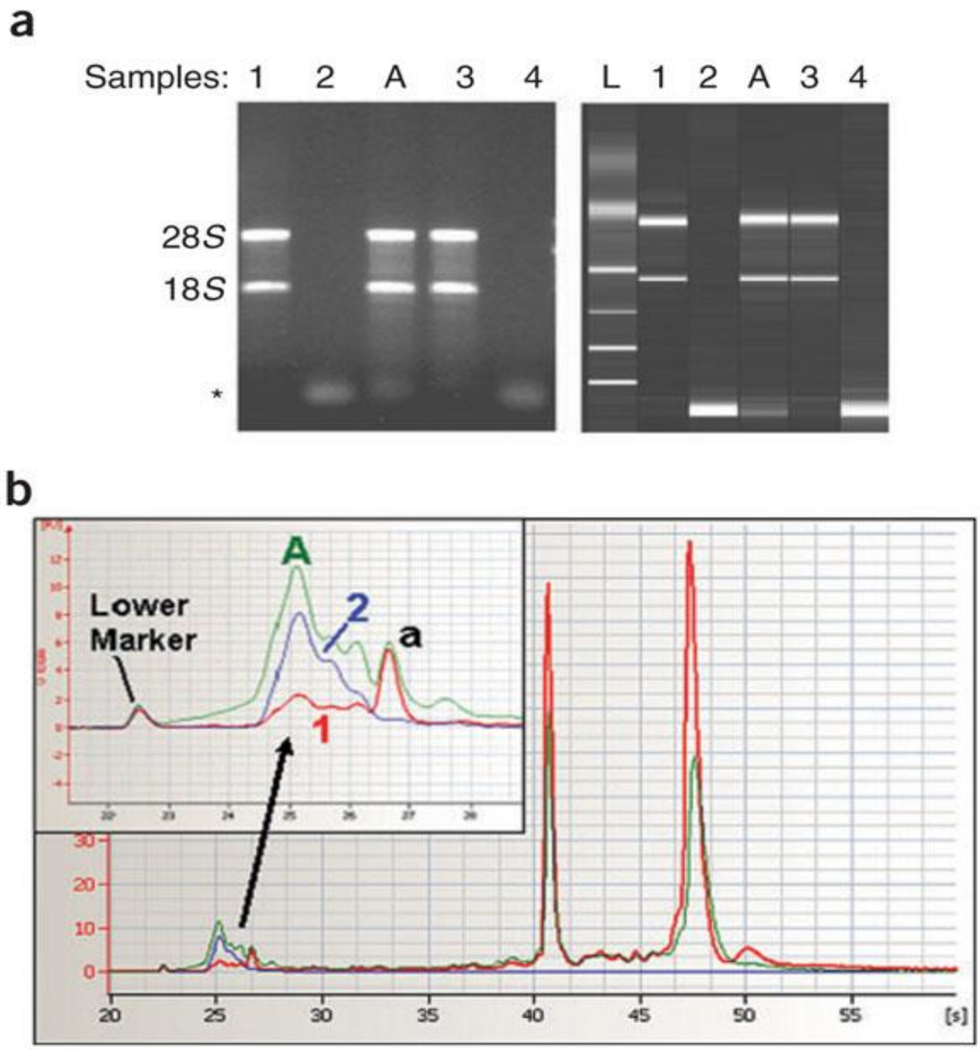

Figure 1.1: Agilent Bioanalyzer. (a) Agarose gel (1%, left) of total RNA (sample A), LMW “low-molecular-weight” RNAs (samples 2 and 4) and HMW “high-molecular-weight” RNAs (samples 1 and 3) compared to the digital gel (right) obtained with the Agilent 2100 Bioanalyzer (L, ladder; 500 ng per sample for all RNAs). (b) Screen capture of Agilent 2100 Bioanalyzer electropherogramsof sample A (total RNA), sample 1 (HMW RNA) and sample 2 (LMW RNA). Peak labeled "a" (inset) was detected only in total and HMW RNA samples (A and 1) but not in LMW RNA–enriched sample (2).[6]

1.3. LIBRARY PREPARATION

Before to sequencing, the RNAs in a sample are converted into a cDNA library, representing all the RNA molecules in the sample. This step is performed because in practice, RNA molecules are not directly sequenced, instead DNAs are sequenced due to their better chemical stability, and are also more amenable to the sequencing chemistry and protocols of each sequencing platform. Therefore, the library preparation has two purposes, the first is to adequately represent the RNAs in the sample and secondly to convert RNA into DNA. Each RNA-sequencing platform (e.g., Illumina, Solid, Ion Torrent) has its own specific protocol. Third-party library preparation kits are also available and are being used successfully. It is also possible to create one's own kit using commonly available molecular biology components although this lacks the convenience, optimization, and support of commercial products.

The major steps in library preparation involves the following[7]:

1. Obtain pure, intact, and quality-checked total RNA of approximately 1–10 μg. The exact amount needed depends on the application and platform.
2. Purify mRNA from the total RNA. Typically, this is done by annealing total RNA to oligo-dT magnetic beads. Two rounds of purification may be performed to remove nonspecifically bound ribosomal and other RNAs from the oligo-dT. Then release or dissociate mRNAs from oligo-dT beads.
3. Fragment purified mRNA by incubation with fragmentation reagent. This breaks the mRNA strands into multiple small fragments.
4. Prime the fragmented mRNAs with random hexamer primers.
5. Reverse-transcribe the fragmented mRNAs with Reverse Transcriptase, thus producing cDNAs.
6. Synthesize the second/opposite strand of the cDNA and remove the RNA. The product will be double-stranded cDNA **(ds cDNA).**
7. Purify the ds cDNA from free nucleotides, enzymes, buffers, and RNA. This can be done by binding the DNA with Solid-Phase Reversible Immobilization (SPRI) beads, for example. The advantage of using these paramagnetic beads is that once bound, the beads can be washed to purify the ds cDNA which remains on the beads. Once washed, the ds cDNA can be eluted from the beads for the next procedure.
8. Perform end-repair on purified eluted ds cDNA.
9. Purify the end-repaired ds cDNA. This can also be done on SPRI beads.
10. Adenylate 3′ ends of eluted end-repaired ds cDNA.
11. Ligate adaptors to the end-repaired ds cDNA. Adaptors will ligate to both ends of the ds cDNA. These adaptors can be indexed for each library reaction. In other words, each adaptor can have a six-nucleotide difference in the adaptor sequence. Using a different index for each library reaction allows for pooling libraries later for sequencing, yet still allowing for tracing the sequence back to the original library based on the adaptor sequence.
12. Purify the adaptor ligated, end-repaired ds cDNA. Again, this can be done with SPRI beads.
13. Enrich the library by polymerase chain reaction (PCR) amplification. Using sequences from the adaptor as primers, small numbers of cycles (12–16) are used to amplify the sequences already present.
14. Purify the PCR-enriched, adaptor-ligated, end-repaired ds cDNA. This is now the library representing the original mRNAs in the sample.
15. Validate and quality-control the library. This can be done in several ways by (1) selectively amplifying via PCR-specific genes that should be present in the library; (2) quantifying the yield of ds cDNA in the library; (3) visualizing the abundance and size distribution of the library by polyacrylamide gel electrophoresis, or capillary electrophoresis on an Agilent Bioanalyzer.
16. Normalize and pool libraries. As the capacity to sequence in a single flow cell is enormous, it is possible to sequence many libraries (up to 24 libraries/flow cell lane is possible). Normalization acts to even out the amounts of ds cDNA in each library. For example, all libraries can be diluted to 10nM ds cDNA and then pooled at even volumes, so that each library is equally represented.
17. Send normalized and pooled libraries to sequencing facility for cluster generation and sequencing protocol which is dependent on the specific platform (Illumina, Solid, 454, etc.).

The Figure 1.2 shows an improved RNA-seq library preparation workflow that includes highly efficient rRNA removal (Ribo-Zero technology) followed by a rapid, ligation-free cDNA synthesis procedure for preparing directional RNA-seq libraries (ScriptSeq v2 technology). [8]

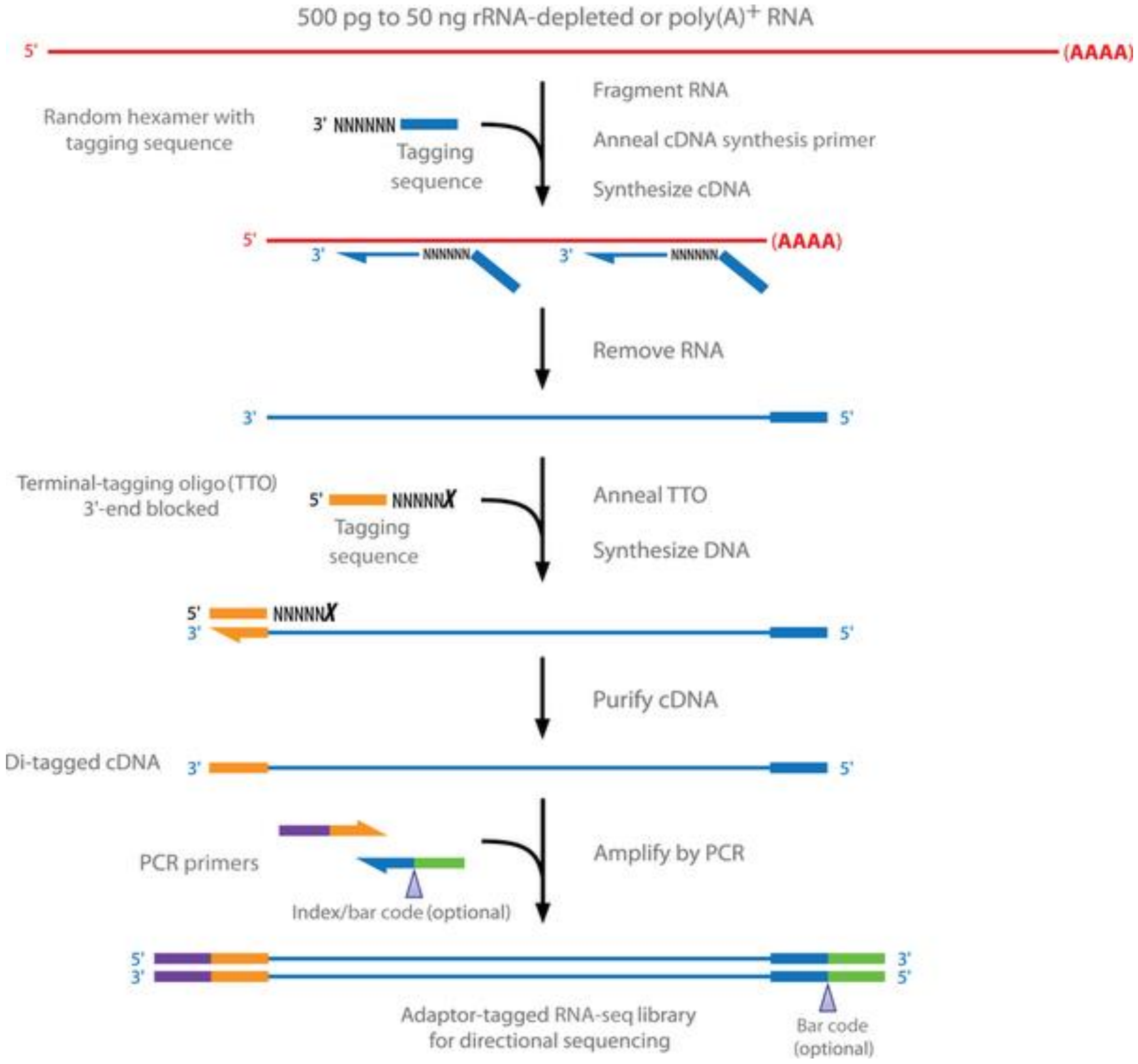

Figure 1.2: Overview of the ScriptSeq v2 RNA-seq library preparation method.[8]

1.4. RNA-SEQ PLATFORMS

1.4.1 ILLUMINA

After libraries are made, ds cDNA is passed through a flow cell which will hybridize the individual molecules based on complementarity with adaptor sequences. Hybridized sequences held at both ends of the adaptor by the flow cell will be amplified as a bridge. These newly generated sequences will hybridize to the flow cell close by and after many cycles a region of the flow cell will contain many copies of the original ds cDNA. This entire process is known as cluster generation. After the clusters are generated, and one strand removed from the ds cDNA, reagents are passed through the flow cell to execute sequencing by synthesis. Sequencing by synthesis describes a reaction where in each synthesis round, the addition of a single nucleotide, which can be A, C, G, or T, as determined by a fluorescent signal, is imaged, so that the location and added nucleotide can be determined, stored, and analyzed. Reconstruction of the sequence of additions in a specific location on the flow cell, which corresponds to a generated ds cDNA cluster, gives the precise nucleotide sequence for an original piece of ds cDNA [9]. As it is illustrated in the Figure 1.3 [10].

There are also two modes in which sequencing can be performed. If sequencing is performed at one end of the ds cDNA only, it is single read mode. If sequencing is performed from both ends, it is termed paired-end read mode. Illumina provides a wide range of instruments with different throughputs. The Hi-Seq 2500 instrument produces up to 6 billion paired-end reads in a single run. At PE100 (paired-end read with length 100 nucleotides), this represents 600Gb of data. This is massively more sequence data than is typically needed for a single study, so that, in practice, the libraries are indexed and several libraries are normalized and combined to be run on a single flow cell. It is normal practice to have as many as a hundred libraries run in total on a 16-lane flow cell. If this is too much sequencing capacity for a laboratory, Illumina also provides a smaller, yet more personal sequencer with lower throughput. The MiSeq system can produce 30M reads in PE250 mode representing 8.5Gb of data within a 2-day runtime. [11]

1.4.2 SOLID

SOLID stands for sequencing by oligonucleotide ligation and detection and is a platform. The sequencing chemistry is via ligation rather than synthesis. In the SOLID platform, a library of DNA fragments (originally derived from RNA molecules) is attached to magnetic beads at one molecule per bead. The DNA on each bead is then amplified in an emulsion so that amplified products remain with the bead. The resulting amplified products are then covalently bound to a glass slide. Using several primers that hybridize to a universal primer, di-base probes with fluorescent labels are competitively ligated to the primer. If the bases in the first and second positions of the di-base probe are complementary to the sequence, then the ligation reaction will occur and the label will provide a signal. Primers are reset five times by a single nucleotide, so at the end of the cycle, at least four nucleotides would have been interrogated twice due to the dinucleotide probes and the fifth nucleotide at least once. The ligation steps continue until the sequence is ready.[12]

The unique ligation chemistry allows for two checks of a nucleotide position and thus provides greater sequencing accuracy of up to 99.99%. While this may not be necessary for applications such as differential expression, it is critical for detecting single-nucleotide polymorphisms (SNPs). The newest instruments such as the 5500W do away with bead amplification and use flow chips in place of amplifying templates. The throughput can be up to 320Gb of data from two flow chips. As with other platforms, indexing/barcoding can be used to multiplex libraries so that hundreds of library samples can be run simultaneously on the instrument. Figure 1.4 illustrates Solid sequencing process. [13]

1.4.3 ROCHE 454

This platform is also based on adaptor-ligated ds DNA library sequencing by synthesis chemistry. ds DNA is fixed onto beads and amplified in a water–oil emulsion. The beads are then placed into picotiter plates where sequencing reactions take place. The massive numbers of wells in picotiter plates provide the massively parallel layout needed for NGS.

The detection method differs from other platforms in that the synthesis chemistry involves detection of an added nucleotide via a two-step reaction. The first step cleaves the triphosphate nucleotide, then releasing pyrophosphate. The second step converts pyrophosphate into adenosine triphosphate (ATP) via the enzyme ATP sulfurylase. The third step uses the newly synthesized ATP to catalyze the conversion of luciferin into oxyluciferin via the enzyme luciferase and this reaction generates a quantum of light that is captured from the picotiter plate by a charge-coupled camera.

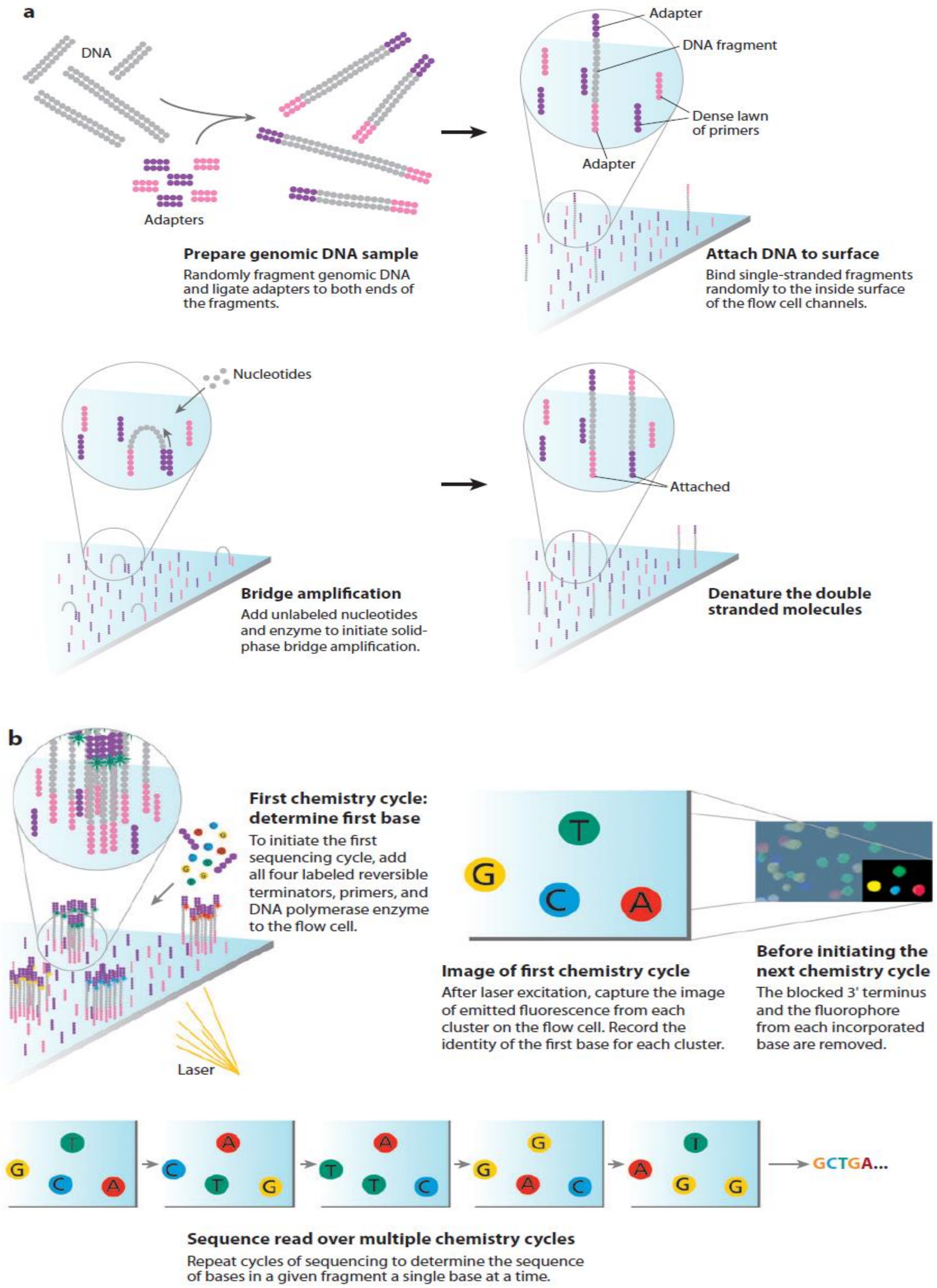
a
DNA
Adapters
Prepare genomic DNA sample
Randomly fragment genomic DNA and ligate adapters to both ends of the fragments.
Adapter
DNA fragment
Dense lawn of primers
Adapter
Attach DNA to surface
Bind single-stranded fragments randomly to the inside surface of the flow cell channels.
Nucleotides
Bridge amplification
Add unlabeled nucleotides and enzyme to initiate solid-phase bridge amplification.
Attached
Denature the double stranded molecules
b
First chemistry cycle: determine first base
To initiate the first sequencing cycle, add all four labeled reversible terminators, primers, and DNA polymerase enzyme to the flow cell.
Laser
T
G
C
A
Image of first chemistry cycle
After laser excitation, capture the image of emitted fluorescence from each cluster on the flow cell. Record the identity of the first base for each cluster.
Before initiating the next chemistry cycle
The blocked 3' terminus and the fluorophore from each incorporated base are removed.
T G C A
A C T G
A T T C
G G A C
T A G G
GCTGA...
Sequence read over multiple chemistry cycles
Repeat cycles of sequencing to determine the sequence of bases in a given fragment a single base at a time.

Figure 1.3: The Illumina sequencing-by-synthesis approach.[10]

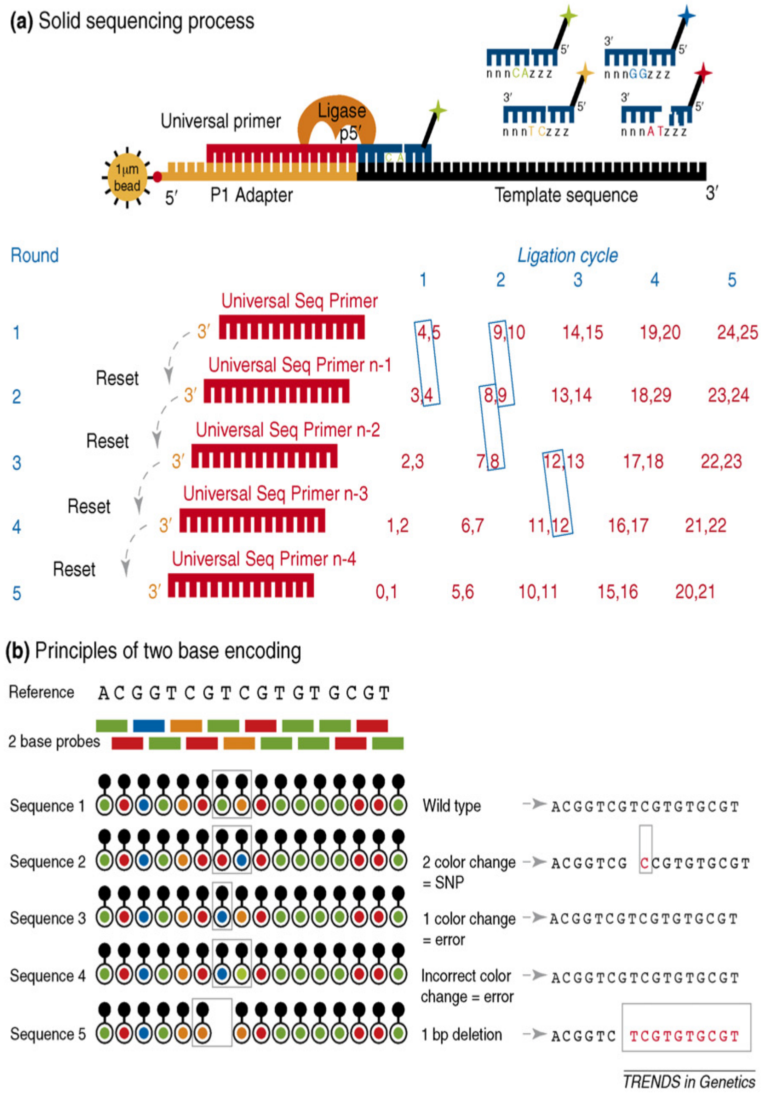

Figure 1.4: AB SOLiD sequencing. (a) AB SOLiD sequencing by ligation first anneals a universal sequencing primer then goes through subsequent ligation of the appropriate labeled 8mer, followed by detection at each cycle. (b) Two base encoding of the AB SOLiD data greatly facilitates the discrimination of base callingerrors from true polymorphisms or indel events. Figures related to the SOLiD(tm) System are reproduced with permission from Applied Biosystems.[13]

Free nucleotides and unreacted ATP are degraded by a PYRase after each addition. These steps are repeated until a predetermined number of reactions have been reached. Recording the light generation and well location after each nucleotide addition allows for reconstruction of the identity of the nucleotide and the sequence for each well. The advantage of this sequencing chemistry is that it permits for longer reads when compared to other platforms. Read lengths of up to 1000 bases can be achieved on this platform. Roche provides the current GS FLX+ system as well as a smaller GS junior system. With up to 1 million reads per run, and an average of 700nt per read, 700Mb of sequence data can be achieved in less than 1 day of run time. (Figure 1.5) [10].

1.4.4 ION TORRENT

This newer platform utilizes the adaptor-ligated library followed by sequencing-by-synthesis chemistry of other platforms. However, it has a unique feature, instead of detecting fluorescent signals or photons, it detects changes in the pH of the solution in a well when a nucleotide is added and protons are produced. These changes are miniscule; however, the Ion Torrent device utilizes technologies developed in the semiconductor industry to achieve detectors of sufficient sensitivity and scales that are useful for nucleic acid sequencing. One limitation that has been pointed out is that homo-polymers may be difficult to read as there is no way to stop the addition of only one nucleotide if the same nucleotide is next in the sequence. Ion Torrent produces overall fewer reads than the others in a single run. For example, 60–80M reads at 200 bases per read are possible on the proton instrument in a run producing 10Gb of data. However, the run time is only 2–4h instead of 1–2 weeks on other platforms. The machine has a small footprint, can be powered down when not in use and easily brought back to use, and requires minimal maintenance. With the convenience, size, and speed, it has found sizable applications in microbe sequencing, environmental genomics, and clinical applications where time is critical. This platform is also very popular for amplicon sequencing and use of primer panels for amplicon sequencing developed by specific user communities. Its low-cost and small footprint have also made it attractive to laboratories wishing to have their own personal sequencer [14].Figure 1.6 illustrate Ion Semiconductor sequencing workflow.

1.4.5 PACIFIC BIOSCIENCES

This is a platform representative of the third generation. The chemistry is still similar to second generation sequencing (SGS) as it is a sequencing-by-synthesis system; however, a major difference is that it requires only a single molecule, and reads the added nucleotides in real time. Single-molecule, real-time (SMRT) sequencing developed by Pacific BioSciences offers longer read lengths than the SGS technologies, making it well-suited for unsolved problems in genome, transcriptome, and epigenetics research, particularly assembly and determination of complex genomic regions, gene isoform detection, and methylation detection. [15]

PacBio sequencing captures sequence information during the replication process of the target DNA molecule. The template, called a SMRTbell, is a closed, single-stranded circular DNA that is created by ligating hairpin adaptors to both ends of a target double-stranded DNA (dsDNA) molecule[16] (Figure 1.7). When a sample of SMRTbell is loaded to a chip called a SMRT cell, a SMRTbell diffuses into a sequencing unit called a zero-mode waveguide (ZMW) [17].

SMRT uses zero-mode waveguides (ZMWs) as the basis of their technology. ZMWs are space-restricted chambers that allow guidance of light energy and reagents in the smallest available volume for light detection. In each ZMW, a single polymerase is immobilized at the bottom, which can bind to either hairpin adaptor of the SMRTbell, so a single DNA molecule is sequenced in real time, then start the replication. (Figure 1.8A) [18].Four fluorescent labeled nucleotides, which generate distinct emission spectrums, are added to the SMRT cell and can be detected as a nucleotide chain is being synthesized (Figure 1.8B) [18].

The replication processes in all ZMWs of a SMRT cell are recorded by a ‘‘movie” of light pulses, and the pulses corresponding to each ZMW can be interpreted to be a sequence of bases (called a continuous long read, CLR). Because the SMRTbell forms a closed circle, after the polymerase replicates one strand of the target dsDNA, it can continue incorporating bases of the adapter and then the other strand. If the lifetime of the polymerase is long enough, both strands can be sequenced multiple times (called ‘‘passes”) in a single CLR. [20]

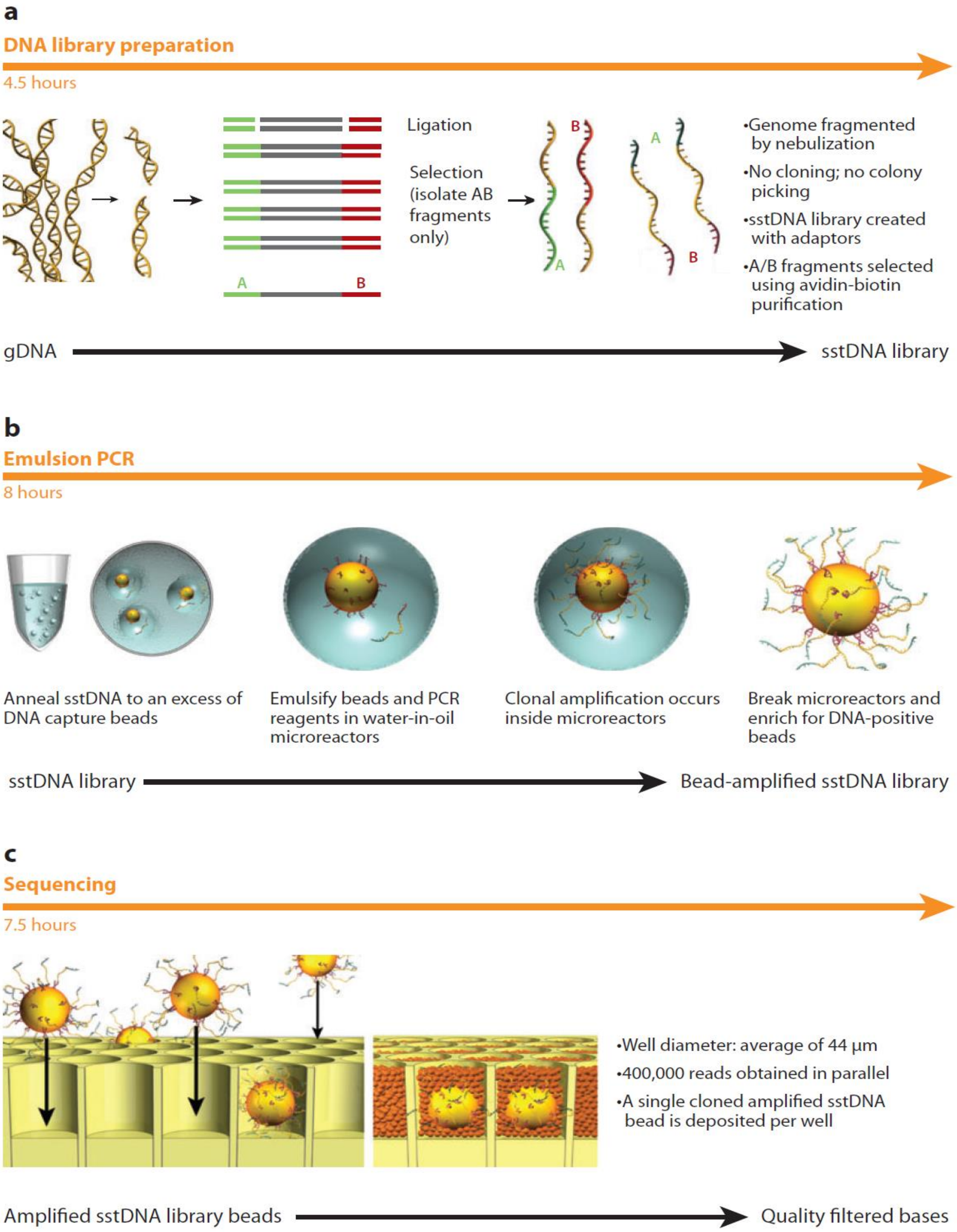
a
DNA library preparation
4.5 hours
Ligation
Selection (isolate AB fragments only)
A
B
•Genome fragmented by nebulization
•No cloning; no colony picking
•sstDNA library created with adaptors
•A/B fragments selected using avidin-biotin purification
gDNA
sstDNA library
b
Emulsion PCR
8 hours
Anneal sstDNA to an excess of DNA capture beads
Emulsify beads and PCR reagents in water-in-oil microreactors
Clonal amplification occurs inside microreactors
Break microreactors and enrich for DNA-positive beads
sstDNA library
Bead-amplified sstDNA library
c
Sequencing
7.5 hours
•Well diameter: average of 44 μm
•400,000 reads obtained in parallel
•A single cloned amplified sstDNA bead is deposited per well
Amplified sstDNA library beads
Quality filtered bases

Figure 1.5: Roche/454 sequencer. The method used by the Roche/454 to amplify single-stranded DNA copies from a fragment library on agarose beads. [10]

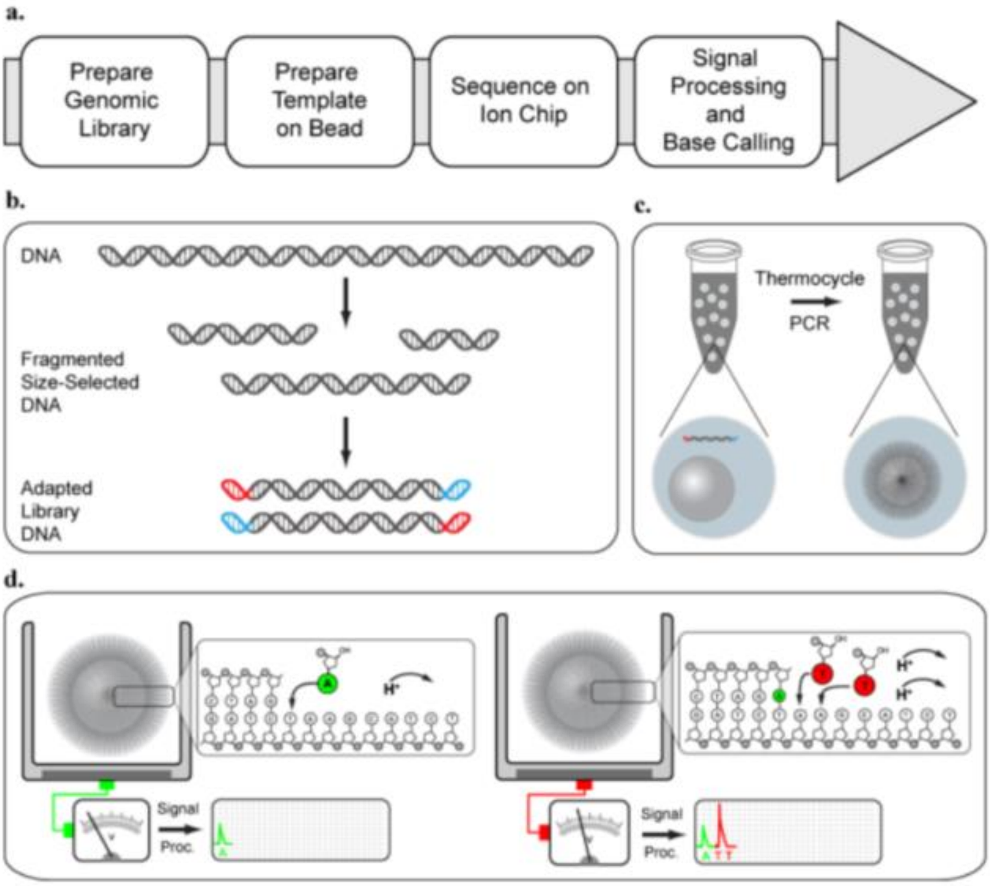

Figure 1.6: Ion Torrent Sequencing Workflow[19]

The latest platform, PacBio RS II, typically produces sequencing movies 0.5–4 h long, up to 250Mb of sequence in a single run, so even throughput is not compromised. The advantage of speed is enormous. Average read lengths can be 5000nt. Improvements in the enzyme and synthesis chemistry can produce routine reads of up to 10,000nt and with longest reads up to 30,000nt. The current version of the instrument called the PacBio RS II can thus produce up to 250Mb of sequence in a single run. As a consequence of direct DNA sequencing of single molecules, it was noticed that nucleic acid modifications such as 5-methyl cytosine caused consistent and reproducible delays in the kinetics of the sequencing DNA polymerase. This has been exploited in the platform to provide sequencing of DNA modifications. Currently, detection of up to 25 base modifications is claimed to be possible on this platform. [21]

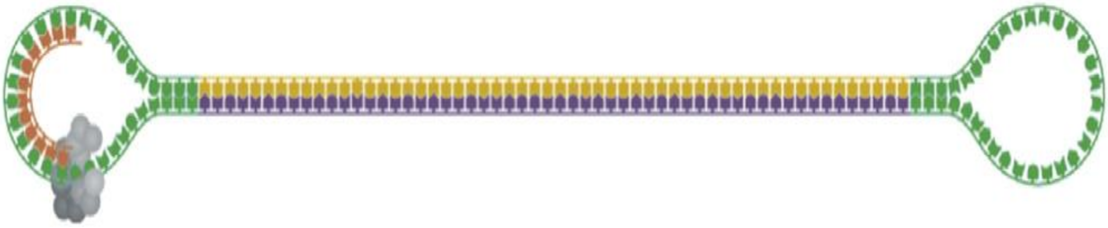

Figure 1.7 Hairpin adaptors. (Green) are ligated to the end of a double stranded DNA molecule. (Yellow and purple), forming a closed circle. The polymerase (gray) is anchored to the bottom of a ZMW and incorporates bases into the read strand (orange). [15]

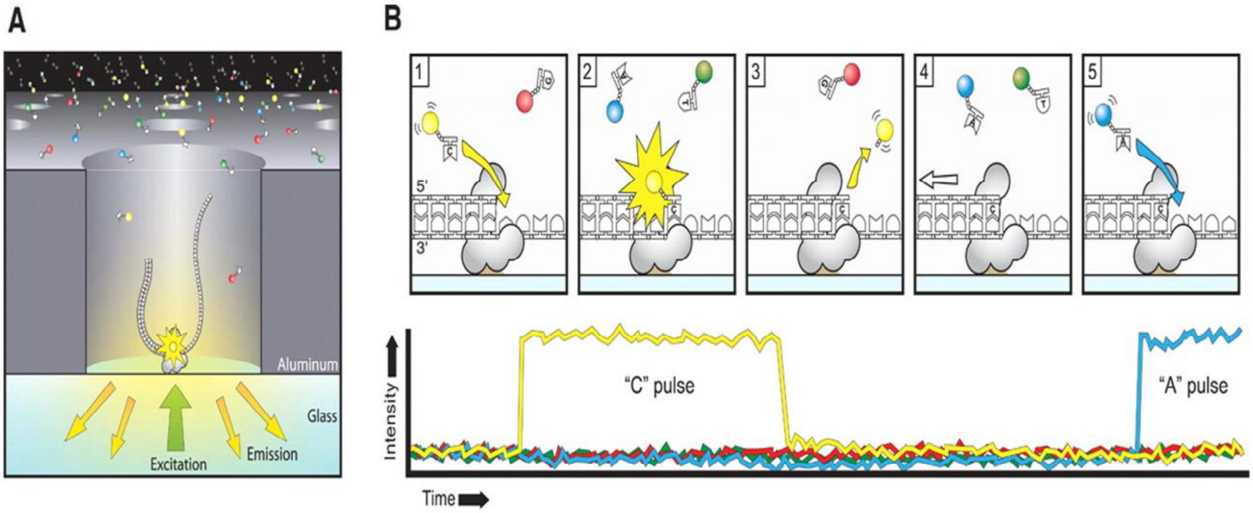

Figure 1.8: PacBio, sequencing via light pulses emission. (A): A SMRTbell (gray) diffuses into a ZMW, and the adaptor binds to a polymerase immobilized at the bottom. (B) Each of the four nucleotides is labeled with a different fluorescent dye, so that they have distinct emission spectrums. As a nucleotide is held in the detection volume by the polymerase, a light pulse is produced that identifies the base. (1) A fluorescently-labeled nucleotide associates with the template in the active site of the polymerase. (2) The fluorescence output of the color corresponding to the incorporated base is elevated. (3) The dye-linker-pyrophosphate product is cleaved from the nucleotide and diffuses out of the ZMW, ending the fluorescence pulse. (4) The polymerase translocates to the next position. (5) The next nucleotide associates with the template in the active site of the polymerase, initiating the next fluorescence pulse, which corresponds to base A here. The figure is adapted from [18] with permission from The American Association for the Advancement of Science.

1.4.6 NANOPORE TECHNOLOGIES

Despite the impressive gains in throughput and low per base cost of current sequencing, efforts continue to improve sequencing technologies. While current nanopore technologies are in development, they so far have had minimal impact on RNA-seq studies. However, their impact in the future may be greater. Nanopore sequencing is a third-generation single-molecule technique where a single enzyme is used to separate a DNA strand and guides it through a protein pore embedded in a membrane. Ions simultaneously pass through the pore to generate an electric current that is measured. The current is sensitive to specific nucleotides passing through the pore, thus A, C, G, or T disturb the current flow differently and produce a signal that is measured in the pore (Figure 1.9). The advantage of this system is its simplicity leading to small-platform device size (as USB stick-sized device), but the system is technically challenging due to the need to measure very small changes in current at single-molecule scale. The efforts to commercialize this technology are led by Oxford Nanopore, however Illumina also has nanopore sequencing under development. Oxford Nanopore technologies are slated to measure directly RNA, DNA, or protein as it passes through a manufactured pore. Although this technology is not widely available at a commercial level, it shows a lot of promise.[22]

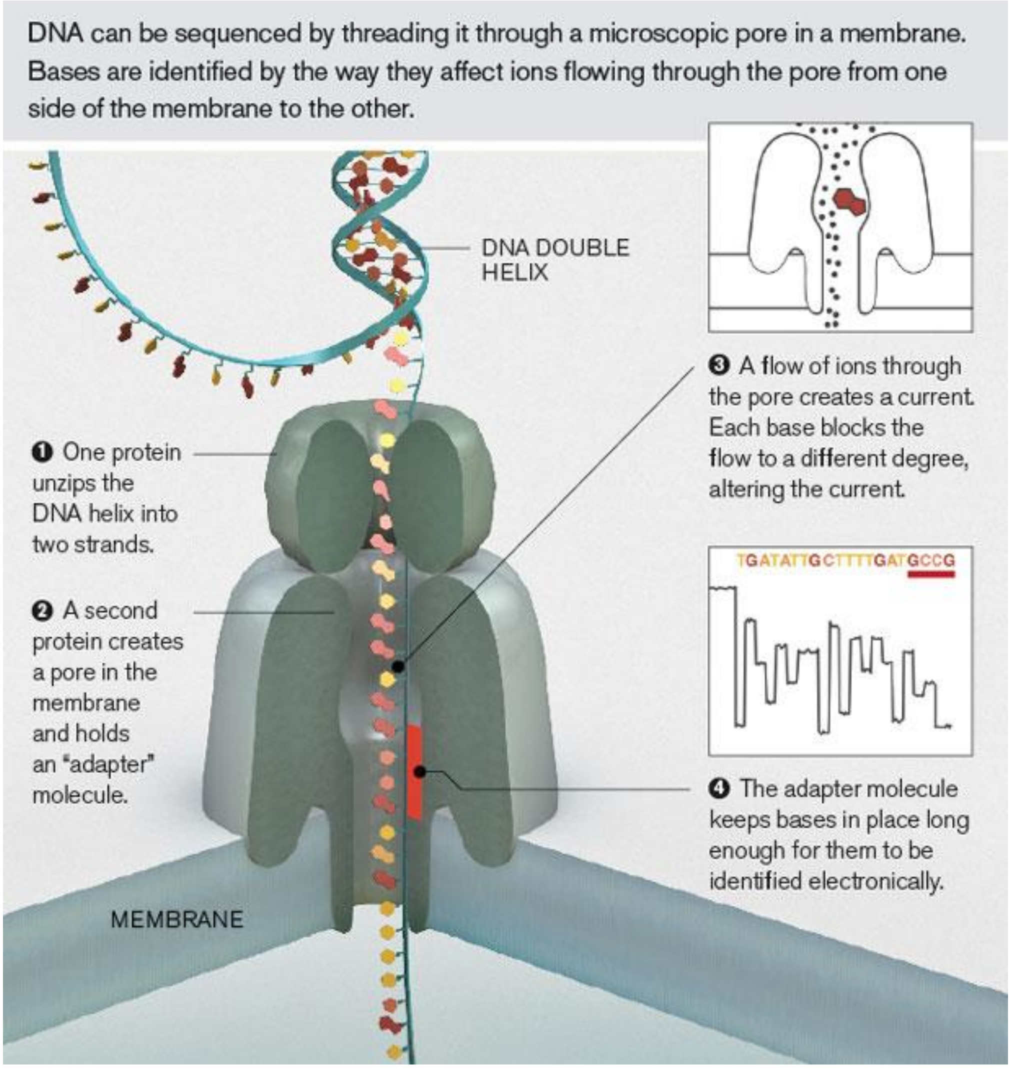
DNA can be sequenced by threading it through a microscopic pore in a membrane. Bases are identified by the way they affect ions flowing through the pore from one side of the membrane to the other.
DNA DOUBLE HELIX
❶ One protein unzips the DNA helix into two strands.
❷ A second protein creates a pore in the membrane and holds an "adapter" molecule.
❸ A flow of ions through the pore creates a current. Each base blocks the flow to a different degree, altering the current.
TGATATTGCTTTTGATGCCG
❹ The adapter molecule keeps bases in place long enough for them to be identified electronically.
MEMBRANE

Figure 1.9: Nanopore Technologies Sequencing workflow [23]

1.5. RNA-SEQ APPLICATIONS

The purposes behind RNA-seq are to identify the sequence, structure, and abundance of RNA molecules in a particular sample. Identifying the structure means the gene structure (i.e., location of promoter, intron–exon junctions, 5′ and 3′ untranslated regions (UTRs), and polyA site). Secondary structure provides the locations of complementary nucleotide that forming stem-loop, or hairpin RNA [24]. Tertiary structure provides the three-dimensional shape of the molecule. However, identifying abundance means, the numerical amounts of each particular sequence both as absolute and normalized values (Figure1.10). Sequence can be used to identify known protein-coding genes, novel genes, or long noncoding RNAs. Once sequence has been determined, folding into secondary structures can reveal the class of molecules such as tRNA or miRNA. Comparison of the abundance of reads for each RNA species can be made between samples derived from different developmental stages, body parts, or across closely related species. [2]

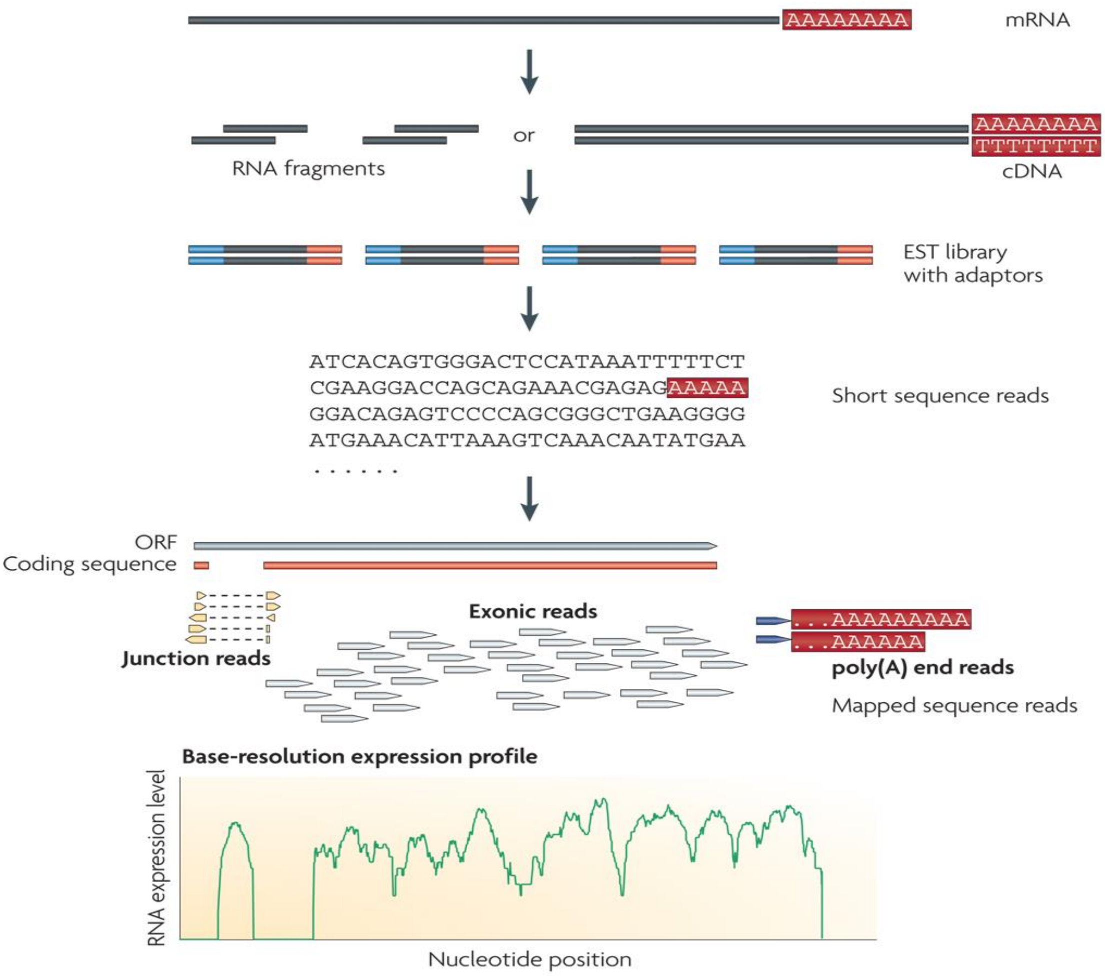

Figure1.10: A typical RNA-Seq experiment. Long RNAs are first converted into a library of cDNA fragments Sequencing adaptors are subsequently added to each cDNA fragment and a short sequence is obtained from each cDNA using high-throughput sequencing technology. The resulting sequence reads are aligned with the reference genome or transcriptome, and classified as three types: exonic reads, junction reads and poly(A) end-reads. These three types are used to generate a base-resolution expression profile for each gene. [2]

In the following is presented the common applications of using RNA-seq data.

1.5.1 PROTEIN CODING GENE STRUCTURE

Earlier transcriptomic methods such as microarray expression analysis, cloning and Sanger sequencing of cDNA libraries, and serial analysis of gene expression (SAGE), as well as computational prediction from genomic sequences, have already provides gene structures. These structure annotations have been archived in databases and provide an easily accessible source for comparing raw RNA-seq data with known protein coding genes. The first important step is to map the RNA-seq reads to known protein-coding genes.

Furthermore, RNA-seq data analysis can be used for confirming exon–intron boundaries, as well as the existence of completely novel exons. Therefore, using RNA-seq can define what is called a gene model, which is a collection of exons and introns that make up a gene. Since RNA-seq is quantitative, it can also specify within a sample the alternative exons usage: for example, when a specific exon is used five times more often than another one. The 5′ transcription start site (TSS) can be identified precisely using RNA-seq data. Similarly, at the 3′ end of the molecule, the 3′UTR can be identified as well, such that the site of polyadenylation can be observed in the RNA-seq reads. Alternative polyadenylation sites can also be observed in the same way as alternative TSS as well as their respective abundances. As RNA-seq is massively parallel, sufficient reads will permit these gene structures and their alternatives to be mapped for presumably every protein-coding gene in a genome. Thus, RNA-seq can provide the 5′TSS, 5′UTR, exon–intron boundaries, 3′UTR, polyadenylation site, and alternative usage of any of these if applicable[25]. A simplified scheme of gene structure illustrated in Figure 1.11. Figure 1.12 shows analysis protocol of gene annotation from RNA-seq data[26].

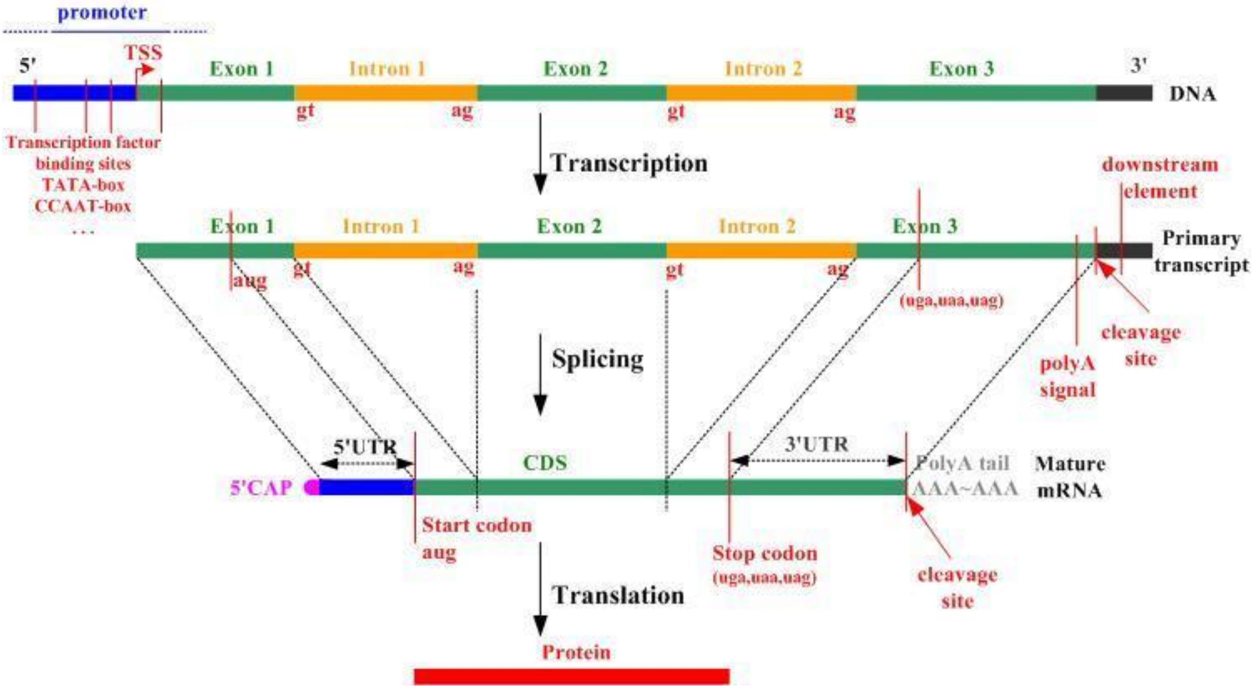

Figure 1.11 Schematic gene structure and simplified transcription, splicing and translation process[27]

1.5.2 NOVEL PROTEIN-CODING GENES

Previously, the annotations of protein-coding genes relied on computational predictions based on genomic sequences. This was fine as long as genome data were available, the gene model elements fit common expected size and distance parameters, and there were transcriptomic data in the form of expressed sequence tag (EST) data sets or orthology data available to verify the predictions. However, it was easy to see that these criteria fit well only a very limited number of organisms under scientific investigation. Therefore, RNA-seq, with its high throughput, could verify many of the previous predictions, but also in cases where no prediction existed, it could identify novel protein-coding genes. It was especially useful in cases where no genome sequence was available, so a transcriptome of an organism could be built entirely from RNA-seq data. A recent example of this application has been in the sequencing of the giant panda genome [28].

1.5.3 QUANTIFYING AND COMPARING GENE EXPRESSION

Once the sequence and gene structure have been elucidated, it is logical that abundance values can be attributed to each gene as well as various features in their structures. As many studies would like to compare the abundance of RNA transcripts from healthy versus sick, nontreated versus treated, or time point 0 versus 1, it is logical that comparative studies are made. The range and types of comparative studies are virtually unlimited. In one of the earliest RNA-seq studies, transcripts from adult mouse brain, liver, and skeletal muscle were sequenced and compared [25].

More than 40M single-end reads at 25nt were sequenced on an Illumina platform and the authors found novel TSSs, alternative exons, and alternative 3′UTRs. The study demonstrated the shallowness of previous annotations of gene structure and thus highlighted how the breadth and depth of annotations provided by RNA-seq technology could change our view of gene structure. These results thus paved the way for subsequent RNA-seq studies. Another RNA-seq study two years later, followed the expression of RNA transcripts from mouse skeletal muscle cells during differentiation after 60 h, 5 days, or 7 days [29]. The technology improved so that more than 430M paired-end reads at 75nt were used to identify greater than 3700 previously unannotated transcripts. TSSs were also shown to change in more than 300 genes during differentiation. It is also possible to study RNA transcripts in whole animals. The total RNA from whole animals could be isolated and subjected to RNA-seq, in recent study Over 30M reads from water- or ethanol-treated animals were obtained [30]. Ethanol exposure could be seen to increase RNA transcripts of detoxification enzyme genes and decrease transcripts involved in endoplasmic reticulum stress.

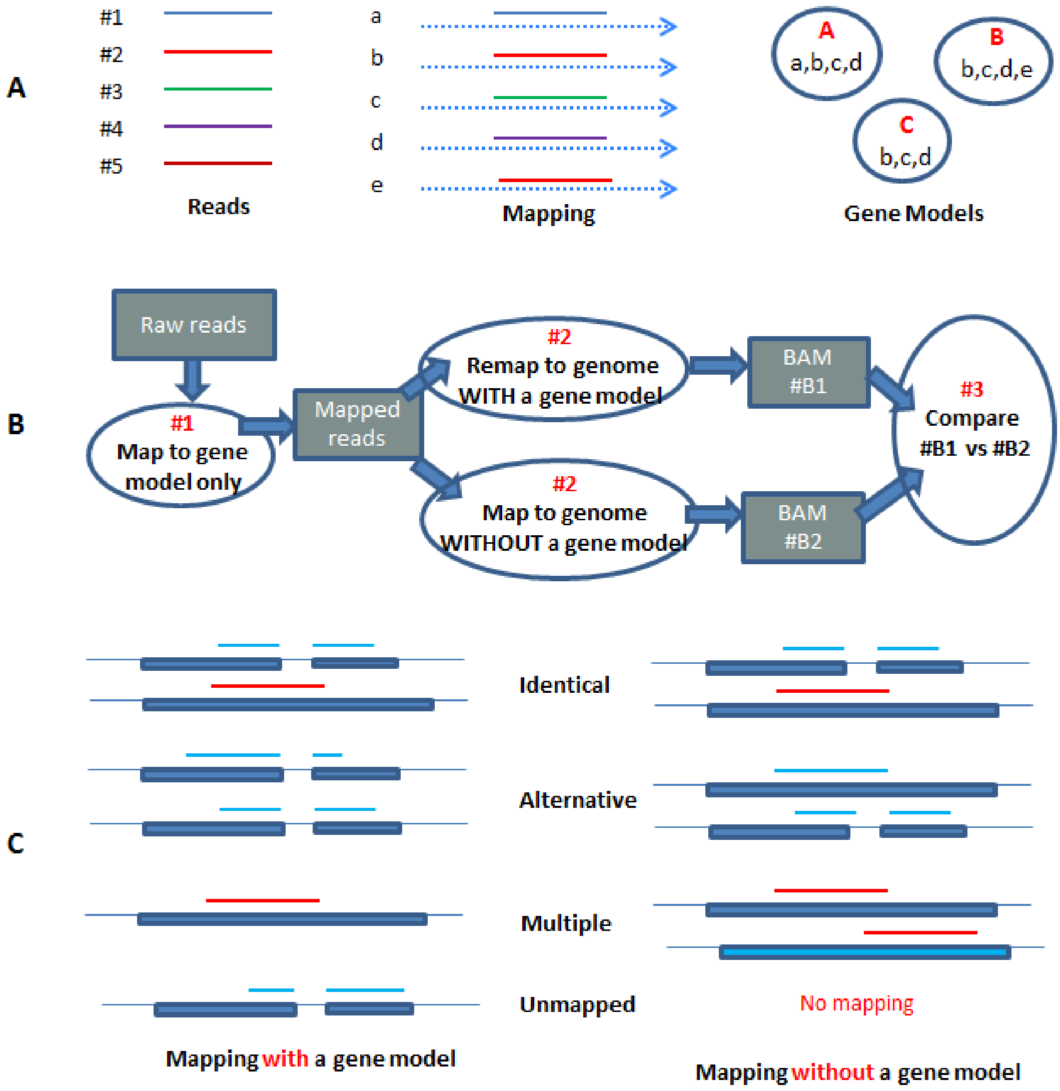

Figure 1.12: Analysis protocol of gene annotation from RNA-seq data. (A) The mapping result for a sequence read is gene model dependent; (B) “two-stage” mapping protocol: at Stage #1, all RNA-Seq reads are mapped to a reference transcriptome; at Stage #2, the mapped reads at Stage #1 are re-mapped to the genome with and without the use of a gene model, respectively; (C) the protocol for classifying uniquely mapped sequence reads into four categories, i.e., “Identical”, “Alternative”, “Multiple” and “Unmapped” (or Fail).[26]

1.5.4 EXPRESSION QUANTITATIVE TRAIT LOCI (EQTL)

RNA-seq studies have become so pervasive that they have been used to study quantitative traits, especially in the context of genome variation One of the most prominent directions One of the most prominent directions One of the most prominent directions has been the extensive set of studies on expression quantitative trait loci (eQTLs), namely, the discovery of genetic variants that explain variation in gene expression. Such studies have offered promise not just for the characterization of functional sequence variation but also for the understanding of basic processes of gene regulation and interpretation of genome-wide association studies. An eQTL is a locus that explains a fraction of the genetic variance of a gene expression phenotype [31]. Traditionally, quantitative traits loci studies in the form of genome-wide association studies have linked SNPs with a quantitative trait such as height, weight, cholesterol level, or risk to obtain type II diabetes. eQTL provides gene expression changes that can be correlated with known SNPs [32]. The basis for this correlation can be a local action, called cis-eQTL, for example, where an SNP is located on an enhancer region and changes the expression, or a distal action, called trans-eQTL, for example, where an SNP changes the structure of a transcription factor that no longer works on its target gene. Thus, gene expression levels, as determined by RNA-seq, can provide a link with phenotype through its correlation with SNPs. An extension of this idea has been to correlate also gene-splicing sites and usage with SNPs. This approach, termed eQTL, suggests splicing as playing a significant role in regulating overall gene expression [33]. In addition to human disease research, this approach has been applied in traditional fields such as plant breeding where quantitative traits are important.

1.5.5 SINGLE-CELL RNA-SEQ

RNA-seq is a variation of RNA-seq where the source of total RNA for sequencing comes from a single cell. Typically, total RNA is not isolated, but rather cells are individually harvested from their source and reverse-transcribed. Methodology for library preparation is similar to RNA-seq: RNA is reverse-transcribed to cDNA, adaptors are ligated, barcodes for each cell are added, and ds cDNA amplified. Due to the low complexity of RNA species, single isolated cells or individual libraries are sometimes pooled prior to sequencing. In one example of this approach, a single mouse blastomere was collected and RNA-sequenced from its contents. The authors found 5000 genes expressed and >1700 novel alternative splice junctions, indicating both the robustness of the approach as well as the complexity of splicing in a single cell [34]. In another example of the approach, single cells from the nematode C. elegans at an early multicell developmental stage were isolated and libraries prepared from total RNAs. New transcription of genes could be monitored at each individual stage of development via profiling the transcripts of individual cells [35].

1.5.6 FUSION GENES

As read numbers and lengths increased, and paired-end sequencing became available, the ability to identify rare, but potentially important transcripts increased. Such is the case with fusion genes, which are transcripts generated from the fusion of two previously separate gene structures. Fusion partners can contribute 5′UTRs, coding regions, and 3′polyadenylation signals. Conditions for this event to occur happen during genomic rearrangement found in cancer tissues and cells. Cytogenetic derangements such as genomic amplifications, translocations, and deletions can bring together two independent gene structures. For example, 24 novel and three known fusion genes were detected in three breast cancer cell lines using paired-end sequencing of libraries sized 100 or 200nt in length [36]. One of these fusion genes, *VAPB-IKZF3,* was found to be functional in cell growth assays[37]. Recent RNA-seq studies have found fusion genes to be present in normal tissue, suggesting that fusion gene events might have normal biological function as well[38].

1.5.7 GENE VARIATIONS

As the amount of RNA-seq data accumulates, it is possible to mine the data for gene variation. Mostly bioinformatic approaches by downloading publicly available data have been used to scan SNPs in transcriptomic data [39]. In this study, 89% of SNPs derived from RNA-seq data at a coverage of 10× were found to be true variants. SNP detection can also be obtained directly from original RNA-seq data. A group performed RNA-seq on muscle from *Longissimus thoraci* (Limousine cattle) muscle mRNAs [40]. They were able to identify >8000 high-quality SNPs from >30M paired-end reads. A subset of these SNPs was used to genotype nine major cattle breeds used in France, demonstrating the utility of this approach.

One recent application of NGS has been to identify variations in the protein coding gene sequences from genomic DNA samples. Termed "exome-sequencing or exome-capture," this approach is technically not RNA-seq since it relies on sequencing fragmented genomic DNA that has been enriched for exons via hybridization to exonic sequences. This has been motivated by human disease studies, where variations, typically SNPs, need to be identified from a large cohort of individuals. As exons are overwhelmingly located in protein-coding genes, this has the advantage of finding variations that have direct effects on protein structure. It is one of the most popular applications of NGS and many commercially available kits have been developed for this purpose. [41]

1.5.8 LONG NONCODING RNAS

Another application of RNA-seq has been to find transcripts that are present, but do not code genes. Long noncoding RNAs (lncRNAs) were known before RNA-seq technologies were available. However, the extent of their existence and pervasiveness was not fully appreciated until RNA-seq methods were able to uncover the many different species of lncRNAs in living cells. lncRNAs are generally described as transcripts that fall outside of known noncoding RNAs such as tRNAs, ribosomal RNAs, and small RNAs, do not overlap a protein-coding exon, and are >200nt in length [42]. lncRNAs can control transcription as enhancers (eRNA) epigenetically by binding and altering the function of histone proteins, as competitors to RNA-processing machinery [competitive endogenous RNA (ceRNA)], or as noise generated randomly. It can now be appreciated that lncRNAs may play a role in disease such as Alzheimer's disease [43].

1.5.9 SMALL NONCODING RNAS (MIRNA-SEQ)

RNA-seq can be used to identify the sequence, structure, function, and abundance of small noncoding RNAs. The most well-known example of these being miRNAs (miRNA-seq), but other small noncoding RNAs such as small nucleolar RNAs (snRNA), microRNA offset RNAs (moRNAs), and endogenous silencing RNAs (endo-siRNAs) can also be studied using miRNA-seq approaches. The methods used for miRNA-seq are similar to RNA-seq. The starting materials can be total RNA or size-selected/fractionated small RNAs. Most of the common sequencing platforms will sequence small RNAs once converted into ds cDNAs, such that much of the difference in the experimental protocols occur before sequencing. Figure 1.13 shows how the non-coding exon can form the transcriptional landscape of the genome [44]. There are many applications for characterizing these molecules not only in the studies of basic biochemistry, physiology, genetics, and evolutionary biology, but also in medicine as a diagnostic tool for cancer or in aging processes. A recent study of the nematode Panagrellus redivivus has presented the identification of >200 novel miRNAs and their precursor hairpin sequences while also providing gene structure models, annotation of the protein-coding genes, and the genomic sequences in a single publication [45].

1.5.10AMPLIFICATION PRODUCT SEQUENCING (AMPLI-SEQ)

It is sometimes the case that whole transcriptomes do not need to be sequenced, but only a small number of genes. While one can always obtain a subset of genes of interest from a whole transcriptome sequence analysis, the effort, time, and resources required may be more than necessary. By using a panel of PCR primers consisting of 10–200 pairs, one can perform reverse transcription-PCR (RT-PCR) and instead of cloning each individual product and isolating plasmid DNA for Sanger sequencing, one can sequence the pool of PCR products to obtain the sequence. This has practical applications where the number of samples to be interrogated is large, and the number of genes is small [46].

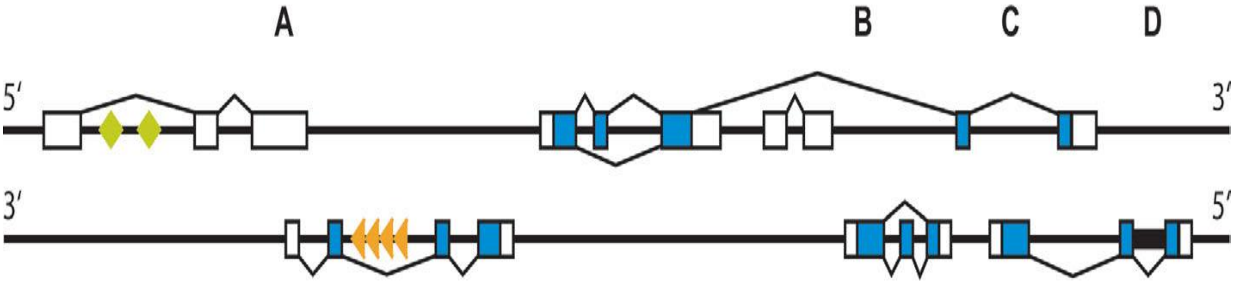

Figure 1.13: The complexity of the transcriptional landscape in mammals. White boxes represent non-coding exonic sequences and dark blue boxes protein-coding exonic sequences. Green diamonds represent snRNAs and orange triangles represent miRNAs. Indicated are (**A**) antisense transcripts with overlapping exons, (**B**) nested transcripts on both strands, (**C**) antisense transcripts with interlacing exons and (**D**) retained introns.[44]

1.6. OSTEOBLAST CELLS DIFFERENTIATION

Skeletal component cells including osteoblasts, chondrocytes, adipocytes, myoblasts, tendon cells, and fibroblasts, are derived from mesenchymal stem cells [47]. while Osteoclast is a hematopoietic cell derived from CFU-GM (colony forming unit- granulocyte, monocyte) and branches from the monocyte-macrophage lineage early during the differentiation process [48].

Bone is constructed through 3 processes: osteogenesis, modeling, and remodeling. It is constantly being remodeled in a dynamic process where osteoblasts are responsible for bone formation (or ossification), and osteoclasts for its resorption. Osteoblast and osteoclast work in tight cooperation, and together constituting a “bone multicellular unit”[49]. Fine tuning of this system is crucial for the development of bones, for repairing fractures, and for the correct maintenance of the skeleton throughout life. The Figure 1.14 shows the relationship between osteoclasts, osteoblasts, and growth factors[50].

Bone resorption is the process by which osteoclasts decompose the bone tissue for maintenance, repair, and remodeling. The Osteoclast carve out the shape to fit the physical environment (modeling) and adjust it to the demands of the body growth and the changing circumstances (remodeling). It disassembles and digests the composite of hydrated protein and mineral at a molecular level by secreting acid and a collagenase. This process also helps regulate the level of blood calcium, and release the minerals, resulting in a transfer of calcium from bone tissue to the blood [51].

Osteoblasts have also a role in the regulation of bone resorption through Receptor Activator of Nuclear-factor Kb Ligand (RANKL), that links to its receptor RANK on the surface of pre-osteoclast cells, inducing their differentiation and fusion. Furthermore, osteoblasts secrete a soluble decoy receptor (osteo-protegerin, OPG) that blocks RANK/RANKL interaction by binding to RANKL and, thus, prevents osteoclast differentiation and activation. Therefore, the balance between RANKL and OPG determines the formation and activity of osteoclasts. Another factor can influence bone mass is leptin, a hormone produced by adipocytes that have a dual effect. It can act through the central nervous system and reduce osteoblasts activity, or can have an osteogenic effect by binding directly to its receptors on the surface of osteoblast cells[52].Figure 1.16 depicts a schematic osteoclast differentiation and regulation process [54].

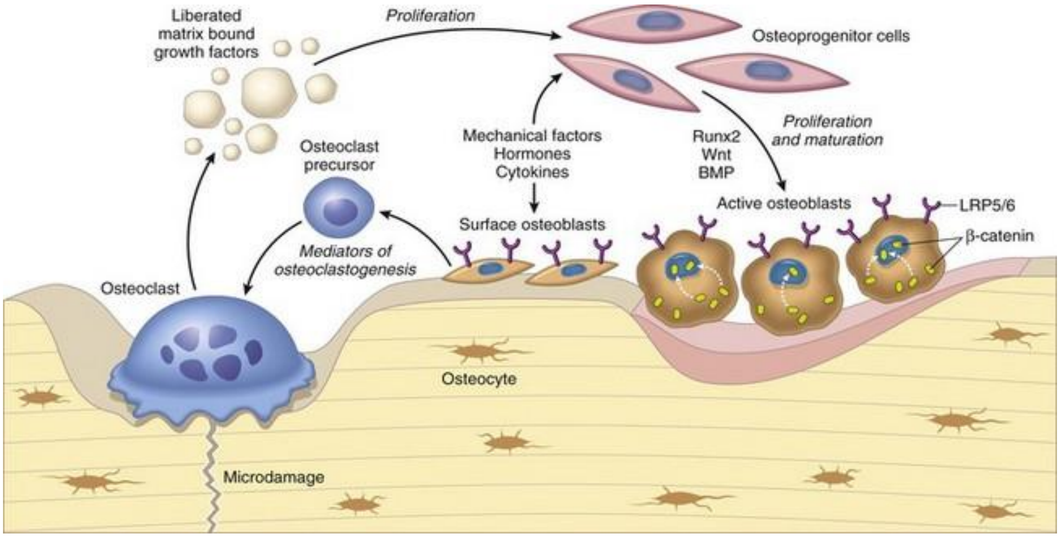

Figure 1.14: Osteogenesis, modeling, and remodeling. Schematic diagram of the relationship between osteoclasts, osteoblasts, and growth factors.[50]

Ossification is a regulated process for generating new bone material, performed by specialized cells called osteoblasts, that synthetize the bone extracellular matrix (osteogenesis). Osteoblasts are bone-building cells of mesenchymal origin; they differentiate from mesenchymal progenitors, either directly or via an osteo-chondro-progenitor. The direct pathway is typical for intramembranous ossification of the skull and clavicles, while the latter is an indication of endochondral ossification of the axial skeleton and limbs. The pathways merge at the level of pre-osteoblasts. Osteoblasts can also differentiate into osteocytes, which are stellate cells populating narrow interconnecting passages within the bone matrix [52].In the Figure 1.15 representation of biogenesis of osteoblasts.[53].

Osteoblasts are present throughout life; however, their highest activity is during embryonic skeletal formation and growth. In an adult organism, osteoblasts are activated when there is need to regenerate a defect or when the bone matrix has been depleted[55]. Dysregulation of this process may cause inadequate or excessive mineralization of bones or ectopic calcification, all of which have grave consequences for human health [53].A therapeutic boost to the osteoblast activity could potentially prolong millions of human lives. Attempts have been made to transplant autologous stem cells to bone defects [56].

Bone formation requires differentiated and active osteoblasts to synthesize the extracellular matrix that will support the mineralizing process, However Osteoblasts do not produce bone material immediately from the moment they are mature, it takes about 4 months until the synthesis of bone matrix by the cell is detected[57].

The osteoblasts that have encircled themselves with the bone matrix, eventually differentiate into osteocytes, which are interconnected stellar cells that regulate the turnover of bone material. Osteoblasts that remain on the surface of bone covering by the periosteum, have two destinies, either to become inert bone-lining cells, or undergo apoptosis. When the mature osteoblast population grows thin (either as a result if a natural turnover or a regenerative process demanding massive recruitment), new osteoblasts are differentiated from mesenchymal progenitor cells; however, their resource is limited [58].

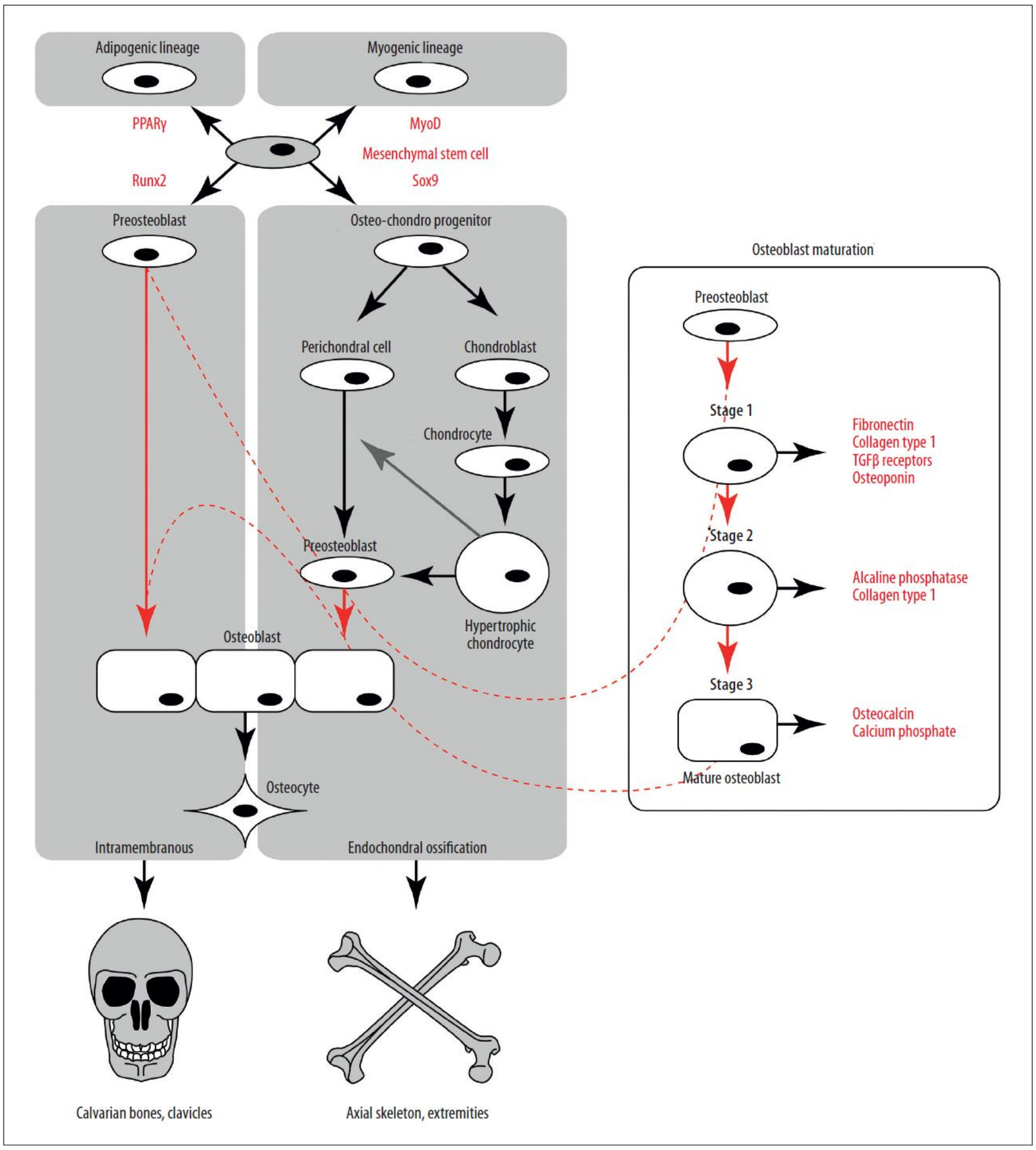

Figure 1.15 : A flowchart of the biogenesis of osteoblasts. Mesenchymal stem cells can differentiate to 4 lineages (top left) by expressing corresponding transcriptional regulators: PPARg for adipogenic, MyoD for myogenic, Runx2 for osteoblastic, and Sox9 for chondrocytic lineages. In intramembranous ossification (osteogenesis in the scull and clavicles), pre-osteoblasts are generated directly by mesenchymal stem cells, while in endochondral (osteogenesis of the axial skeleton and the limbs) a common osteo-chondro-progenitor gives rise to both cell types. Hypertrophic chondrocytes in a paracrine manner (gray arrow) regulate transformation of perichondral cells into pre-osteoblasts, or might itself transform into one. The process of maturation of pre-osteoblasts is shown in the enlargement on the right[53].

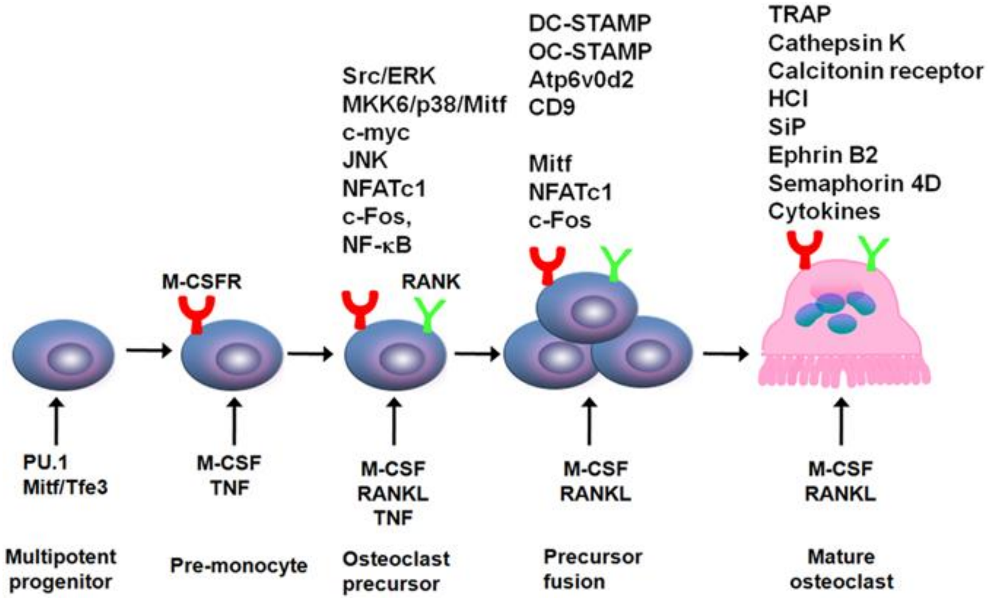

Figure 1.16: Regulation of osteoclast formation and differentiation. [54]

Osteoblasts secrete and mineralize the bone matrix. The mineralized extracellular matrix is mainly composed of type I collagen and smaller but significant amounts of osteocalcin (OC), matrix gla protein, osteopontin (OPN), bone sialoprotein (BSP), BMPs, TGF-β, and the inorganic mineral hydroxyl apatite. [52]

Osteoblast differentiation can be characterized in three stages[53]:

a) Cell proliferation
b) Matrix maturation
c) Matrix mineralization

In Stage 1 the cells continue to proliferate and express fibronectin, collagen, TGFb receptor 1, and osteopontin. In Stage 2 they exit the cell cycle and start differentiating, while maturating the extracellular matrix with Alp and collagen. In Stage 3 matrix mineralization occurs when the organic scaffold is enriched with osteocalcin, which promotes deposition of mineral substance. Osteocalcin is in fact the second most abundant protein in bone after collagen [59]. At this stage the osteoblast assumes its characteristic cuboidal shape [58].

1.7. ALTERNATIVE SPILCING

Genetic information of an organism is stored in the genes, this information is transcribed from DNA into a messenger RNA (mRNA) template by a process called transcription. However, in eukaryotes, before the mRNA can be translated into proteins, non-coding portions of the sequence, called introns, must be removed and protein-coding parts, called exons, joined by RNA splicing to produce a mature mRNA. Recent estimates indicate that the expression of nearly 95% of human multi-exon genes involves alternative splicing.[60]

Alternative splicing of precursor mRNA is an essential mechanism for gene regulation and for generating proteomic diversity, it produces different protein products that function in diverse cellular processes, including cell growth, differentiation, and organism development. Furthermore, it has a largely hidden function in quantitative gene control, by targeting RNAs for nonsense-mediated decay. In the Figure 1.18 illustrate the general concept of alternative splicing.

Regulation of alternative splicing is a complicated process in which numerous interacting components are involved. Additional molecular features, such as chromatin structure, RNA structure and alternative transcription initiation or alternative transcription termination, collaborate with these basic components to generate the protein diversity due to alternative splicing.

Splicing is carried out by the spliceosome, a massive structure in which five small nuclear ribonucleoprotein particles (snRNPs) (U1, U2, U4, U5 and U6), that are associated with a large number of auxiliary proteins cooperate to accurately recognize the splice sites and catalyze the steps of the splicing reaction. The auxiliary elements known as Exon Splicing Enhancers (*ESEs*), and Intron Splicing Enhancers (*ISEs*), in addition to Exon Splicing Silencers (*ESSs*), and Intron Splicing Silencers (*ISSs*)[61]. These auxiliary elements are involved in defining both constitutive and alternative exons.

The splicing process occurs in cellular machines called spliceosomes, in which the snRNPs are found along with additional proteins. The primary variety of spliceosome is one of the most plentiful structures in the cell.

Spliceosome assembly can be abbreviated with the following steps[62]:

1- The positions and sequences of the consensus cis-acting elements help to define the splice sites on Pre-mRNA. The splice donor site(SDS) includes an almost invariant sequence GU at the 5' end of the intron.
A branch point site is located anywhere from 18 to 40 nucleotides upstream from the 3' end of the intron which contains A (A typical sequence of the branch point is YNYYRAY, where Y indicates a pyrimidine, N denotes any nucleotide, R denotes any purine, and A denotes adenine).
The splice acceptor site (SAS) at the 3' end of the intron terminates the intron with an almost invariant AG sequence. Upstream (5'-ward) from the AG there is a region high in pyrimidines (C and U), called polypyrimidine tract.

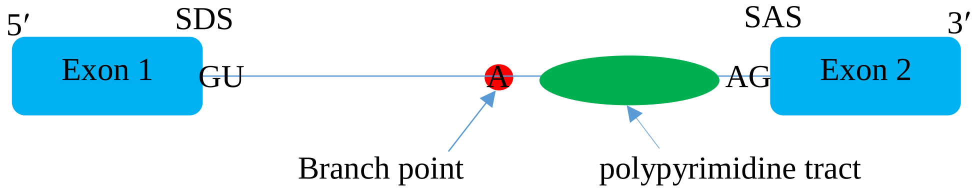

Figure 1.17 Splicing sites and cis-acting elements

2- Slicing begins with the recognition of the 5′ splice site by the snRNP **U1** and the binding of splicing factor 1 (**SF1**) to the branch point. The U2 auxiliary factor (**U2AF**), a dimer of 65 and 35 kDa subunits binds to the polypyrimidine tract, and 3′ terminal of AG respectively. and SR proteins (**SRp**) bind to ESEs and contact **U2AF**, U1 snRNP and the branch point. This assembly is ATP independent and results in the formation of the E complex.

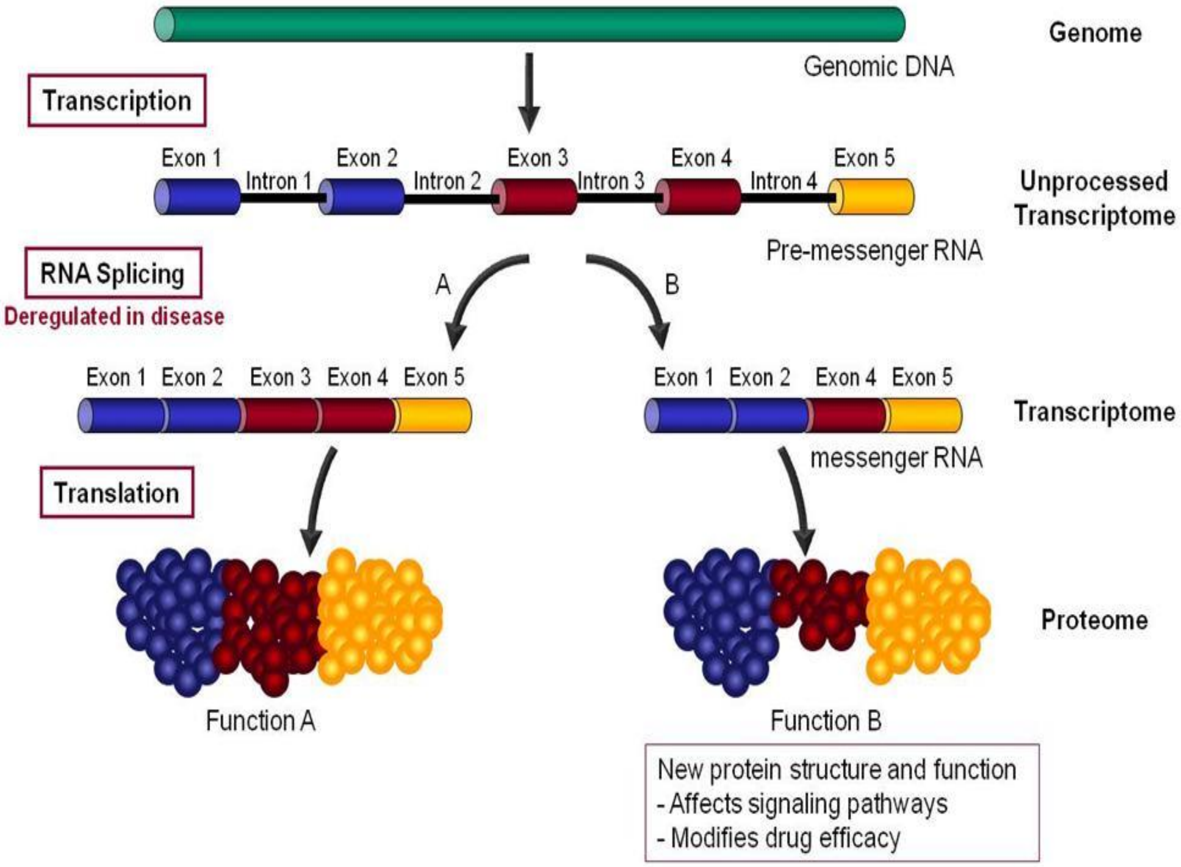

Figure 1.18 Genetic scheme of general concept of RNA alternative splicing [63].

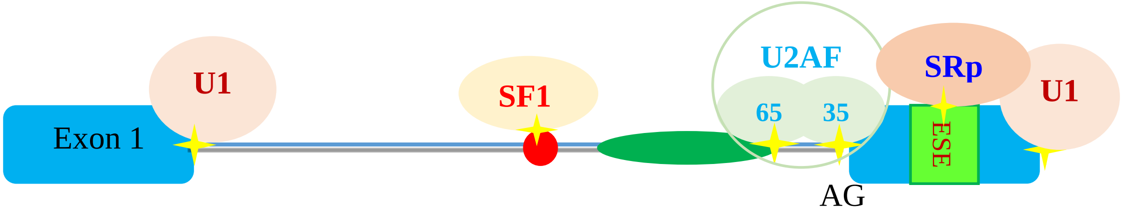

Figure 1.19: E complex, binding snRNP U1, U2AF and SF1

3- Replacing of SF1 by the snRNPU2at the branch point. This is A complex which is ATP dependent. Formation of the A complex is usually the key step in determining the ends of the intron to be spliced out, and defining the ends of the exon to be retained.

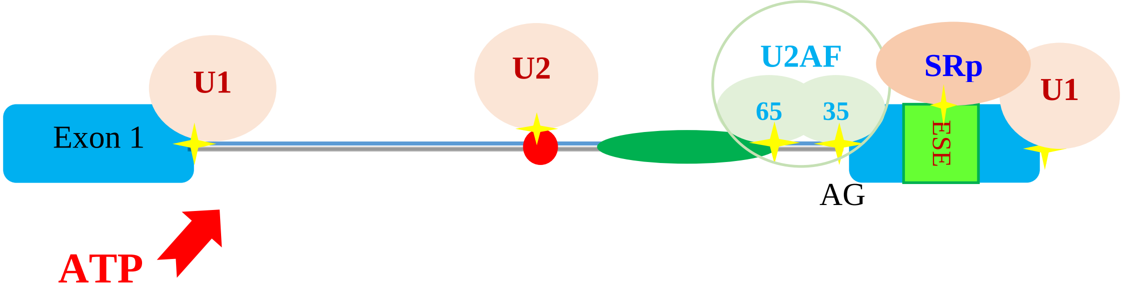

Figure 1.20: A complex, replace SF1 with snRNP U2

4- Complex B (pre-catalytic spliceosome): The U5, U4/U6 snRNP trimer binds, completing the spliceosome assembly, and the U5 snRNP binds exons at the 5' site, with U6 binding to U2.

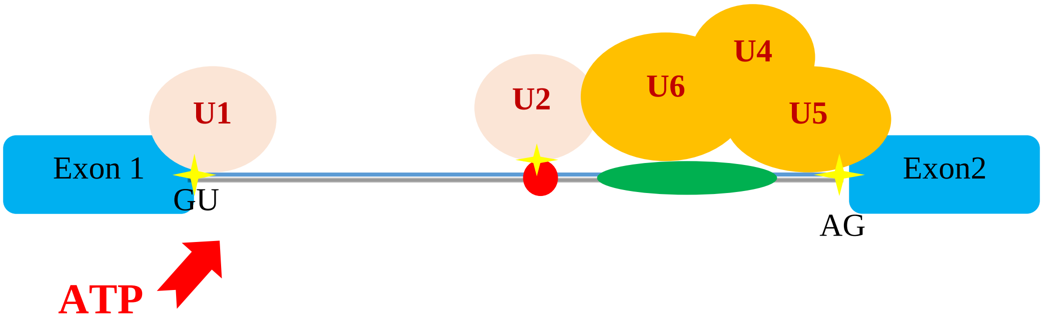

Figure 1.21: B complex, U4,U5,U6 snRNPs binding

5- Complex C (catalytic spliceosome): The 5′ end of the intron that attached to snRNP U1 is cleaved. The cut end then attaches to the conserved branch point region downstream through pairing of Guanine (from the splice site 5′ end) and Adenine (on the branch point) nucleotides, to form a looped structure known as a *lariat*. The bonding of the guanine and adenine bases takes place via a chemical reaction known as transesterification, in which a hydroxyl (OH) group on a carbon atom of the adenine "attacks" the bond of the guanine nucleotide at the splice site. the snRNPs U2 and U4/U6 appear to contribute to positioning of the 5′ end and the branch point in proximity.

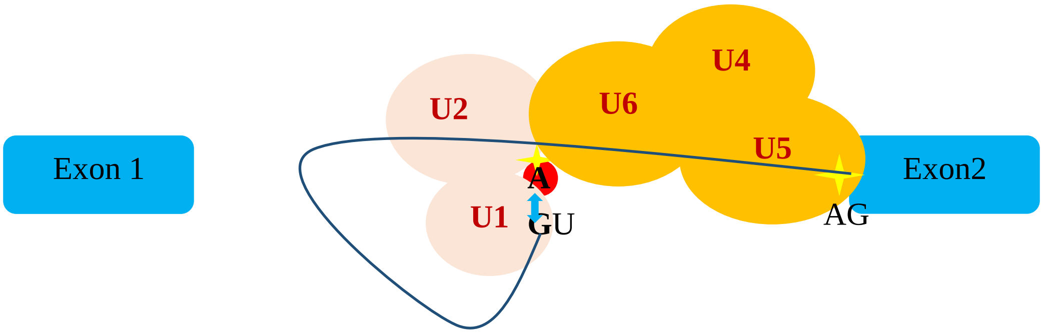

Figure 1.22: Forming lariat by A-G bonding

6- The U1 and U4 snRNPs are released and U5 and U6 snRNPs shift positions. U2 keeps connected to the lariat formed by A-G pairing

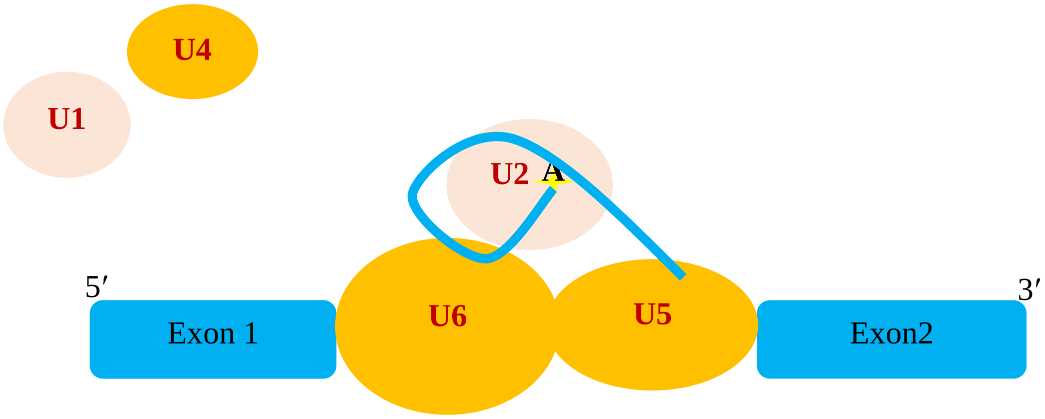

Figure 1.23: U1 and U4 snRNPs are released and U5 and U6 snRNPs shift positions

7- This step occurs by transesterification; in this case, an OH group at the 3′ end of the exon attacks the phosphodiester bond at the 3′ splice site. The adjoining exons are covalently bound, and the resulting lariat is released with U2, U5, and U6 bound to it.

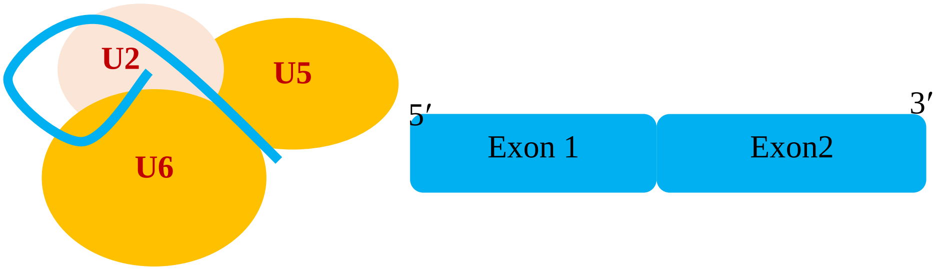

Figure 1.24 Splicing process final result

In addition to consensus sequences at their splice sites, eukaryotic genes with long introns also contain exonic splicing enhancers (ESEs). These sequences, which help position the splicing apparatus, are found in the exons of genes and bind proteins that help recruit splicing machinery to the correct site. Most splicing occurs between exons on a single RNA transcript, but occasionally trans-splicing occurs, in which exons on different pre-mRNAs are ligated together.

Five basic modes of alternative splicing are generally recognized.[60]

- *Exons skipping*: Discrete exons that can be independently included or excluded from the mRNA. This is the most common mode in mammalian pre-mRNAs.[64]
- *Mutually exclusive exons:* One of two exons is retained in mRNAs after splicing, but not both.
- *Alternative donor site:* An alternative 5' splice junction (donor site) is used, changing the 3' boundary of the upstream exon.
- *Alternative acceptor site*: An alternative 3' splice junction (acceptor site) is used, changing the 5' boundary of the downstream exon.
- *Intron retention:* A sequence may be spliced out as an intron or simply retained. This is distinguished from exon skipping because the retained sequence is not flanked by introns. If the retained intron is in the coding region, the intron must encode amino acids in frame

with the neighboring exons, or a stop codon or a shift in the reading frame will cause the protein to be non-functional. This is the rarest mode in mammals.[64]

In addition to these primary modes of alternative splicing, there are two other main mechanisms by which different mRNAs may be generated from the same gene; multiple promoters and multiple polyadenylation sites. Use of multiple promoters is properly described as a transcriptional regulation mechanism rather than alternative splicing; by starting transcription at different points, transcripts with different 5'-most exons can be generated. At the other end, multiple polyadenylation sites provide different 3' end points for the transcript. Both of these mechanisms are found in combination with alternative splicing and provide additional variety in mRNAs derived from a gene [60][62].

The following Figure 1.25 illustrates the basic alternative splicing events, whereas a number of auxiliary elements can influence alternative splicing. These are categorized by their location and activity as exon splicing enhancers and silencers (ESEs and ESSs) and intron splicing enhancers and silencers (ISEs and ISSs). Enhancers can activate adjacent splice sites or antagonize silencers, whereas silencers can repress splice sites or enhancers. Exon inclusion or skipping is determined by relative concentrations of the cognate RNA-binding activator and repressor proteins. Elementary alternative splicing events [62]

1.8. QUALITY CONTROL AND PREPROCESSING

High throughput sequencers can generate tens of millions of sequences in each run. Before analyzing this RNA-seq data and using it for transcriptome study to draw biological conclusions, quality control must be performed to ensure that RNA-seq data are of high quality and suitable for subsequent analyses without biases. Quality problems typically originate either in the sequencing itself or in the preceding library preparation. They include low-confidence bases, sequence-specific bias, 3′/5′ positional bias, polymerase chain reaction (PCR) artifacts, untrimmed adapters, and sequence contamination. These problems can seriously affect mapping to reference, assembly, and expression estimates. Many of those defects can be corrected for by filtering, trimming, error correction, or bias correction. While some cannot be corrected for, but they must be taken into consideration when interpreting results.

During my research I dedicated considerable time for quality control of FASTQ files[65]. Either by using the available Quality Control (QC) tools as FastQC[66], RSeQC[67] and Trimmomatic [68]. Or by coding a number of tools using HTSeq library in Python [69], to check the eligibility of RNAseq data, as described in the following.

1.8.1 FASTQ FORMAT:

FASTQ has emerged as a common file format for sharing sequencing read data combining both the sequence and an associated per base quality score. It provides an additional extension to the FASTA format, it is the ability to store a numeric quality score associated with each nucleotide in a sequence. However, it is lacking the clear formal definition. Furthermore, there are three incompatible variants of FASTQ format; original Sanger standard, the Solexa, and Illumina variants. Over time, the FASTA format has developed by consensus; however, in the absence of an obvious standard. For example, some parsers will fail to handle the very long '>' title lines or very long sequences without line wrapping. There is also no standardization for record identifier [65].

To introduce the three FASTQ variants, first I need to define PHRED, as a software reads DNA sequencing trace files, calls bases and assigns a quality value to each base called[70][71]. While the PHRED quality score of a base call, defined in terms of the estimated probability of error:

$$Q_{PHRED} = -10 \times \log_{10}(P_e) \qquad \textbf{(1)}$$

This means for example, if the probability of the error in calling a base is 0.01, so the PHRED score is 20.

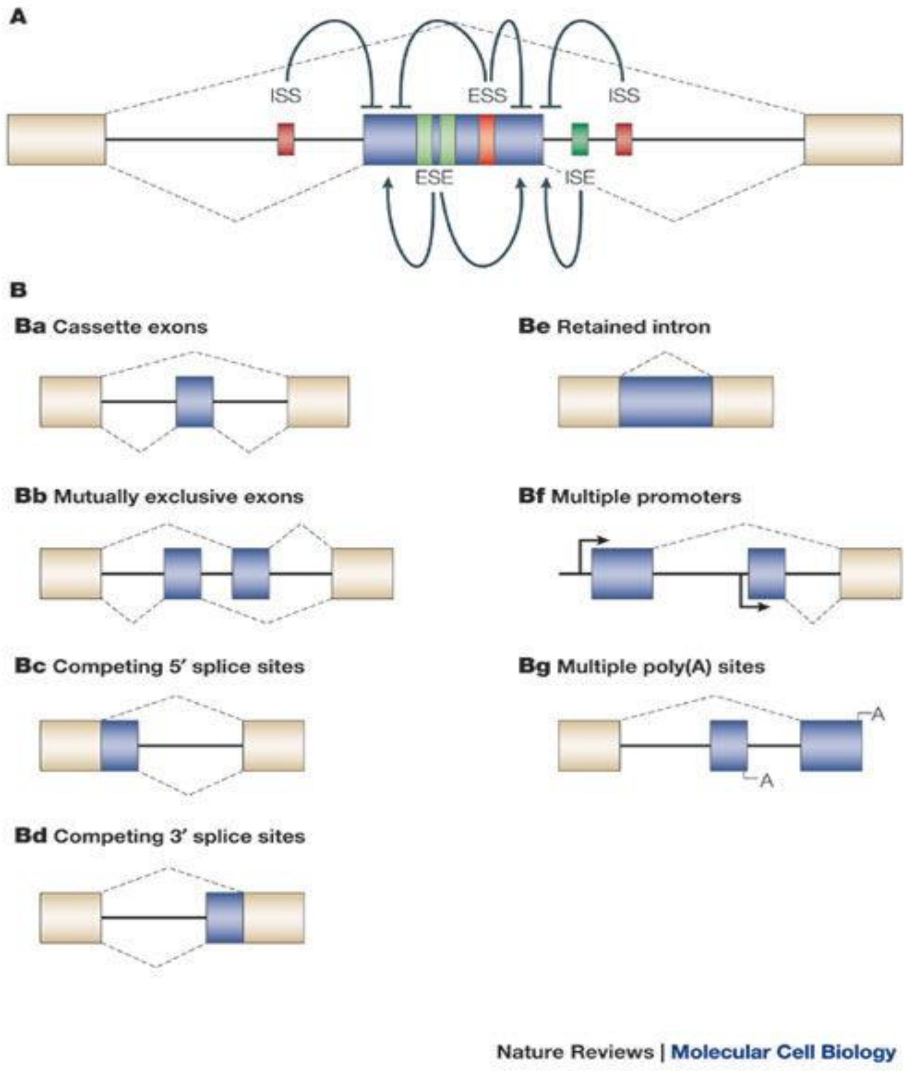
A
ISS
ESS
ISS
ESE
ISE
B
Ba Cassette exons
Be Retained intron
Bb Mutually exclusive exons
Bf Multiple promoters
Bc Competing 5′ splice sites
Bg Multiple poly(A) sites
A
A
Bd Competing 3′ splice sites
Nature Reviews | Molecular Cell Biology

Figure 1.25 Elementary alternative splicing events and regulatory elements.[62]

According to the Open Bioinformatics Foundation (OBF, http://www.open-bio .org), FASTQ has three variants; the original or standard FASTQ format, as the Sanger variant, using the format name ‘fastq-sanger’, Solexa FASTQ format ‘fastq-solexa’ and Illumina 1.3+ FASTQ format ‘fastq-illumina’ (Table 1).

In Sanger FASTQ format, in order that the file be human readable and easily edited, storing the PHRED scores was in ASCII printable characters 32–126 (decimal), and since ASCII 32 is the space character, Sanger FASTQ files use ASCII 33–126 to encode PHRED qualities from 0 to 93 (i.e. PHRED scores with an ASCII offset of 33). [72]

In Solexa FASTQ format, Solexa also produced other files with quality scores for all four bases, although the FASTQ format only records a single quality score per letter. Furthermore, in order to represent low-quality information more comprehensively, an alternative logarithmic mapping was used [73]. But rather they introduced their own incompatible (and indistinguishable) version of the FASTQ format [65]. Solexa quality scores are defined as:

$$Q_{\text{Solexa}} = -10 \times \log_{10}\left(\frac{P_e}{1-P_e}\right) \qquad \textbf{(2)}$$

Although Illumina initially continued to use the Solexa FASTQ variant, from Genome Analyzer Pipeline version 1.3 onwards, PHRED quality scores rather than Solexa scores were used [74]. The Illumina 1.3+ FASTQ variant encodes PHRED scores with an ASCII offset of 64, and so can hold PHRED scores from 0 to 62 (ASCII 64–126), although currently raw Illumina data quality scores are only expected in the range 0–40. (Table 1.1)

FASTQ variant	ASCII Characters		Quality Score	
	Range	Offset	Type	Range
Sanger standard *'fastq-sanger'*	33 - 126	33	PHRED	0 to 93
Solexa/early Illumina 'fastq-solexa'	59 - 126	64	Solexa	-5 to 62
Illumina 1.3+ 'fastq-illumina'	64 - 126	64	PHRED	0 to 62

Table 1.1 FASTQ variants between different sequencing platforms

1.8.2 TRIMMING LOW QUALITY READS:

The raw data of the next generation sequencing usually suffered, beside the attached adapters which must be removed, from low quality sequencing bases along the reads, that can easily result in suboptimal downstream analyses. Nevertheless, it is considerable to precisely identify such sequences, including partial adapter sequences, while leaving valid sequence data pristine [75]. Trimmomatic is the optimal choice designed to work on NGS data for identification of adapter sequences and quality filtering. It is able to process paired-end samples and optimized for Illumina NGS data [76].

The trimming procedures that Trimmomatics performed classified in the following list:

1.8.2.1 Removing technical sequences:

Identifying adapter or other contaminant sequences within a dataset is inherently a tradeoff between sensitivity (ensuring all contaminant sequences are removed) and specificity (leaving all non-contaminant sequence data intact). This issue is even more critical when only a small part of the contaminant sequence is included within the read. To detect technical sequences within the reads, Trimmomatic adopts two approaches for this purpose.

The first, referred to as 'simple mode', the advantage of this mode, that is working for all technical sequences, including adapters and polymerase chain reaction (PCR) primers. It performs by finding an approximate match between the read and the user supplied technical sequence, which can be passed to the tool as Fasta file. Such sequences can be detected in any location or orientation within the reads but requires a substantial minimum overlap between the read and technical sequence to prevent false-positive findings. However, short partial adapter sequences, which attached at the ends of reads, are unable to meet this overlap requirement and therefore are not detectable. In simple mode, each read is scanned from the $5'$ end to the $3'$ end to determine if any of the user-provided adapters are present. The standard 'seed and extend' approach is used to find initial matches between the technical sequences and the reads [77]. Based on this seed match, a local alignment is performed. If the alignment score exceeds the match threshold, the aligned region plus the remainder after the alignment are removed. [76]

Figure 1.26 illustrates the alignments tested for each technical sequence. In panel (A) In the beginning of the read, a partial overlap occurs of the 3′ end of the technical sequence with the 5′ end, so the rest of the read is trimmed. Even though with a complete overlap at the 5′ end (B) scenarios, the entire read is clipped as well. If the contaminant is found within the read (C), the bases from the 5′ end of the read to the beginning of the alignment are retained. The testing process continues until only a partial alignment on the 3′ end of the read remains (D).

The second mode, called as 'palindrome mode', is specifically optimized for the detection of 'adapter read-through'. However, it can only be used with paired-end data. When 'read-through' occurs, both reads in a pair will consist of an equal number of valid bases, followed by contaminating sequence from the 'opposite' adapters. Furthermore, the valid sequence within the two reads will be reverse complements. By combining these three symptoms, Palindrome mode algorithm can identify adapter read-through with high sensitivity and specificity.

How the algorithm works? The adapter sequences are prepended to their respective reads, and then the combined read-with-adapter sequences from the pair are aligned against each other. A high-scoring alignment indicates that the first parts of each read are reverse complements, while the remaining parts of the reads match the respective adapters. The alignment is implemented using a 'seed and extend' approach, similar to that in simple mode. Global alignment scoring is used to ensure an end-to-end match across the entire overlap.

The alignments in palindrome algorithm is illustrated in Figure 1.27 The process starts by align the forward and reverse reads against each other, so if the adapter of the reverse read aligned to 5′ end of the forward read, and vice versa as shown in panel (A).

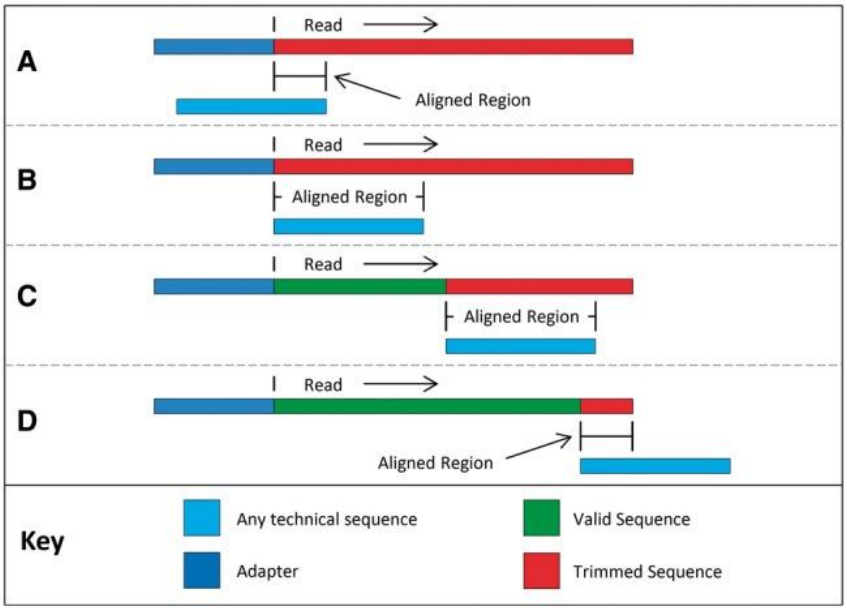

Figure 1.26: Simple mode in Trimmomatic. The alignment process begins with a partial overlap at the 5′ end of the read (A), increasing to a full-length 5′ overlap (B), followed by full overlaps at all positions (C) and finishes with a partial overlap at the 3′ end of the read (D).[76]

This interpreted that the read pair contains uninformative sequence which could be caused by the direct ligation of the adapters, therefore the both reads will be dropped. In (B) the adopters align within the reads, so the sequence portion before the adopter is accepted and the fragment that aligns to the adopter and on is dropped. Even when only a small fragment of the adapter is overlapping, as shown in (C).

1.8.2.2 Quality filtering:

Trimmomatic offers two main quality filtering alternatives; sliding window and maximum information. The Sliding Window uses a relatively standard approach. This works by scanning from the 5′ end of the read, and removes the 3′ end of the read when the average quality of a group of bases drops below a specified threshold. This prevents a single weak base causing the removal of subsequent high-quality data, while still ensuring that a consecutive series of poor-quality bases will trigger trimming.

During trimming we need to take in the consideration to find the balance in the retaining read length. It is clear that short reads are almost worthless because they occur multiple times within the target sequence and thus they give only ambiguous information. However, beyond a certain read length, retaining additional bases is less beneficial. Therefore, the smaller potential benefit of retaining additional bases must be balanced against the increasing risk of retaining errors, which could cause the existing read value to be lost.

As such, it is worthwhile for the trimming process to become increasingly strict as it progresses through the read, rather than to apply a fixed quality threshold. The ‘Maximum Information’ quality filtering implements this adaptive approach. It uses a combination of three factors to determine how much of each read should be retained. [76]

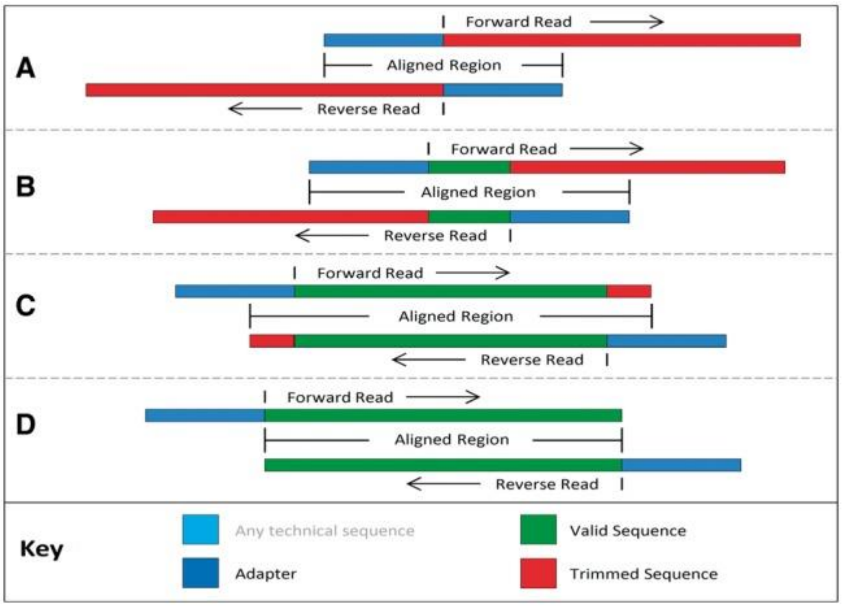

Figure 1.27: Palindrome mode in Trimmomatic. The alignment process begins with the adapters completely overlapping the reads (A) testing for immediate ‘read-through’, then proceeds by checking for later overlap (B), including partial adapter read-through (C), finishing when the overlap indicates no read-through into the adapters (D) [76]

The first factor models the ‘length threshold’ concept, whereby a read must be of at least a minimal length to be useful for the downstream application. On the other hand, most long reads can be mapped to few locations in the target sequence. So, if they cannot be uniquely mapped, because of the repetitive regions, it is unlikely that a small number of additional bases will resolve this. For reads between these extremes, the marginal benefit of a small number of additional bases is considerable, as these extra bases may make the difference between an ambiguous and an informative read. A logistic curve was chosen to implement this scoring behavior, as it gives a relatively flat score for extreme values, while providing a steep transition around the user-specified threshold point. The following equation gives a length threshold score:

$$Score_{LT}(l) = \frac{1}{(1 + e^{t-l})}$$

Whereas: *t* is a target length, and *l* is the putative length after trimming.

The second factor models ‘coverage’, it provides a linear score based on retained sequence length:

$$Score_{cov}(l) = l$$

The third factor models the ‘error rate’, it uses the error probabilities from the read quality scores to determine the accumulated likelihood of errors over the read. To calculate this score, we simply take the product of the probabilities that each base is correct, giving:

$$Score_{Err} = \prod_{i=l}^{l} P_{corr}[i]$$

The correctness probabilities P_{corr} of each base are calculated from the sequence quality scores. The error score typically begins as a high score at the start of the read, depending on the read quality, typically drops rapidly at some point during the read.

The Maximum Information algorithm determines the combined score of the three factors for each possible trimming position, and the best combined score determines how much of the read to trim. A strictness parameter s can be set between 0 and 1, controls the balance between the ‘coverage’ factor (for s = 0) and the ‘error rate’ factor (for s =1). This gives the following formula:

$$Score(l) = \frac{1}{(1 + e^{t-l})} \,.\, l^{(1-s)} . \left(\prod_{i=l}^{l} P_{corr}[i]\right)^{s}$$

Figure 1.28 illustrates how the three factors are combined into a single score. The peak score is then used to determine the point where the read is trimmed

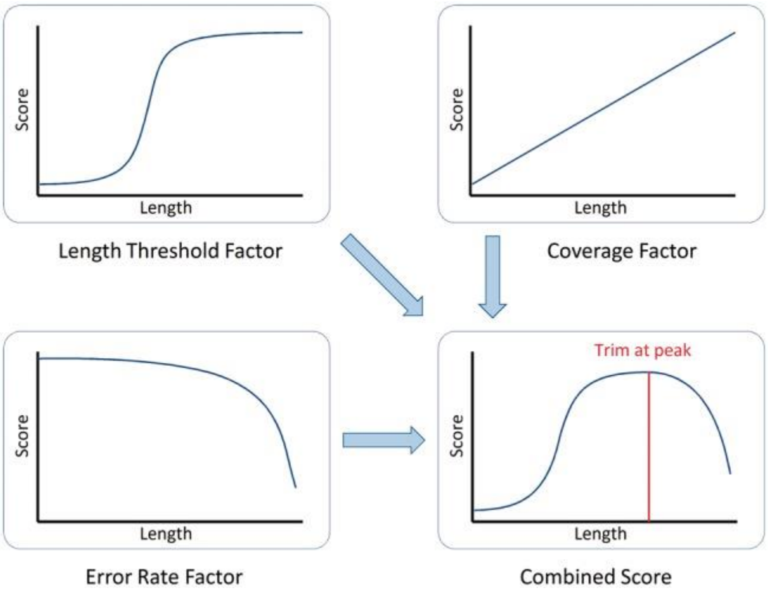

Figure 1.28: Maximum Information mode. It combines length threshold, coverage, and error rate scores to determine the optimal trimming point [76]

1.9. SEQUENCE ALIGNMENT

The purpose of sequence alignment is to figure out where the sequences are similar and how high the similarity is. Aligning or “mapping” reads to a reference genome or transcriptome allows us to estimate where the read originated from. Mapping reads to genome provides genomic location information, which can be used for discovering new genes and transcripts, and for quantifying expression. If a reference genome is not available, or if our target is to quantify only known transcripts, reads can be mapped to a transcriptome instead.

Aligning reads to a reference genome is a challenging task for many reasons; reads are relatively short and there are millions of them, while genomes can be large and contain ambiguous sequence regions or indistinct such as repetitive regions and pseudogenes, this can impact the mapping to these areas. Furthermore, aligners have to cope with mismatches and indels (insertions-deletions) caused by genomic variation and sequencing errors. Eventually, many organisms have introns in their genes, so RNA-seq reads align to genome non-contiguously. Placing spliced reads across introns and determining exon–intron boundaries correctly is difficult, because sequence signals at splice sites are limited and introns can be thousands of bases long.

Alignment information are stored in sequence alignment/map (SAM) format. They are generated from different types of aligners, as contiguous aligners or spliced aligners as I describe later on.

1.9.1 SEQUENCE ALIGNMENT/MAP (SAM) FORMAT

The Sequence Alignment/Map (SAM) format is a common alignment format that supports all sequence types and aligners creates a well-defined interface between alignment and downstream analyses, including variants detection, genotype prediction and gnome assembly. It supports single- and paired-end reads and combining reads of different types, including color space reads from AB-SOLiD. It is designed to tabulate alignment sets of 10^{11} or more base pairs, which is typical for the deep sequencing of an individual human genome[78].

It is a generic alignment format for storing read alignments against reference sequences, supporting short and long reads (up to 128 Mbp) produced by different sequencing platforms. It is a TAB-delimited text format consisting of a header section, and an alignment section. The header must be prior to the alignments. Header lines start with ‘@’, while alignment lines do not.

Each header line has record type code of two letters follow character ‘@’. The fields in the header line is TAB-delimited, each data field follows a format ‘TAG:VALUE’ where TAG is a two-letter string that defines the content and the format of VALUE. Detailed definitions of record types and tags in SAM header line is provided in “Sequence Alignment/Map Format Specification” manual file[79], available in Sam-Tools[78] github. While each alignment line has 11 mandatory fields for essential alignment information such as mapping position, and variable number of optional fields for aligner specific information.

SAM format is human understandable, and easy to check for errors. However, SAM is a slow to parse especially for full eukaryote genome alignment. Therefore, it was required to have a binary representation of SAM to improve the performance for intensive data processing. It is called BAM (Binary Alignment/Map) format. it is used in most production pipelines and keeps exactly the same information as SAM. BAM is compressed by the BGZF library, a generic library developed to achieve fast random access in a compressed file[78].

1.9.2 VARIOUS ALIGNMENT TYPES

SAM can store various types of sequence alignment, classified in the following categories.[80]

1. **Linear alignment**. An alignment of a read to a single reference sequence that may include insertions, deletions, skips and clipping, but may not include direction changes (i.e. one portion of the alignment on forward strand and another portion of alignment on reverse strand). A linear alignment can be represented in a single SAM record
2. **Clipped alignment**: In Smith-Waterman alignment[81], a sequence may not be aligned from the first residue to the last one. Subsequences at the ends may be clipped off. It is flagged in SAM with ‘S’ to define Soft clipped alignment. Here is an example.
3. **Spliced alignment:** When RNA-Seq reads align to genome reference, they span over introns and sometimes exons according to the splicing event. To distinguish in the alignment splicing from deletions in exons, SAM flag this alignment by ‘N’ to represent long skip on the reference sequence.
4. **Multiple alignment:** Read alignment could be ambiguous. One query sequence may be aligned to multiple places on the reference genome due to repeats for example. In this case, there will be multiple read alignments for the same read. In SAM, one of these alignments is considered primary, while all the other alignments have the secondary alignment flag.
5. **Padded alignment**. Alignment with inserted sequences fully aligned is called padded alignment. Padded alignment is always produced by de novo assemblers and is important for an alignment viewer to display the alignment properly. To store padded alignment, SAM introduced operation ‘P’ which can be considered as a silent deletion from padded reference sequence.
6. **Alignments in color space**. Color alignments are stored as normal nucleotide alignments with additional tags describing the raw color sequences, qualities and color-specific properties.

1.9.3 ALIGNMENT PROGRAMS

Tens of alignment programs have been developed, offering various approaches to overcome alignment challenges. Fonseca et al. [81] provide a comprehensive survey of aligners and update the listing on the web [82]. Typically, aligners apply some heuristics and use different indexing schemes to speed up the process. Many tools can consider base quality values when scoring mismatches and they can also make use of the expected distance and relative orientation of paired-end reads.

Aligners report the confidence in the mapping location as mapping quality ($Q = -10\log_{10} P$, where P is the probability that the read originated elsewhere). Mapping quality can depend on several things, but the most important one is uniqueness. Some aligners are able to distribute multimapping reads proportionally to the coverage between the equally matching locations. Spliced aligners specific for RNA-seq reads use different approaches for aligning spliced reads. This can include performing an initial alignment to discover exon junctions, which then guide the final alignment. If genomic annotation is available, aligners can use it for placing spliced reads. Spliced aligners differ in their alignment yield, splice-detection performance, base-wise accuracy, tolerance for mismatches, and indel detection, as shown by the systematic evaluation performed by Engström et al. [82]. The main consideration when choosing an aligner for RNA-seq studies is whether spliced alignments are needed or not. If the organism does not have introns or if microRNAs were sequenced, it is fine to use contiguous aligners like Bowtie [83] or BWA [84], which were originally developed for DNA. These aligners can also be used if reads are mapped to a transcriptome rather than a genome. However, if RNA-seq reads are mapped to genomes which contain introns, a spliced aligner like TopHat [85], STAR [86], or GSNAP [86] is necessary.

In our research, we needed spliced aligner as TopHat for alternative splicing analysis and defining spliced junctions.

1.9.3.1 TopHat aligner:

TopHat is an optimal fast junction aligner for RNA-Seq reads to mammalian-sized genomes, in order to identify exon-exon splice junctions. It is built on the ultra-high-throughput short read mapping program Bowtie, which is an ultrafast, memory-efficient alignment program for aligning short DNA sequence reads to large genomes. Bowtie extends previous Burrows-Wheeler techniques with a novel quality-aware backtracking algorithm that permits mismatches [87].

TopHat has the ability to detect splice junctions without a reference annotation of known junctions. By mapping RNA-Seq reads to the genome firstly, TopHat identifies potential exons, due to the fact that many RNA-Seq reads will contiguously align to the genome. Therefore, a database of possible splice junctions will be built, using this initial alignment information, then TopHat maps the reads against these junctions to confirm them. It has the ability to identify novel splice sites with direct mapping to known transcripts. Many exons are shorter than reads produced by short reads sequencing machines. Therefore, they might be missed in the initial mapping. TopHat solves this problem mainly by splitting all input reads into smaller segments which are then mapped independently. The segment alignments are put back together in a final step of the program to produce the end-to-end read alignments.

TopHat generates its database of possible splice junctions based on two sources of evidence. The first and strongest source of evidence for a splice junction is when two segments from the same read, with minimum length 45bp, are mapped to the same genomic sequence, but at a specific distance, or when an internal fragment fails to map directly to the genome, this scenario suggesting that such reads are spanning multiple exons. The second source is pairings of "coverage islands", which are distinct regions of piled up reads in the initial mapping. Neighboring islands are often spliced together in the transcriptome, so TopHat looks for ways to join these with an intron. [88]

Another issue can arise during assembly, that the genes transcribed at low levels will be sequenced at low coverage, hence the exons in these genes may have gaps. To solve of this issue TopHat has a parameter that controls when two distinct but nearby exons should be merged into a single exon. This parameter defines the length of the longest allowable coverage gap in a single island. Due to the fact that, introns in mammalian are rare be shorter than 70 bp [89], any value less than 70 bp for this parameter is reasonable. However, TopHat chose conservative value of this default gap of 6 bp.

The second phase of TopHat alignment procedures is to remap the initially unmapped reads (IUM). To map reads to splice junctions, TopHat first enumerates all approved donor and acceptor sites within the island sequences. Next, it considers all pairs of these sites that could form canonical (GT–AG) introns between neighboring islands. Each possible intron is checked against the IUM reads for reads that span the splice junction. By default, TopHat only examines potential introns longer than 70 bp and shorter than 20 000 bp, but these default minimum and maximum intron lengths can be adjusted by the user. These values describe the vast majority of known eukaryotic introns. For example, more than 93% of mouse introns in the UCSC known gene set are within this range. To avoid reporting false positives, TopHat excludes donor–acceptor pairs that fall entirely within a single island. However, to detect junctions without sacrificing performance and specificity, the algorithm enumerates introns within islands, only if they are deeply sequenced. During the island extraction on the first phase of the pipeline, the algorithm computes the following statistic for each island spanning coordinates i to j in the map [88]:

$$D_{ij} = \frac{\sum_{m=i}^{j} d_m}{j - i} \cdot \frac{1}{\sum_{m=0}^{n} d_m}$$

where dm is the depth of coverage at coordinate m in the Bowtie map, and n is the length of the reference genome. the single-island junctions tend to fall within islands have high D. Thus, TopHat looks for junctions contained in islands withD≥300, this parameter can be changed by the user too. A high D -value will prevent TopHat from looking for junctions within single islands, which will improve running time. A low D -value will force TopHat to look within many islands, slowing the pipeline, but potentially finding more junctions.

using a seed-and-extend strategy, Tophat remap the IUM reads to each splice junction, in order to find reads that span junctions. The pipeline indexes the IUM reads using a lookup table to speed up and simplified searching for a spliced alignment over many reads.

As illustrated in Figure 1.29, TopHat finds any reads that span splice junctions by at least k bases on each side (k=5 bp by default). For each read, the table contains $(s - 2k + 1)$ entries corresponding to possible positions where a splice may fall within a read, where s is the length of the high-quality region on the $5'$ end (default = 28 bp). For longer reads may be useful to increase s to improve sensitivity. Lowering s will improve running time, but may reduce sensitivity. Increasing k will improve running time, but may limit TopHat to finding junctions only in highly expressed genes.

Afterward TopHat takes each possible splice junction and makes a 2k-mer 'seed' for it by chain the k bases downstream of the acceptor to the k bases upstream of the donor. The IUM read index is then queried with this 2k-mer to find all reads which contain the seed. This exact 2k-mer match is extended to find all reads that span the splice junction. To extend the exact match for the seed region, TopHat aligns the portions of the read to the left and right of the seed with the left island and right island, respectively.

1.10.GENE EXPRESSION ANALYSIS

Once reads have been mapped to a reference genome, their mapping locations can be identified by genomic annotation. This enables us to quantitate gene expression by counting reads per genes, transcripts and exons. Quantitation of gene expression is an integral part of most RNA-Seq studies. In principle, calculating the count of mapped reads provides a direct way to estimate transcript abundance, it has been found that read count is approximately linearly related to the abundance of the target transcript[25], but in practice several complications need to be taken into account. Eukaryotic genes typically produce several transcript isoforms via alternative splicing and promoter usage. However, quantitation at transcript level is not easy with short reads, because transcript isoforms often have common or overlapping exons. Furthermore, the coverage along transcripts is not uniform because of mappability issues and biases introduced in library preparation. Because of these complications, expression is often estimated at the gene level or the exon level instead. However, gene level counts are not optimal for differential expression analysis for those genes which undergo isoform switching, because the number of counts depends on transcript length. This challenge can be overcome by applying the appropriate reads count normalization in addition to an effective statistical model for significant variability estimation, where A number of model-based methods have been developed that attempt to deconvolve the expression levels of individual transcripts for each gene from RNA-seq data, essentially by leveraging information from reads unambiguously assigned to regions where isoforms differ as RSEM [90], and cuffdiff from cufflinks package [91].

Differential expression analysis of RNA-seq data differs from microarray. In RNA-Seq the observed data are in the form of discrete counts generated from a sampling process, while microarray measurements are continuous measurements of a fluorescence signal.

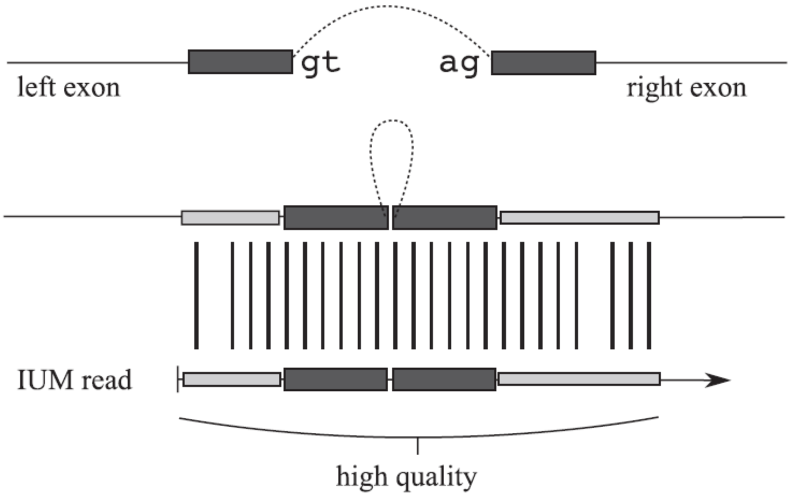

Figure 1.29: TopHat seed-and-extend strategy. The seed and extend alignment used to match reads to possible splice sites. For each possible splice site, a seed is formed by combining a small amount of sequence upstream of the donor and downstream of the acceptor. This seed, shown in dark gray, is used to query the index of IUM (Initially UnMapped) reads. Any read containing the seed is checked for a complete alignment to the exons on either side of the possible splice. In the light gray portion of the alignment, TopHat allows a user-specified number of mismatches. Because reads typically contain low-quality base calls on their 3′ ends, TopHat only examines the first 28 bp on the 5′ end of each read by default.[88]

1.10.1ASSIGNING SEQUENCE READS TO GENOMIC FEATURES

The essential information required for downstream analysis is the number of reads mapping to each genomic feature, depending on the next-gen application, the genomic features might be exons, genes, promotor regions, gene bodies or other genomic intervals [92]. This reads summarization process is performed following the reads alignment in term to give an adequate biological interpretation. There are several tools to serve this task. However, those Read count programs need to accommodate both DNA and RNA sequencing as well as single and paired-end reads. The reads or paired-end fragments to be counted may incorporate insertions, deletions, or fusions relative to the reference genome, and these complications should be accounted for when comparing the location of each read or fragment to each possible target genomic feature.

Counting RNA-Seq reads is somewhat more complex because of the need to accommodate exon splicing. One way is to count reads overlapping each annotated exon, an approach that can be used to test for alternative splicing between experimental conditions[93]. Another common approach is to summarize counts at the gene level, by counting all reads that overlap any exon for each gene[94], Gene annotation from RefSeq or Ensembl is often used for this purpose. In RNA-Seq, reads count at the gene level provide an overall summary of the expression level of the gene but do not distinguish between transcripts that have been expressed from the same gene. Reads can generally be assigned to genes with good confidence, but estimating the expression levels of individual isoforms is intrinsically more difficult because different isoforms of the gene typically have a high proportion of genomic overlap. Many statistical analysis methods have been developed to detect differential expression or differential binding on the basis of read counts ([95], [96],[97]).Recent comparisons have concluded that the read count methods perform relatively well for the purposes of gene-level differential expression ([98], [99]) or detection of splice variation [93].

A few software for read count tools are currently available. The software packages GenomicRanges and IRanges [100] , developed by the core team of the Bioconductor project[101], include functions for counting reads that overlap genomic features. The countOverlaps function of IRanges is designed for counting reads overlapping exons or other simple genomic regions, whereas the *summarizeOverlaps* function of *GenomicRanges* is designed for counting reads at the gene level. Another tool is the HTSeq-count script distributed with the HT-Seq Python framework for processing RNA-seq or DNA-seq data [102]. Both of these software tools are popular and well tested, but they use extensively programming computer languages as R or Python, so they are not optimal for efficiency and speed.

BED-Tools is a popular tool for finding overlaps between genomic features that can be used to count overlaps between reads and features[103]. It is fully implemented in the compiled language C++, making it faster than the aforementioned tools. However, it is not specifically designed for RNA-seq data, so can count reads for exons or interval features only, similar to countOverlaps.

Although several read count tools are available, I used in my research featureCounts tool [94], it is highly optimized read count program, can be used to quantify reads generated from either RNA or DNA sequencing technologies in terms of any type of genomic feature. It implements chromosome hashing, feature blocking and other strategies to assign reads to features with high efficiency. It supports multithreading (using multi-processors in parallel), which provides further speed improvements on large data problems. It is available either as a Unix command or as a function in the R package Rsubread [104]. In either case, all the core functionality is written in the C programming language. The R function is a wrapper for the compiled C code that provides the convenience of the R programming environment without sacrificing any of the efficiency of the C implementation.

featureCounts function in Rsubread has 13 categories of arguments[105], the main arguments I needed to set for reads summarizing in our research, as the following:

1- *files*: Input files containing read mapping results. The files are in BAM format.

2- *annot.ext*: is a character string giving name of gene annotation file. The annotation is in GTF format downloaded from Ensembl (GRCm38 release 86).

3- *isGTFAnnotationFile*: is a logical indicating whether the annotation provided via the annot.ext argument is in GTF format. FALSE by default. Since a GTF annotation file is used, it is TRUE.

4- *GTF.featureType*: is a character string giving the feature type used to select rows in the GTF annotation which will be used for read summarization. "exon" by default. This argument is only applicable when isGTFAnnotationFile is TRUE. I kept the default.

5- *GTF.attrType*: is a character string giving the attribute type in the GTF annotation which will be used to group features (eg. exons) into meta-features (eg. genes). It is set to “gene_name”.

6- *IsPairedEnd:* is a logical indicating if paired-end reads are used. In our data is TRUE.

The output of featureCounts is a list of the following components, they are accessed by '$' after the featureCounts object then the name of the component (their letters in lowercase):

1- *counts:* a data matrix containing read counts for each feature or meta-feature. It is a matrix the rows are the features (in our research genes names), and the columns are the libraries or biological conditions (osteoblast differentiation time points).

2- *annotation:* a data frame with six columns including GeneID, Chr, Start, End and Length. Since read summarization was performed at meta-feature level, each row in the data frame is a meta-feature (gene) and columns in the data frame give the annotation information for the features included in each meta feature except the Length column (in our annotation matrix the GeneID column is the genes names, because we set *GTF.attrType* to "gene_name"). The Length column gives the total length of genomic regions covered by features included in that meta-feature.

3- *targets:* a character vector giving sample information.

4- *stat:* a data frame giving numbers of unassigned reads and the reasons why they are not assigned (e.g. ambiguity, multi-mapping, secondary alignment, mapping quality, fragment length, chimera, read duplicate, non-junction and so on), in addition to the number of successfully assigned reads for each library

1.10.2TECHNICAL VERSUS BIOLOGICAL REPLICATES

The word replicate means that we get more than one measurement of the quantity of interest. Replication is considered one of the three cornerstones of proper experimental design outlined by Fisher (1935)[106]: randomization, replication, and blocking. An excellent explanation of these concepts in the context of RNA sequencing can be found in a paper by Auer and Doerge [107], which is a highly recommended to read before planning an experiment. The purpose of replication is to be able to estimate the variability between and among groups, which is important for hypothesis testing.

Technical replication is used to estimate the variability of the measurement technique, as RNA sequencing techniques. While biological replication is used to find out the variability within a biological group. Changes in gene expression between two groups can only be called significant if the difference between the groups is large compared to the variability within the group, while taking the sample size into consideration.

There can be different types of technical replicates, for example, sequencing the same library in two different lanes of a sequencer or different library preparations performed on the same sample of extracted RNA. Typically, the RNA extraction would be the same in technological replicates, but different in biological replicates.

There are also borderline cases where it is hard to call replicates "biological" even if they are taken from sources that could be considered different, for instance, different cultures of the same genetically homogeneous cell line. In these cases, the important thing is to think about what questions a given differential expression comparison would answer?

The number of replicates depends on the specifics of the experiment. The biological homogeneity of the different samples, the purpose of the experiment and the desired level of statistical power, among other things, will affect the number of replicates needed. Many sequencing core facilities require or suggest using at least three or four replicates per group to be compared. With three replicates, there is the risk that at least one sample might fail in library preparation or sequencing, so we would end up with only two replicates in one of the groups, whereas two is too few. Particularly for complex diseases, very large numbers of replicates (perhaps hundreds or thousands) may be needed to observe differential expression between cases and controls, to avoid considerable variation between individuals, especially human blood or tissue samples. For cell lines or samples from distinct tissues, only a few replicates may be needed.

1.10.3 GENERAL CONCEPTS OF STATISTICAL DISTRIBUTIONS IN RNA-SEQ DATA

Expression levels of the same gene across different cells have been shown to follow a log-normal distribution as measured by quantitative PCR[108]. To figure out the variation in biological functions and features among various tissues or biological conditions, we need to statistically analyze the significant variance of the quantitative abundance of a specific genetic feature (gene, transcript or exon).

For RNA-seq, the read count that assign to a transcript is linearly related to its abundance (with good approximation) [25].The read count can be approximated by Poisson distribution, if we accept the assumption that sequenced reads are independently sampled from the sequencing library or the population, and we have fixed fractions of genes.

However, we would expect slightly different counts even for the same library in an idealized circumstance, where it was sequenced twice under the same conditions. This inevitable noise which arises from the sampling process is called shot noise, and often the variability between technical replicates in RNA-seq can be described quite well by this type of Poisson noise. In some researches, Poisson distribution has been used to test for differential expression[109][110]. However, the assumption that the reads are Poisson-distributed, is quite restrictive, because it predicts smaller variations than what can be found in the data. Therefore, this statistical model does not control type-I error (the probability of false discoveries).[111][112]

To come over this over-dispersion problem, when samples are taken from biologically distinct sources, such as different individuals, the variability between them has often been modeled by a negative binomial distribution or gamma-Poisson distribution. This distribution can be described as an over-dispersed Poisson distribution; a version of that distribution but with higher variance. While a Poisson distribution has the same variance as its mean μ, the negative binomial distribution's variance can be written as

$$\sigma^2 = \mu + \left({}^{1}\!/_{r}\right).\mu^2$$

The parameter r is a positive integer; therefore, the variance will always be larger than the mean.

There are many methods for differential expression analysis of RNA-seq data. DESeq [95] and edgeR [113] use the negative binomial distribution as the basis of their modeling of RNA-seq counts. However, RNA-seq count data also show some characteristics like zero inflation (a large proportion of values with zero counts) which makes it harder to fit a negative binomial distribution. Whereas edgeR moderates the dispersion estimate for each gene toward a common estimate across all genes, using a weighted conditional likelihood. While DESeq method detects and corrects dispersion

estimates that are too low through modeling of the dependence of the dispersion on the average expression strength over all samples. BBSeq method [114] models the dispersion on the mean, it reduces the influence of outliers by estimating the mean absolute deviation of dispersion. DSS [115] uses a Bayesian approach to provide an estimation of the dispersion for individual genes. baySeq [116] and ShrinkBayes [117] estimate priors for a Bayesian model over all genes, and then provide posterior probabilities or false discovery rates (FDRs) for differential expression.

2. AIMS OF THE DOCTORAL THESIS

In recent years several pipelines were founded for high throughputs sequence data analysis, many tools are available for quality control of RNA-Seq data, reads mapping, comparative analysis of gene expression and alternative exons usage, and for finding de novo long non-coding RNA. However, most of the analysis fail to deal the integrity, large datasets, and to produce descriptive results, that biologists can interpret without addition effort.

The main aim of this doctoral work is to introduce an integrated RNA-Seq data analysis pipeline, that produces illustrated outputs, especially in the transcriptomic characterization experiment. This type of experiment is based on an expanded investigation of a comparative gene expression between different biological conditions, where we have a numerous amount of outputs needed to be examined to get the significant and informative results. Based on those outputs, we can build hypothesis describes the gene regulation stands behind that biological mechanism. The main aims we achieved in this doctoral thesis can be listed as follows:

1- Introducing several codes for RNA-Seq data quality control, which provide tables of summarized reads statistics in samples, and give the mean of Phred quality scores across all the bases in a sample. In addition to plotting the mean quality of each base in all samples, we established a method to check the coverage uniformity. Especially when polyadenylated RNA library is used, there is usually concern that the coverage might vary across the gene's features.

2- Establishing a comprehensive framework for differential gene expression analysis, that produces descriptive outputs and facilitates the biological interpretation of the experiment.

3- Presenting an analyzing approach for multiple conditions experiment, we called it "ON/OFF genes", which can define the silent genes in a particular condition of the comparative analysis. This approach can highlight the functional roles, that genes can play in different conditions, and give a wider view of the genetical reasons behind the distinction between biological conditions.

4- Improving the performance Improving the performance of Bioconductor package DEXSeq [93]for differential exons usage, by specifying if the differentially used exonic part is within the ORF. This procedure will help to figure out if the differential used exons are involved in the alternative pathways, or distinctive functions of a gene's transcripts.

5- Suggesting a new approach to analyze differential expressed long non-coding RNA, by finding the functional correlation of lncRNA with neighboring differential expressed protein coding genes within the TAD (Topological associated Domain), to obtain an illustrated view concerning the regulation mechanism in a dataset.

3. RNA-SEQ DATA ANALYSIS WORKFLOW

3.1. OSTEOBLAST DIFFERENTIATION EXPERIMENT

The RNA-Seq data which our research based on, are from differentiated osteoblast cells. Although osteoblast differentiation was well characterized, a detailed transcriptional analysis of osteoblast differentiation based on RNA sequencing (RNA-seq) analyses is still missing. Therefore, we used RNA-seq to obtain a high-resolution transcriptome data set of murine osteoblast differentiation in vitro. The cells were harvested at four distinct time points: within proliferation, during maturation, terminal differentiation, and at the onset of mineralization.

Primary calvaria osteoblasts (pCOBs) were harvested from newborn C57BL/6 wild-type mice (P0-P4) as described in the project publication [118]. Cells were seeded on 6-well plates in Alpha-Mem (Lonza, Basel, Switzerland) containing 10 % fetal calf serum (FCS; Gibco, Life Technologies, Carlsbad, California, USA) as well as Pen/Strep (100U/mL, Lonza) and 2 mM ultra-glutamine (Lonza).

At confluence, we defined it as day 0, cells were harvested and on the other plates medium was supplemented with 50 µM L-ascorbate-2-phosphate and 10 mM beta-glycerophosphate to promote osteoblast differentiation. Cells were washed with PBS before lysis in RNAPure (PeqLab) at day 0 (i.e. confluent cultures without stimulation), day 3, day 6, and day 12. Total RNA was isolated using phenol/chloroform extraction. RNA integrity was confirmed using the Agilent 2100 bioanalyzer with Agilent RNA 6000 Nano Kit according to the manufacturer's instructions. This procedure was repeated three times to obtain biological replicates.

As a consequence, we got 12 samples, 3 for each differentiation time point:

1- Day 0: The confluency of cells in the culture plate, before promoting the differentiation.
2- Day 3: Harvesting cells on the third day after the differentiation starts.
3- Day 6: Harvesting on the sixth day after the differentiation promoted.
4- Day 12: On the twelfth day of differentiation.

3.2. QUALITY CONTROL AND REPROCESSING METHODS

The first procedure needed to be implemented after we have the RNA-Seq data as FASTQ files, is checking the quality of the raw read sequencing. Once reads have been aligned to a reference genome, additional quality metrics can be investigated based on the location. These include coverage uniformity along transcripts, saturation of sequencing depth, ribosomal RNA content, and read distribution between exons, introns, and intergenic regions. Finally, once aligned reads have been counted per genes, sample relations and batch effects can be visualized with heatmaps and PCA plots.

3.2.1 FASTQC

FastQC [66] is available as a standalone interactive Java application with a graphical user interface (GUI), and it can be run as well in a command line as non-interactive mode, where it would be suitable for integrating into a larger analysis pipeline for the systematic processing of large numbers of files. The input files can be FASTQ or SAM/BAM files. In addition to list the number of reads and their quality encoding, FastQC reports and visualizes information on base quality and content, read length, and k-mer content, also on the presence of ambiguous bases, over-represented sequences, and duplicates.

As we described in a previous chapter about the osteoblast dataset, we have 4 osteoblast differentiation time points with 3 replicates of each biological condition. They are12 samples of forward and reverse reads, so 24 FASTQ in total.

Using the Bash Script, we coded a function to input the samples of original raw reads to FastQC to check the read qualities. This function can serve large experiment to pass samples to FastQC and get outputs by one command. User does not to care about how to input a set of samples, neither getting the outputs. As described in the Code-box 3.1.

FastQC generates QC report contains 12 analysis modules as follows:

1. *Basic Statistics module:* it summarizes statistical information about the sequencing reads, such as:

   - Filename: The original filename of the analyzed file.
   - File type: That means whether the file contains actual base calls or color-space data which had to be converted to base calls.
   - Encoding type as which ASCII encoding of quality values was found in this file.
   - Total Sequences: A count of the total number of sequences processed.
   - Filtered Sequences: the count of Sequences flagged as poor quality.
   - Sequence Length: Provides the length of the shortest and longest sequence in the set. If all sequences are the same length only one value is reported.
   - %GC: The overall %GC of all bases in all sequences

2. *Per Base Sequence Quality module:* This module generates a plot, shows an overview of the range of quality values across all bases at each position in the FastQ file. This module produces errors if the lower quartile for any base is less than 5 or if the median for any base is less than 20. In our data, there are 3 samples with this low-quality issue, Day0 Replicate2 forward strand, and Day3 Replicate3 forward strand, and Day12 Replicate1 forward strand. Later I described how to overcome this quality deficiency in the samples. In the Figure 3.1 an example of per base sequence quality plot of raw reads sample.

3. *Per Sequence Quality Scores module:* it allows to see if a subset of the sequences have universally low quality values. However, these should represent only a small percentage of the total sequences. As shown in Figure 3.2.

4. *Per Base Sequence Content module:* it plots out the proportion of each base position in the sequencing reads for each of the four DNA bases has been called. In a random library would be expected little to no difference between the different bases of a sequence run, so the lines in this plot should run parallel with each other. It's worth noting that some types of library will always produce biased sequence composition, normally at the start of the read. Libraries produced by priming using random hexamers (almost all RNA-Seq libraries) and those which were fragmented using transposases inherit an intrinsic bias at the start positions od reads. This bias does not concern an absolute sequence, but instead provides enrichment of a number of different K-mers at the 5' end of the reads. Whilst this is a true technical bias, it isn't something which can be corrected by trimming and in most cases, doesn't seem to adversely affect the downstream analysis. All or samples showed this bias, as it is shown in Figure 3.3.

5. *Per Sequence GC Content module:* it measures the GC content distribution across the whole length of each sequence in a file and compares it to a modelled normal distribution of GC content. An unusually shaped distribution could indicate a contaminated library or some other kinds of biased subset. A normal distribution which is shifted indicates some systematic bias which is independent of base position.

6. *Per Base N Content module:* If a sequencer is unable to make a base call with sufficient confidence then it will normally substitute an N rather than a conventional base call. This module plots out the percentage of base calls at each position for which an N was called.

Code-Box 3.1: FastQC insert function

```
#!/bin/bash
## This script encodes a function to insert data to FastQC
## It puts the output into a new directory called FASTQC_OUTPUT
## Author Layal Abo Khayal 19.03.2015

fastqc_insert(){
   # First argument is the path to the raw FASTQ files
   pathRNA $1
   #Second argument is the path where the FastQC program
   FASTQC$2
   # Third argument is the path to where we write the output data
   out_path$3
   # Fourth argument is to define the name of the folder, this is useful
   to save the data before and after trimming in two different folders
   QC$4
   # Create the directory. We will write FASTQC output to a subdirectory
   called QC
   if[ ! -d$out_path];then
      echo"Making directory $out_path"
      mkdir$out_path
   fi
   if[!-d$out_path/$QC ];then
      echo"Making directory $out_path/$QC"
      mkdir$out_path/$QC
   fi
   for Dir in $(find$pathRNA -mindepth 1 -maxdepth 1 -type d );
   do
     i=$(basename $Dir);## is the name of the experiment, e.g., 666004
     echo "Calling FASTQC on $Dir for experiment $i"
     ## Call FASTQC for forward and reverse strands
     $FASTQC ${Dir}/${i}_1.fq.gz -o$out_path/$QC/
     $FASTQC ${Dir}/${i}_2.fq.gz -o $out_path/$QC/
   done
}
# call function, example
pathRNA=/home/server/RNAseqData
FASTQC=/home/user/FastQC_v0.11.5 /fastqc
out_path=/home/user/OSTEO/FASTQC_OUTPUT
QC="QC_raw"
fastqc_insert$pathRNA $FASTQC $out_path $QC
```

7. *Sequence Length Distribution module:* Some high throughput sequencers generate sequence fragments of uniform length as in our raw RNA-Seq data the length of the reads is 101. Even within uniform length library, our pipeline trims sequences to remove Illumina clips and poor-quality base calls from the end and across the reads, so the length distribution of the reads after trimming is not uniform. The module generates a graph showing the distribution of fragment sizes. For variable length FASTQ files this will show the relative amounts of each different size of sequence fragment. The Figure 3.4 compares the distribution of sequence length before and after trimming.

8. *Duplicate Sequences module:* In a diverse library most sequences will occur only once in the final set. A low level of duplication may indicate a very high level of coverage of the target sequence, but a high level of duplication is more likely to indicate some kind of enrichment bias (PCR over amplification). This module counts the degree of duplication for every sequence in a library and creates a plot showing the relative number of sequences with different degrees of duplication.

9. *Overrepresented Sequences module:* A normal high-throughput library contains a diverse set of sequences. Finding that a single sequence is very overrepresented in the set, either means that it is highly biologically significant, or indicates that the library is contaminated, or not as diverse as it is expected. This module lists all of the sequence which make up more than 0.1% of the total. For each overrepresented sequence, the program will look for matches in a database of common contaminants and will report the best hit it finds. Hits must be at least 20bp in length and have no more than 1 mismatch. However, it doesn't certainly give the source of the contamination, but indicates the right direction. It's also worth pointing out that many adapter sequences are very similar to each other so a reported hit isn't technically correct, but gives information about the type of this duplicated sequence.

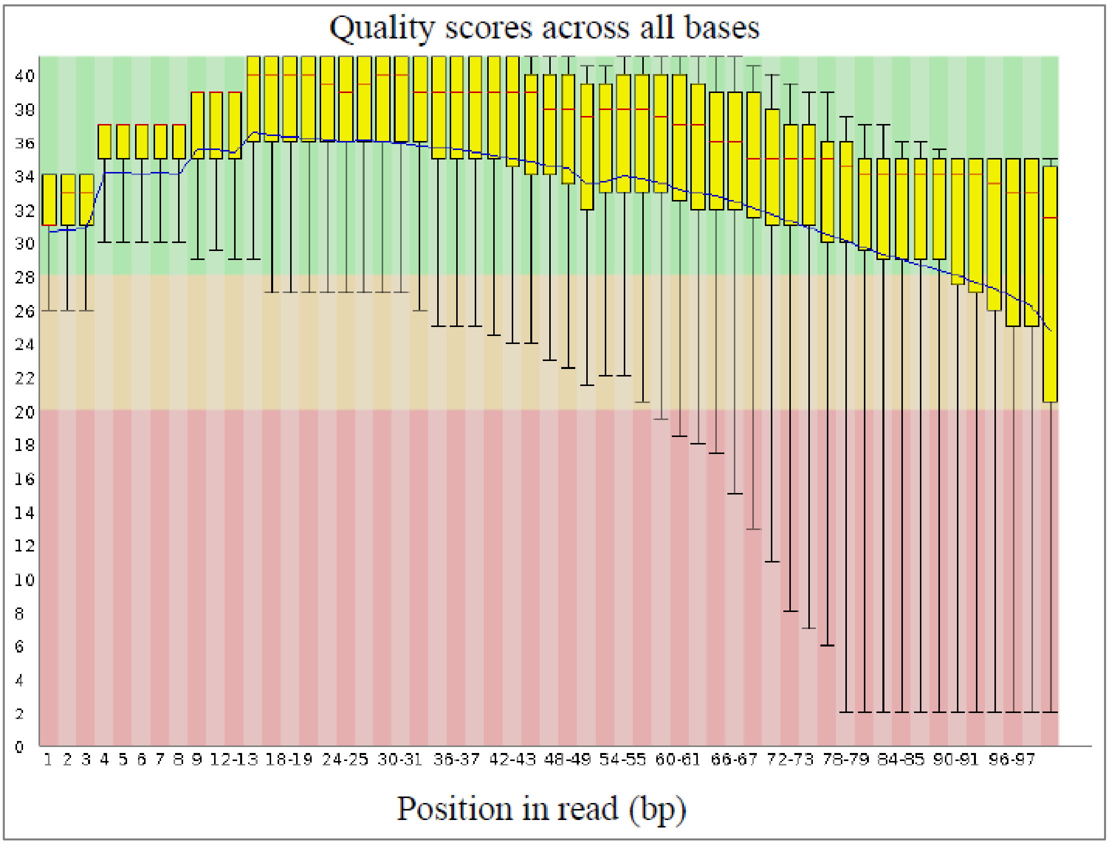

Figure 3.1: Per base sequence quality plot of raw reads in sample (Day3 replicate1 reverse strand)

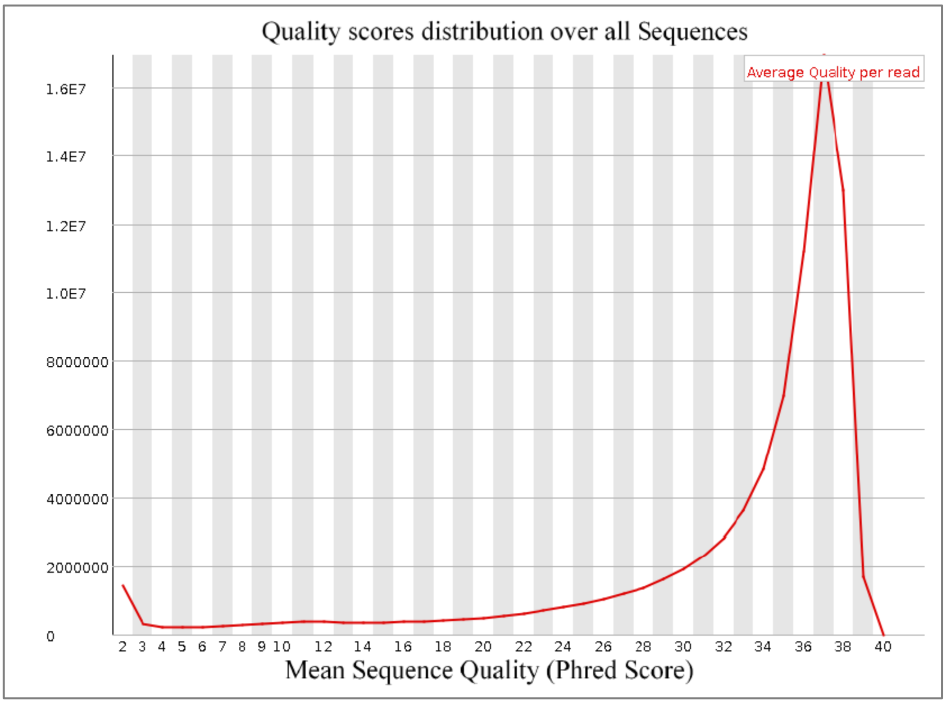

Figure 3.2: Quality scores distribution of raw reads in sample (Day0 replicate1forward strand)

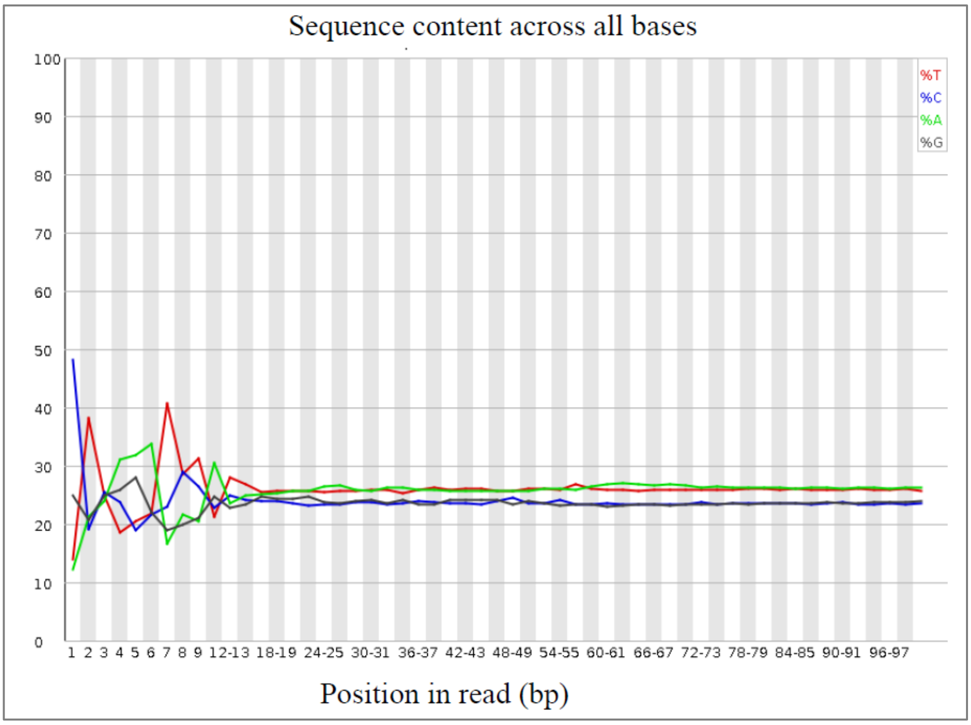

Figure 3.3: Per Base Sequence Content, the bias in the start positions of reads

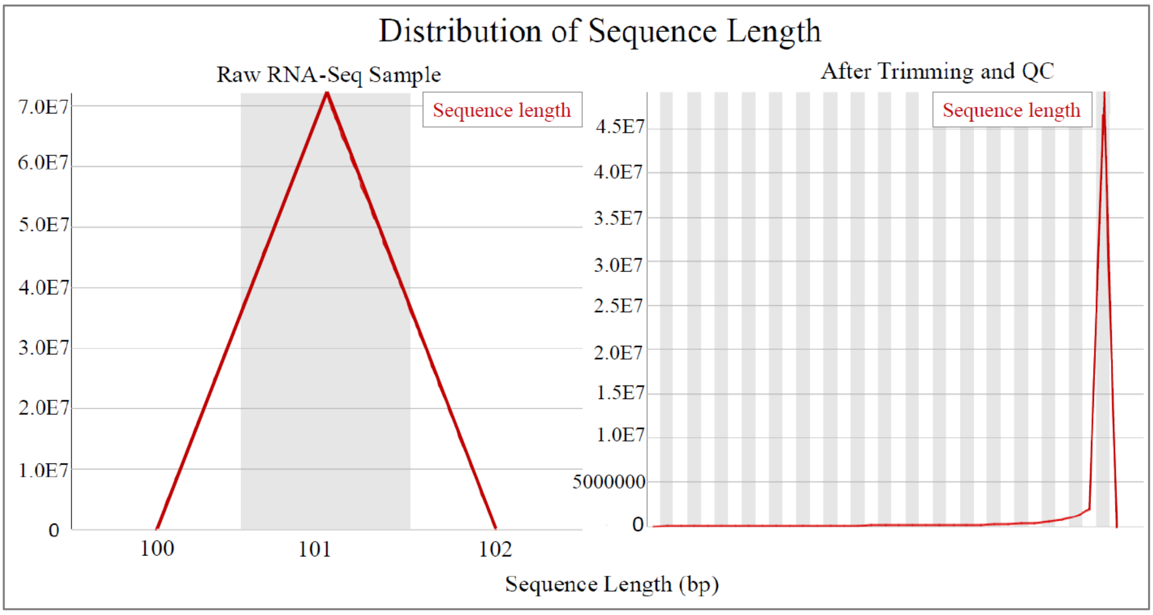

Figure 3.4: Distribution of sequence length, comparing raw reads length and distribution after trimming

10. *Adapter Content module:* The Kmer Content module will do a generic analysis of all the Kmers in the library to find those which do not have even coverage through the length of reads. This can find a number of different sources of bias in the library which can include the presence of read-through adapter sequences building up on the end of the sequences. One obvious class of sequences which might be wanted to analyze are adapter sequences. It is useful to know if the library contains a significant amount of adapter in order to be able to assess whether we need to adapter trim or not. This module therefore does a specific search for a set of separately defined Kmers and gives a view of the total proportion of the library which contain these Kmers. The plot itself shows a cumulative percentage count of the proportion of your library which has seen each of the adapter sequences at each position. Once a sequence has been seen in a read it is counted as being present right through to the end of the read so the percentages you see will only increase as the read length goes on.

11. *Kmer Content module:* The analysis of overrepresented sequences will point an increase in any precisely duplicated sequences. However, this analysis suffers from few problems which might fail:

    - If we have very long sequences with poor quality, then random sequencing errors will dramatically reduce the counts for exactly duplicated sequences.
    - If we have a partial sequence which is appearing at a variety of places within our sequence, then this won't be seen either by the per base content plot or the duplicate sequence analysis.

    There may be biological reasons why certain Kmers are enriched or depleted overall, but these biases should affect all positions within a sequence equally. This module therefore measures the number of each 7-mer at each position in the library and then uses a binomial test to look for significant deviations from an even coverage at all positions. Any Kmers with positionally biased enrichment are reported. The top 6 most biased Kmer are additionally plotted to show their distribution. All our samples failed in this test, because any individually overrepresented sequences, even if not present at a high enough threshold, will cause the Kmers from those sequences to be highly enriched in this module. These will normally appear as sharp spikes of enrichment at a single point in the sequence, rather than a progressive or broad enrichment. Libraries which derive from random primers will nearly always show Kmer bias at the start of the library due to an incomplete sampling of the possible random primers.

12. *Per Tile Sequence Quality module:* This graph is available only if we use an Illumina library (which is the case of our data) that retains its original sequence identifiers. Encoded in these is the flow-cell tile from which each read came. The graph shows the quality scores from each tile across all the bases to see if there was a loss in quality associated with only one part of the flow-cell. The plot illustrates the deviation from the average quality for each tile. The colours are on a cold to hot scale, with cold colours being positions where the quality was at or above the average for that base in the run, and hotter colours indicate that a tile had worse qualities than other tiles for that base.

    As we mentioned before there are three samples showed bad quality in some position of the reads. In the Figure 3.5, we can see which tiles failed to give the average quality of the sequenced reads. I think it was temporary problem caused by bubbles going through the flow-cell. I do not suppose it was permanent such as stains on the flow-cell or fragments inside the flow-cell lane, because we have in general, raw data with good quality.

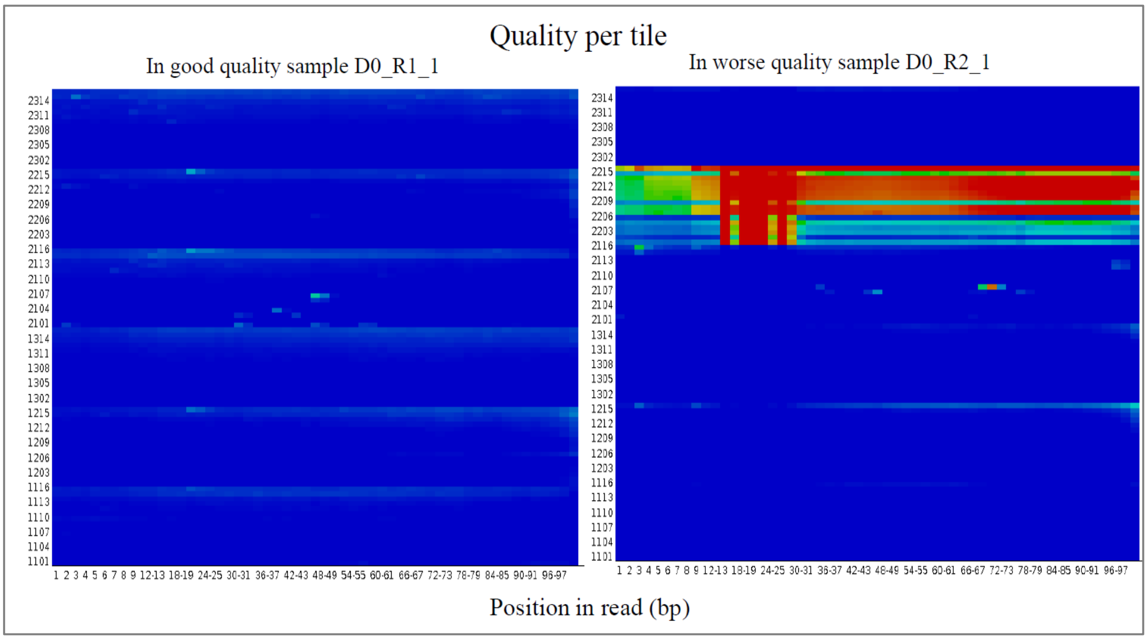

Figure 3.5: Quality per tile in good quality reads, and low-quality reads

3.2.2 IMPLEMENTATION OF READS TRIMMING

For the pair-end read, Trimmomatic requires as inputs both the reverse and forward reads and returns 4 outputs, 2 for the 'paired' output where both reads survived the processing, and 2 for corresponding 'unpaired' output where a read survived, but the partner read did not, as the Figure 3.6 illustrates. In the end, we used the paired reads.

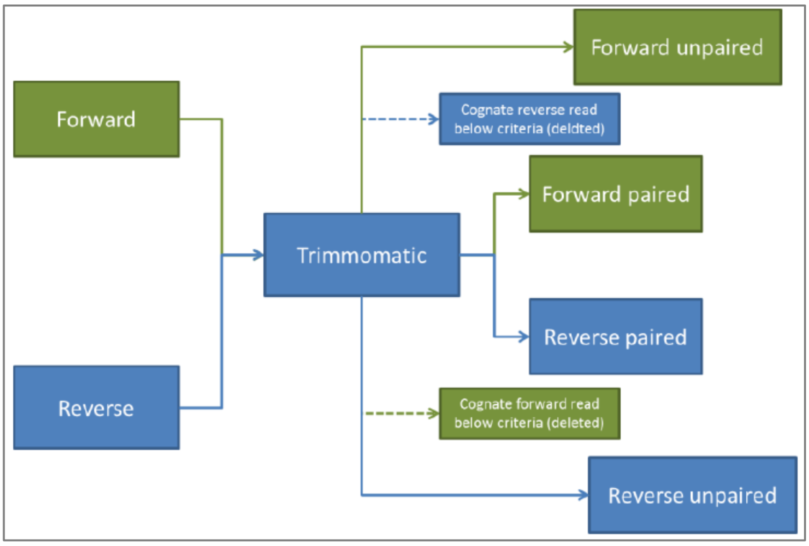
Forward unpaired
Cognate reverse read below criteria (deldted)
Forward
Forward paired
Trimmomatic
Reverse paired
Reverse
Cognate forward read below criteria (deleted)
Reverse unpaired

Figure 3.6: Flow of reads in Trimmomatic Paired End mode [68]

Trimming process must be performed in an order of steps then the optional procedures can be added in the end of the command. It is recommended in most cases that adapter clipping is done as early as possible, since correctly identifying adapters using partial matches is more difficult.

The syntax of calling Trimmomatic for pair-end reads, is as the following:[68]

```
java -jar <path to trimmomatic.jar> PE [-threads <threads>] [-phred33 | -phred64]
[-trimlog <logFile>] [-basein <inputBase> | <input 1><input 2>] [-baseout
<outputBase> | <paired output 1><unpaired output 1><paired output
2><unpaired output 2>]<step 1> ...
```

The main arguments passed to Trimmomatic in the command-line are as the following:

1. *-threads*: indicates the number of threads to use, here I used -threads 10.
2. *-phred33 or -phred64* : specifies the base quality encoding. If no quality encoding is specified, it will be determined automatically
3. *-trimlog:* followed by the path of a trimlog file creates a log of all read trimmings, indicating the following details:

   - The read name.
   - The surviving sequence length.
   - The location of the first surviving base
   - The amount trimmed from the start.
   - The location of the last surviving base in the original read.
   - The amount trimmed from the end.

4. *Input/ Output files*: Paired-end mode requires 2 input files (for forward and reverse reads), they can be zipped fastq files. and 4 output files (for forward paired, forward unpaired, reverse paired and reverse unpaired reads). They can be used either by explicitly naming the 2 input files, or by naming the forward file using the -basein flag, where the reverse file can be determined automatically.
5. *ILLUMINACLIP:* This step is used to find and remove Illumina adapters.Here is the syntax of this argument:

```
ILLUMINACLIP:<fastaWithAdaptersEtc>:<seed
mismatches>:<palindrome clip threshold>:<simple clip
threshold>:<minAdapterLength>:<keepBothReads>
```

Description of ILLUMINACLIP arguments:

- *fastaWithAdaptersEtc*: specifies the path to a fasta file containing all the adapters, PCR sequences etc. “TruSeq3-PE.fa” is the adapters file I used usually in the pipeline.
- *Seed Mismatches*: specifies the maximum mismatch count which will still allow a full match to be performed. I chose strict value of 2 mismatched bases maximum.
- *Palindrome Clip Threshold*: specifies the threshold of the full alignment score for plaindromic matches, how accurate the match between the two 'adapter ligated' reads must be for PE palindrome read alignment. I chose a score of 30 (about 50 bases). The alignment region can be 50 bases, therefore the scoring threshold can be in the range of 30, because each matching base increases the alignment score by 0.6, while each mismatch reduces the alignment score by Q/10.).
- *Simple Clip Threshold:* specifies how accurate the match between any adapter sequence must be against a read. I chose a score of 12, (about 20 bases).
- *minAdapterLength*: In addition to the alignment score, palindrome mode can verify that a minimum length of adapter has been detected. If unspecified, this defaults to 8 bases, for historical reasons. However, since palindrome mode has a very low false positive rate, this can be safely reduced, even down to 1, to allow shorter adapter fragments to be removed.
- *keepBothReads:* After read-though has been detected by palindrome mode, and the adapter sequence removed, the reverse read contains the same sequence information as the forward read, although in reverse complement. For this reason, the default behavior is to entirely drop the reverse read. By specifying „true“ for this parameter, the reverse read will also be retained, which may be useful e.g. if the downstream tools cannot handle a combination of paired and unpaired reads.

6. *SLIDINGWINDOW*:
   Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold. By considering multiple bases, a single poor quality base will not cause the removal of high quality data later in the read

```
SLIDINGWINDOW:<windowSize>:<requiredQuality>
```

- windowSize: specifies the number of bases to average across.
- requiredQuality: specifies the average quality required.

7. MAXINFO:
Performs an adaptive quality trim, balancing the benefits of retaining longer reads against the costs of retaining bases with errors.

```
MAXINFO:<targetLength>:<strictness>
```

- targetLength: This specifies the read length which is likely to allow the location of the read within the target sequence to be determined. I chose 50 bases.
- strictness: This value, which should be set between 0 and 1, specifies the balance between preserving as much read length as possible vs. removal of incorrect bases. A low value of this parameter (<0.2) favors longer reads, while a high value (>0.8) favors read correctness.

8. *LEADING:*

   Removes low quality bases from the beginning. As long as a base has a value below this threshold the base is removed and the next base will be investigated.
   quality: Specifies the minimum quality required to keep a base.

```
LEADING: <quality>
```

9. *TRAILING:*

   Remove low quality bases from the end. As long as a base has a value below this threshold the base is removed. This approach can be used removing the special Illumina “low quality segment” regions, however it is recommended Sliding Window or MaxInfo instead.

```
TRAILING:<quality>
```

10. *MINLEN:*

    Removes reads that fall below the specified minimal length. If required, it should normally be after all other processing steps. Reads removed by this step will be counted and included in the „dropped reads‟ count presented in the trimmomatic summary.
    length: Specifies the minimum length of reads to be kept.

```
MINLEN:<length>
```

To perform the trimming on a set of samples and keep the paired samples, we coded two functions in Bash script for this purpose. One to insert the samples to Trimmomatic (Code-box 3.2), and the second to keep merely the paired samples, which is used for mapping to the reference (Code-box 3.3). Reading through the codes in the Cod-boxes explain the simple concept used to get a function with one line to input a set of samples. Despite the simplicity of our functions, they provide useful service for users with low or no knowledge with shell command-line, which is necessary to run this part of RNA-seq pipeline.

3.2.3 PYTHON _ HTSEQ

HTSeq is a Python library. It offers parsers for many common data formats in High-Throughput Sequencing (HTS) projects, as well as classes to represent data, such as genomic coordinates, sequences, sequencing reads, alignments, gene model information and variant calls. It also provides data structures that allow for querying via genomic coordinates. [69]

Python as a scriptural language is useful to abstract information from output reports as text files. we wrote two scripts for this purpose; the first one to get the basic statistic of Fastq files as the length of reads, sequenced reads number, and %GC. The second script to get the mean reads quality in each sample from FastQC report. Using HTSeq library we coded a script to plot the mean quality of the reads across the position. For checking coverage uniformity across the gene body in Poly-A libraries, we proposed a method based on HTSeq to get the coverage distribution in different gene features.

Code-box 3.2: Bash Script. Trimmomatic insert function

```
#!/bin/bash
## This script encodes a function to input data to Trimmomatic
## Author Layal Abo Khayal 21.03.2015

trimo_insert (){
    # First argument is the path to the raw RNA-seq files
    pathRNA $1
    #Second argument is the path where the Trimmomatic program
    TRIMMO$2
    # Third argument is the path to where we write the output data
    pathTrimRNAout$3

    if[ ! -d$pathTrimRNAout];then
        echo"Making directory $pathTrimRNAout "
        mkdir$pathTrimRNAout
    fi
    for Dir in $(find$pathRNA -mindepth 1 -maxdepth 1 -type d );
    do
       i=$(basename $Dir);## is the name of the experiment, e.g., 666004
       echo "Calling Trimmomatic on $Dir for experiment $i"
       ## Make a separate directory for each output sample
       mkdir $pathTrimRNAout/${i}
       ## call trimmimatic
       java -jar ${TRIMMo}/trimmomatic-0.36.jar PE-threads10-phred33-
       trimlog$pathTrimRNAout/${i}.log $pathRNA/${i}/${i}_1.fq.gz
       $pathRNA/${i}/${i}_2.fq.gz$pathTrimRNAout/${i}/${i}_1_paired_output.fq.gz
       $pathTrimRNAout/${i}/${i}_1_unpaired_output.fq.gz
       $pathTrimRNAout/${i}/${i}_2_paired_output.fq.gz
       $pathTrimRNAout/${i}/${i}_2_unpaired_output.fq.gz
       ILLUMINACLIP:${TRIMMO}/adapters/TruSeq3-PE.fa:2:30:10:4:true LEADING:5
       MAXINFO:50:0.5 SLIDINGWINDOW:4:15 MINLEN:50
    done
}
# call function, example
pathRNA=/home/server/RNAseqData
TRIMO=/home/layal/Trimmomatic-0.36
out_path=/home/user/OSTEO/RNA_afterTrimming

trimo_insert $pathRNA $TRIMO $out_path
```

3.2.3.1 BasicStatistic Funnction:

Each FactQC output has in additional to html report, a plain text file called 'fastqc_data.txt'. In Osteoblast data set we have four differentiation time points (conditions) with three biological replicates for each condition, and each sample has two fastq files for forward and reverse reads, this means 24 FastQC reports in total. To extract the required information, we coded a useful function to read the data from such group of files. This function reads lines in a specific module (Basic Statistics module) in fastqc.data report, and then extract the required information about the reads, as sequence length, total sequenced numbers, encoding type, Sequences flagged as poor quality, and present of GC in the sequence. (Code-box 3.4)

Code-box 3.3 Bash Script, Function.to keep paired samples from Trimmomatic outputs

```
#!/bin/bash
## This script encodes a function to move the unpaired samples from the output of
Trimmomatic of to separate directory for further analysis and rename
"*_paired_output.fq " as "*.fq"
## Author Layal Abo Khayal 21.03.2015

trim_pair(){
   # First argument is the path to the output files of trimmomatic
   pathTrimRNAout $1
   # Second argument is the path to the directory where the unpaired samples will
   be moved
   pathunpairedRNA $2

   if[ ! -d${pathunpairedRNA}];then
mkdir${pathunpairedRNA}
   fi

   forDir in$(find$pathTrimRNAout-mindepth 1-maxdepth1-type d );
   do
     i=$(basename $Dir); ## is the name of the experiment, e.g., 666004
     echo"moving unpaired RNA from $Dir for experiment $i to $pathunpairedRNA "
     mkdir$pathunpairedRNA/${i}
     find$pathTrimRNAout/${i} -type f -name "*_unpaired_output.fq.gz"-exec mv
     {}$pathunpairedRNA/${i} \;
     echo"rename (_paired_output.fq.gz) files for easier uses as this (*_1.fq.gz)"
     find$pathTrimRNAout/${i}-type f -name "*1_paired_output.fq.gz"-exec mv
     {}$pathTrimRNAout/${i}/"${i}_1.fq.gz" \;
     find$pathTrimRNAout/${i}-type f -name "*2_paired_output.fq.gz"-exec mv
     {}$pathTrimRNAout/${i}/"${i}_2.fq.gz" \;
   done
}
# call function, example
pathTrimRNAout =/home/user/OSTEO/RNA_afterTrimming
pathunpairedRNA =/home/user/OSTEO/unpairedRNA

trim_pair $pathTrimRNAout $pathunpairedRNA
```

3.2.3.2 Mean Quality Function:

The function calculates the mean sequencing quality across all the bases in a sample. In FastQC report, there is module called “Per sequence quality scores”, which gives how many reads have a specific quality value (from 2 to 40). For example:

Quality	Count
2	1 290 108
3	314 493
37	14 853 205
40	160

We coded two functions for this task, the main mean quality function “QualityScore”, to calculate the mean quality of all bases in a fastq (Code-box 3.5). And “MQ_Av_Dataset” function, to get the mean quality of all bases in fastq files in a dataset, then calculate the average of mean quality of forward and reverse reads, eventually write the output as plaintext file (Code-box 3.6).

3.2.3.3 Plotting Phred Quality Along Reads Positions:

Using the FastqReader function from Python_HTSeq package, generates objects of class FastqReader from the Fastq files. In this object each read in Fastq is a SequenceWithQualities object, and has three feature slots:

- Name-feature returns the name of the read (read.name)
- Seq-feature returns the sequence of the read (read.seq)
- Qual-feature returns the quality of each base in the read as an array of values (read.qual)

The function "read_qual" iterates over reads in FastqReader object and calculates the mean quality of all reads in each position coordinate, as described in the comments (after # symbol, or between `''' '''`) in the (Code box 3.7). To accomplish this task, I coded another function "*pass_data_MQ"* to apply read_qual through a dataset (Code box 3.8). All the codes are reproducible and helpful to deal with quality control of fastq files.

Code-box 3.4. Python Script. Reads basic statistics function

```
__author__ = 'layal'
import os
import subprocess
import re
import sys
import socket
import zipfile
'''A function to get reads length and total sequenced reads from FastQC report'''
def BasicStatistic ( path ):
#read the zip file in QC which contains fastqc_data.txt
fp = zipfile.ZipFile(path, 'r')
    inModule= False
print ("Analysing file ", path)
for name in fp.namelist(): # iterate through all file names
if name.find('fastqc_data.txt')>=0:
         ## split file into lines fp:
for line in fp.read(name).split("\n"):
if (line.startswith('#')):
continue
             if (line.startswith(">>Basic Statistics")):
                inModule=True
continue
             if (line.startswith(">>END_MODULE")):
                inModule=False
if (inModule):
if (line.startswith('Total Sequences')):
                   Arr1 = line.rstrip().split('\t')
                   numberOfReads = Arr1[1]
if (line.startswith('Sequence length')):
                   Arr = line.rstrip().split('\t')
                   length = Arr[1]
return (numberOfReads, length)
'''A function to analyze a set of Fastqc output files and extract the basic
statistics of a fastq file'''
def AnalyzeDataset (path , outputname):
print ("Path" , path)
    F = open(outputname, 'w')
    F.write(' \n The length and number of read before trimming\n\n')
...
```

```
...
for dirPath, dirNames, fileList in os.walk(path):
for index, item in enumerate(fileList):
#print("item :", item)
if item.endswith("zip"):
print("item :", item)
                dpath= dirPath + '/' + item
print ("PATH ::::", dpath)
                pos = dpath.rfind('/')
                sample = dpath[pos+1:pos+6]
print ("Analyzing directory: " , sample)
                numReads, leng = BasicStatistic(dpath)
print ('number of reads = ', numReads ,'\n reads length = ', leng)
                F.write('Sample : '+ str(sample) + '\t\t Reads number : '
                        + str(numReads) +'\t\t Reads length :' + str(leng)+ '\n' )
enumerate
```

Code-box 3.5 Python Script. "QualityScore" function is to calculate the mean quality of all bases in a fastq file.

```
__author__ = 'layal'

import os
import subprocess
import re
import socket
import zipfile

def QualityScore( path ):
# The FastQC reports are store in a zip file for each sample
archive = zipfile.ZipFile(path, 'r')
    total_base=0
sumqual=0
meanqual=0
inModule= False
for name in archive.namelist(): # iterate through all file names in Zip archive
#  one of the files is 'fastqc_data.txt'
if name.find('fastqc_data.txt')>=0:
for line in archive.read(name).split("\n"): ## split file into lines
if (line.startswith('#')):
continue
                if (line.startswith(">>Per sequence quality scores")):
                    inModule=True
continue
                if (line.startswith(">>END_MODULE")):
                    inModule=False
continue
                if (inModule):
                    ar = line.rstrip().split('\t')
                    phred = int(ar[0])
                    num = float(ar[1])
sumqual+=phred*num
                    total_base+=num
                    meanqual=sumqual/total_base
return (meanqual)
```

Code-box 3.6. Python Script. “MQ_Av_Dataset” function

```
__author__ = 'layal'

'''A function to analyze a set of Fastqc output files and extract the mean quality
using QualityScore function, then calculate the average of mean bases qualities,
then write the outputs as plain text file

def MQ_Av_Dataset (path , outputname, Av_mean_output):
meanqual_samples = {}
print (path)
    F = open(outputname, 'w')
    F.write(outputname[0:-3] +'\n\n')
    H = open(Av_mean_output, 'w')
    H.write(Av_mean_output[0:-3]+'\n\n')
for dirPath, dirNames, fileList in os.walk(path):
for index, item in enumerate(fileList):
if item.endswith("zip"):   # pick out the Zip archives from FastQC
dpath= dirPath + '/' + item
                pos = dpath.rfind('/')
                sample = dpath[pos+1:pos+6]
print ("Analyzing directory: " , sample)
# calling QualityScore function to get mean quality of all bases in a sample
                mqual = QualityScore(dpath)
print ('mean quality = ' , mqual)
if sample in meanqual_samples:
                    meanqual_samples[sample] += mqual
else: meanqual_samples[sample] = mqual
# file contains all the mean quality of all samples
F.write('Sample: '+ str(item) + '\t\t\t mean quality: '
+ str(round(mqual,3))+'\n' )
enumerate
# to calculate the average mean quality forward and reverse RNAseq
for key, value in meanqual_samples.iteritems():
        meanqual_samples[key] = value / 2
print (key, meanqual_samples[key])
        H.write('Sample: '+ str(key) + '\t\t\t mean quality: '
+str(round(meanqual_samples[key],3))+'\n' )
```

Code-box 3.7. Python Script. “read_qual”function to get average quality in each base

```
__author__ = 'layal'
import HTSeq
import itertools
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import socket
#The purpose of this script is to generate plots of the reads mean quality. Using
the HTSeq python library.

'''
First, we set the paths to RNA-Seq data:
The raw fastq files and the files after trimming
'''
# Path to the raw FASTQ files. Exp:
pathraw="/media/layal/OSTEO/RNAseq_raw/"
# Path to Fastq fules after trimming.Exp:
pathAfterTrim= "/media/layal/OSTEO/RNA_afterTrimming/"

'''
Second, write the functions:
read_qual for calculate the mean quality.
plotfigure to set the legend parameters.

fq is FastqRead object produced after handling a FASTQ file to HTSeq.FastqReader
The function reurns (x,y)
x is a range of base position such as [1,2,3,...,n], where n is the length of the
sequence.
y is the average per-position quality (positions same as x)
read.qual is a range of int such as :
    [34 34 34 37 37 37 37 37 38 39 39 39 39 41 41 40 41 40 41 41 41 41 41 41 41
     40 40 41 41 41 41 40 40 41 41 41 41 41 40 40 40 41 41 40 41 38 40 38 38 39
     37 38 40 40 41 41 41 34 37 39 40 41 41 41 41 40 41 41 40 41 41 38 38 39 39
    39 39 37 37 37 35 37 33 35 36 34 35 35 35 35 35 35 35 36 36 36 36 35 35 35 35]

'''
def reads_qual (fq):
    readlen = 101
qualsum= np.zeros(readlen, np.int)
    nreads= np.zeros(readlen, np.int)
for read in fq:
if len(read)<len(qualsum):
           qualsum[:len(read)] += read.qual
           nreads[:len(read)] += 1
else:
#A range of sum quality in each position [331 335 329 366 370 370 366 368 389 388
386 ...]
qualsum += read.qual
            nreads += 1
print("number of reads :", nreads)
print("quality sum  :", qualsum)
# The quality average [34 34 34 37 37 37 37 37 38 39 39 39 39 41 41 ...]
y=qualsum/np.float_(nreads)
print "average of quality in one file  :", y
    x = range(1,len(qualsum)+1,1)
return(x,y)
...
```

*Code-box 3.8. Python Script. “*pass_data_MQ*” to apply read_qual through a dataset, plot the mean quality of bases across the reads positions*

```
...
def pass_data_MQ(path):
    Fx=0
i=0
for dirPath, dirNames, fileList in os.walk(path):
print('dirPath: ' , dirPath)
print(' fileList:' , fileList)
for index, samp in enumerate(fileList):
if(samp.endswith("_1.fq.gz")):
#forward reads sample
Sample_F = dirPath + '/' +samp
print " pass a forward sample to FastqReader" + samp
                fq_F = HTSeq.FastqReader(Sample_F)
                (Fx,Fy) = reads_qual(fq_F)
if(samp.endswith("_2.fq.gz")):
#Reverse reads sample
Sample_R = dirPath + '/' +samp
print " pass a reverse sample to FastqReader" + samp
                fq_R = HTSeq.FastqReader(Sample_R)
                (Rx,Ry) = reads_qual(fq_R)
enumerate
if(Fx==0):
continue
xr=Fx
        yr= (Fy +Ry)/float(2)
        i=i+1
plt.figure(1)
         # plot the mean quality of each sample
        plt.plot(xr ,yr ,color = colours[i],linestyle='solid',label=
lebels[i],linewidth=2.0)
plt.legend(loc='upper center', bbox_to_anchor=(0.56, 1.01),ncol=4, fancybox=True,
shadow=True,fontsize=14 )
# function to set parameters of the figure
def plotfigure( ftitle):
    plt.axis([0.0, 101, 20, 44])
#ax = plt.gca()
    #ax.set_autoscale_on(False)
plt.rc("font", size=16 , family ='serif')
    plt.title(ftitle, fontsize=26 , family ='serif')
    plt.ylabel('Phred score' ,fontsize=20 , family ='serif')
    plt.xlabel('Position in read' ,fontsize=20 , family ='serif')

lebels = ['Day0_1' , 'Day0_2' , 'Day0_3' ,'Day3_1' ,'Day3_2' , 'Day3_3' ,'Day6_1'
,'Day6_2','Day6_3' ,'Day12_1' , 'Day12_2' , 'Day12_3']
colours = [ (0,0,0) , (0,1,0), (0 ,0, 1), (0,1,1 ) , (1,0,0), (1,0,1), (0.5,0,0.5),
(0.5,0.5,0), (0.7,0,0.7), (0,0.4,0), (0.7,0.3,0), (0.9,0.6,0)]
# -RAW RNA-Seq reads
fig1 = plt.figure(1, figsize=(8,8), dpi=120)
plotfigure( 'Mean Read Quality of Raw Data')
pass_data_MQ(pathraw)
fig1.savefig( 'RawQuality' +'.png')
fig1.savefig( 'RawQuality'+'.pdf')
# After TRIMMING RNASeq
fig2 = plt.figure(2, figsize=(8,8), dpi=120)
plotfigure('Mean Read Quality of QC_processed Data')
pass_data_MQ(pathAfterTrim)
fig2.savefig( 'afterQC_quality' +'.png')
fig2.savefig( 'afterQC_quality'+'.pdf')
```

We generated 70 million reads with good high-quality scores as Figure 3.7 illustrates the mean quality of raw reads. However, after we implied quality control enhancement techniques, we got a very good quality score along the positions in the reads, as it is shown in the Figure 3.8, after trimming the mean quality of the bases at the end of the reads improved to be over 28 scores, and across the middle of the reads, looks more linear with Phred scores between 33-38.

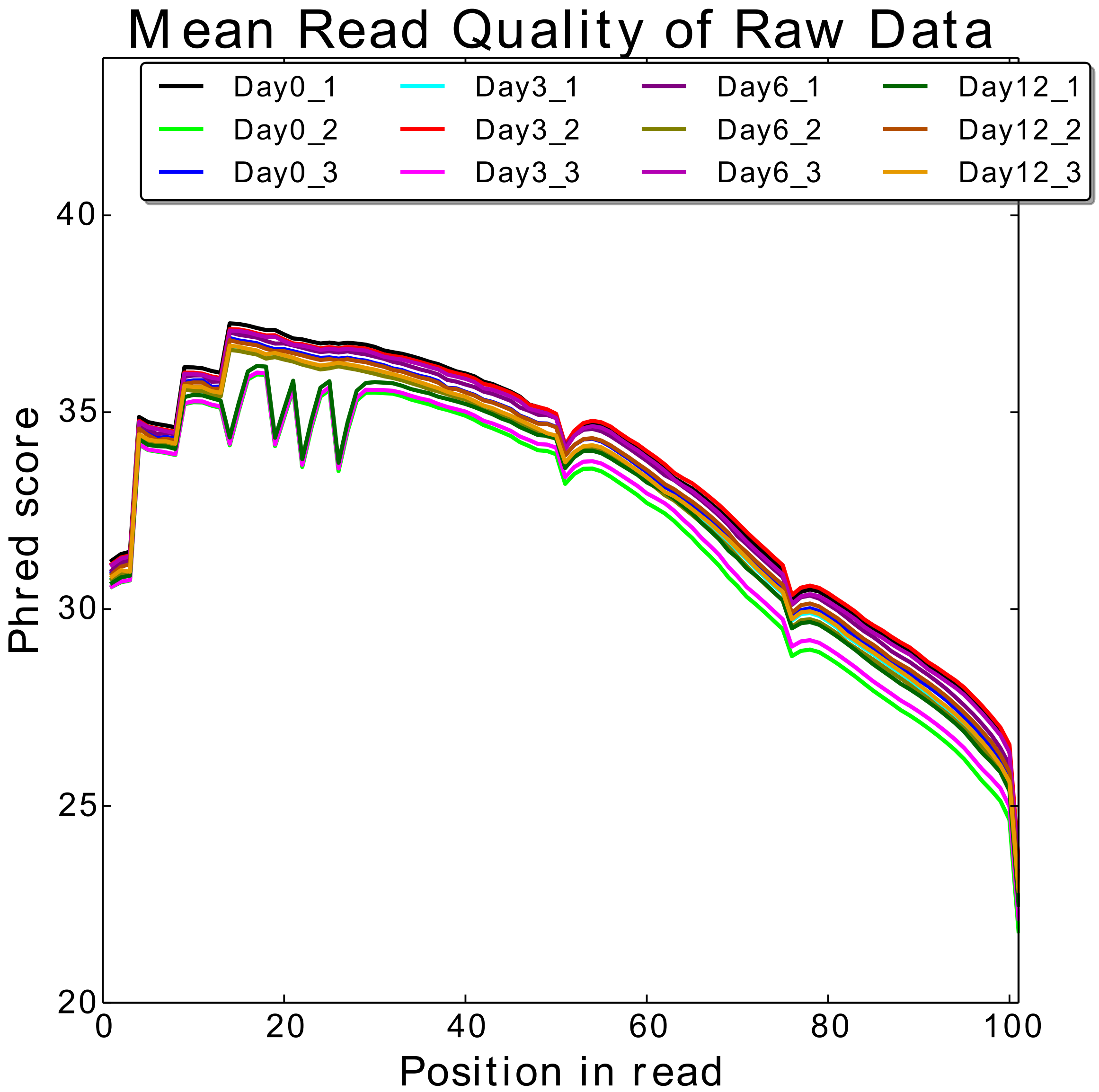

Figure 3.7: Mean Phred Quality Scores of Raw RNA-Seq Reads. The plot shows the mean Phred quality at each base position for all reads of the indicated sample. There are three replicate samples of each biological condition.

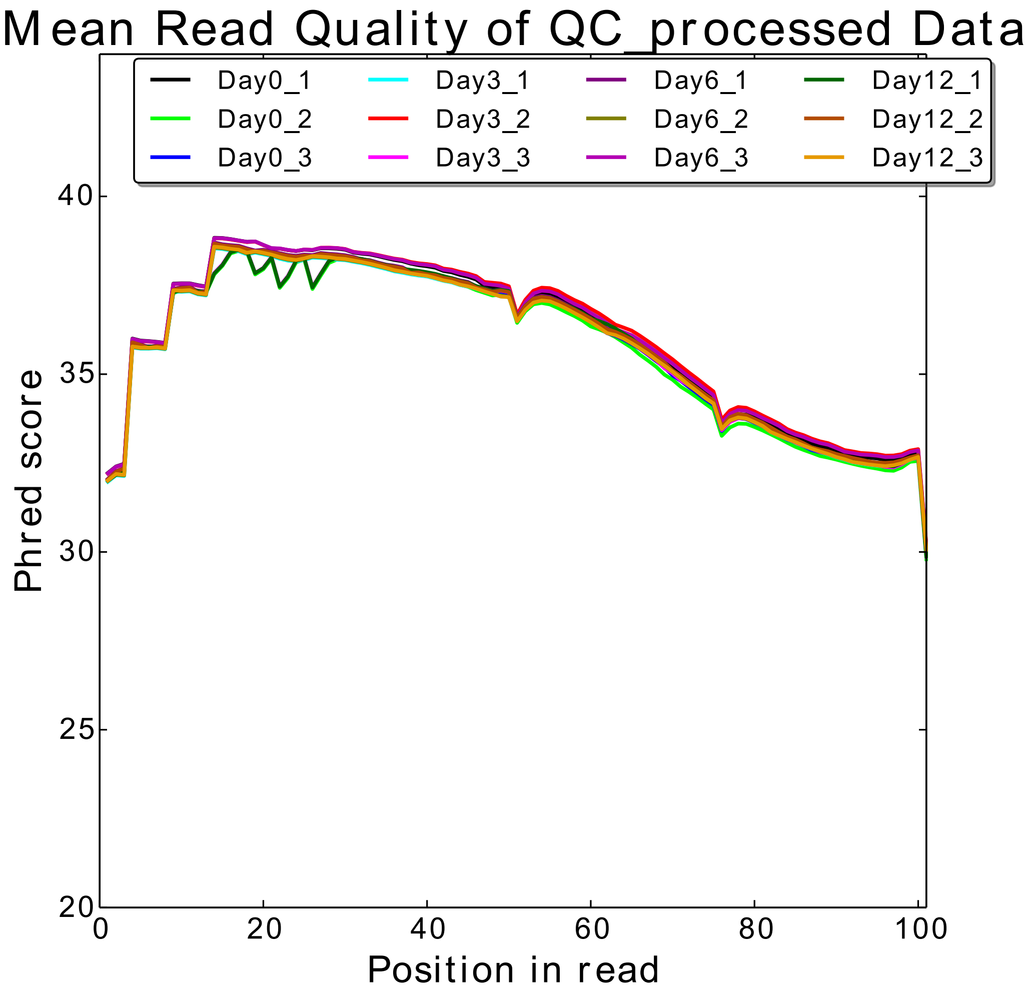

Figure 3.8: Mean Phred Quality Scores of RNA-Seq Reads after QC. The plot shows the mean Phred quality at each base position for all reads after trimming the Illumina clips and the low qualities.

3.2.3.4 Coverage Uniformity:

There is a concern that the coverage might vary across features of the RNA, since the used libraries are based on polyadenylated RNA. To provide an estimation of coverage across the genes feature, we separated the exons to 5' UTR exons, 3' UTR exon and the exons in the translated regions in between, as illustrated in the Python script "Separate Exons according to the translation regions" (Code-box 3.9). Hence, we got 3 annotation GTF files of first (5' UTR), last (3' UTR) and middle exons in the genes. Then we coded a function to calculate the coverage of the reads using HTSeq library. Accomplishing this task was in the following steps:

1. Create GTF objects of the first exons, last exons and the middle exons from the gtf files generated in the previous step, by using GFF_Reader method from HTSeq library. (Code-box 3.11)
2. Create alignment objects from BAM files, the outputs of TopHat Aligner, by using BAM_Reader method from HTSeq library. The BAM_Reader object yields for each alignment line in the BAM file an object of class BAM_Alignment. Every alignment object has a slot read, that contains a SequenceWithQualities object as described previously. (The read object has three features: read.name, reads.seq and read.qual).

Furthermore, every alignment object aln has a slot iv (for “interval”) that shows the positions on the genome where the read was aligned to (if it was aligned). This feature slot of the alignment object holds information of the start and end coordinates on the genome of that align object, the chromosome name, and the strand where that alignment located. (Code-box 3.10)

```
>> aln.iv
[GenomicInterval object 'IV', [246048,246084),
strand '+']
>> aln.iv.chrom
'IV'
>> aln.iv.start
246048
>> aln.iv.end
246084
>> aln.iv.strand
'+
```

3. Creating a GenomicArray data structure to store and retrieve information associated with a genomic position or genomic interval. The key of the GenomicArray is an genomic interval, which we retrieve from "align.iv" of each alignment object and simply iterate through all the reads and add the value 1 at the interval when each read was aligned.
4. To calculate the mean coverage of each aligned read, we passed the GenomicArray "ga" of the interval of each exon in the aligned read "iv" to list method, so we got a coverage values vector, then we calculated the mean of this vector.
5. Eventually we plotted the coverage density of 5' UTR, 3' UTR and exons within coding regions. As shown in Figure 3.9.

Code-box 3.9. Python Script. Separate Exons according to the translation regions to gtf files

```
__author__ = 'layal'
import HTSeq
import itertools
import numpy as np
import os
# here we create new empty files and assign them to file object
F = open('path/first_exons_GRCm38.gtf', 'w')
L = open('path/last_exons_GRCm38.gtf', 'w')
M = open('path/middle_exons_GRCm38.gtf', 'w')

'''
Extract a gene id from a GTF line.
The assumption is that the 9th [8] field has a string like this:
gene_id "ENSMUSG00000102693"; gene_version "1"; gene_name "4933401J01Rik";
gene_source "havana"; gene_biotype "TEC"; havana_gene "OTTMUSG00000049935";
havana_gene_version "1";
gene_id "ENSMUSG00000064842"; gene_version "1"; gene_name "Gm26206"; gene_source
"ensembl"; gene_biotype "snRNA";
the first element [0] in the line split by ';' is the id
the second element [1] in the id part split by " " is the id
In this example, we would return the String
"ENSMUSG00000102693","ENSMUSG00000064842"
'''
…
```

```
...
def get_geneID(line):

    Ar = line.split('\t')
    id = Ar[8].split(';')[0]
    gene_id = id.split('"')[1]
return (gene_id)

plus_strand = dict()
reverse_strand = dict()

## The following iteration puts the exon definition lines into the plus_strand and 
minus_strand dictionaries
'''
the lines in GTF file look like this, in the third field is written the type of 
feature, either gene, transcript, exon ….:
1  havana gene   3073253    3074322    .  +  .  gene_id "ENSMUSG00000102693"; 
gene_version "1"; gene_name "4933401J01Rik"; gene_source "havana"; gene_biotype 
"TEC"; havana_gene "OTTMUSG00000049935"; havana_gene_version "1";
1       ensembl_havana           exon     3214482  3216968  .       -        .
        gene_id "ENSMUSG00000051951"; gene_version "5"; transcript_id 
"ENSMUST00000070533"; …
'''
with open(path) as gtf:
 for line in gtf:
 if line.startswith("#!"):
 continue
 else:
   arr = line.split('\t')
 if arr[2] =='exon':
    arr_exon = line.split('\t')
 if arr_exon[6]== '+':
    gene_idP = get_geneID(line)
 if gene_idP in plus_strand:
    plus_strand[gene_idP].append(line)
 else:
    plus_strand[gene_idP] = [line]
 elif arr_exon[6]== '-':
      gene_idR = get_geneID(line)
 if gene_idR in reverse_strand:
    reverse_strand[gene_idR].append(line)
 else:
    reverse_strand[gene_idR]= [line]

## Now write the corresponding data into the three output files separate the exons
for key in plus_strand:
    ar = plus_strand[key]
    ln = len(ar)
F.write(ar[0])
    L.write(ar[ln-1])
for i in range(1, ln-2):
        M.write(ar[i])
for key in reverse_strand:
    ar1 = reverse_strand[key]
    ln = len(ar1)
    F.write(ar1[ln-1])
    L.write(ar1[0])
for i in range(1, ln-2):
        M.write(ar1[i])
```

Code-box 3.10. Python Script function to calculate the coverage of the exons for one BAM file:

```
_author__ = 'layal'
import os
import numpy as np
import HTSeq

''' This function calculate the coverage of exon feature from an alignment Bam
file'''
def calculateCoverageOfOneBamFile(gtf_file, Alignment, filename_cv):
    C = open(filename_cv, 'w')

# I create a GennomicArray data structure to store and retrieve information
associated with a genomic position or genomic interval.
# I instruct the GenomicArray to add chromosome vectors as needed, by specifying
"auto".
# the stranded is false, because the direction is not relevant
# typecode='i' because we are saving integers in the genomic Array.

print "Asign Genomic array"
cvg= HTSeq.GenomicArray("auto" , stranded=False, typecode = 'i')
#I count the coverage by adding one to the interval of the read which is aligned,
cvg is a GenomicArray object
print "looping over the aligned object to set values in the Genomic array"
for align in Alignment:
if align.aligned:
       cvg[align.iv] +=1
# now we have the interval of the aligned read assigned by adding 1 each time,
# gtf file is an class object from GFF_Reader

 MeanCoverage = []
 print "looping through the features in each gtf file"
for feature in gtf_file:
if feature.type == "exon":
       mv = np.mean(list(cvg[ feature.iv ]))
       C.write(feature.name + "\t" + format(str(mv) + "\n"))
       MeanCoverage.append(mv)
return(MeanCoverage)
```

Code-box 3.11. Python Script. Get the coverage uniformity of Osteoblast dataset

```
__author__ = 'layal'
#this script calcules the coverage for the 12 samples by using the function of
calculatecoverge
import HTSeq
import os
from calculateCoverage import calculateCoverageOfOneBamFile

#Set paths where is your data live
root= '/path/OSTEO/Align_Tophat_GRCm38/'
# the gtf files of separated exons according to the translation regions
gtf_files = '/path/OSTEO/coverage/gtf_file'
#the output coverage of samples from this script
outCov_path=  '/path/OSTEO/coverage/coverage_output/'
# create a directory for the outputs
if not os.path.exists(outCov_path):
    os.makedirs(outCov_path)
#alignment samples
BAMname =[66604 , 66605 , 66606, 66607, 66608, 66609, 66610, 66611, 66613, 66614,
66615, 66616]
...
```

```
...
filesNameBase =['first_exon_cov_' , 'middle_exon_cov_', 'last_exon_cov_']

print "Read the gtf of the first exon"
gtf_file_firstExons = HTSeq.GFF_Reader(gtf_files+'first_exons_GRCm38.gtf')

print "Read the gtf of the last exon"
gtf_file_lastExons = HTSeq.GFF_Reader(gtf_files+'last_exons_GRCm38.gtf')

print "Read the gtf of the middle exon"
gtf_file_middleExons = HTSeq.GFF_Reader(gtf_files+'middle_exons_GRCm38.gtf')
#vector of all the gtf object
gtfFiles= [gtf_file_firstExons, gtf_file_middleExons, gtf_file_lastExons]

# looping across the samples
for k, j in zip(gtfFiles , filesNameBase) :
for i in BAMname:
        filename_cv =outCov_path + j +'%d.txt' %(i)
print "creat alignment  object : "+'%d'%(i)
        Alignment = HTSeq.BAM_Reader(root+ str(i) + '/accepted_hits.bam')
print "calculate the coverage from the annotation of " + j[:10] +
" of the aligned sample :" +'%d'%(i)
        meanCov = calculateCoverageOfOneBamFile(k ,Alignment ,filename_cv)
        cov = open(outCov_path+'meanCov_'+j[:5]+str(i), 'w')
        cov.write(str(meanCov)+'\n')
```

From the density distribution of the base mean which is the sequencing depth of each exon in Osteoblast dataset (Figure 3.9), we found that both 3′ UTR exons and the intervening coding sequences have a similar distribution of base mean values compared to 5′ UTR sequences. Although this slight change in the distribution across features, the level of coverage was similar for each, suggesting that the libraries capture transcriptional complexity across different elements of genes.

3.3. READS MAPPING USING TOPHAT ALIGNER:

TopHat2 improved the performance of TopHat by mapping reads against the known transcriptome, this improves the overall sensitivity and accuracy of the mapping. as illustrates in pipeline schemes (Figure 3.10) [85]. The mapping procedure of TopHat2 consists of three major parts, optional transcriptome alignment (step 1), genome alignment (step 2), and spliced alignment (steps 3–6). Paired-end reads are aligned individually first, and then combined to paired-end alignments by taking into account the fragment length and orientation.

1. If annotation information is available, TopHat2 aligns reads to the transcriptome first. It extracts transcript sequences from the Bowtie2 genome index using a GTF/GFF file. Bowtie2 is then used for indexing this virtual transcriptome and aligning reads to it.

2. The reads that did not fully align to the transcriptome are aligned to the genome with Bowtie2. At this stage, the reads which mappable contiguously to one exon will be mapped, while multiexon spliced reads will not.

3. The unmapped reads are split into short segments (25 bp by default) and mapped to the genome again. If TopHat2 finds reads where the left and the right segment map within a user-defined maximum intron size, it maps the whole read to that genomic region, to find potential splice sites containing known splice signals (GT-AG, GC-AG, or AT-AC). TopHat2 also looks for indels and fusion break points at this step.

4. Genomic sequences bound the potential splice sites are joined and indexed, and unmapped read segments are aligned to this junction flanking index with Bowtie2.

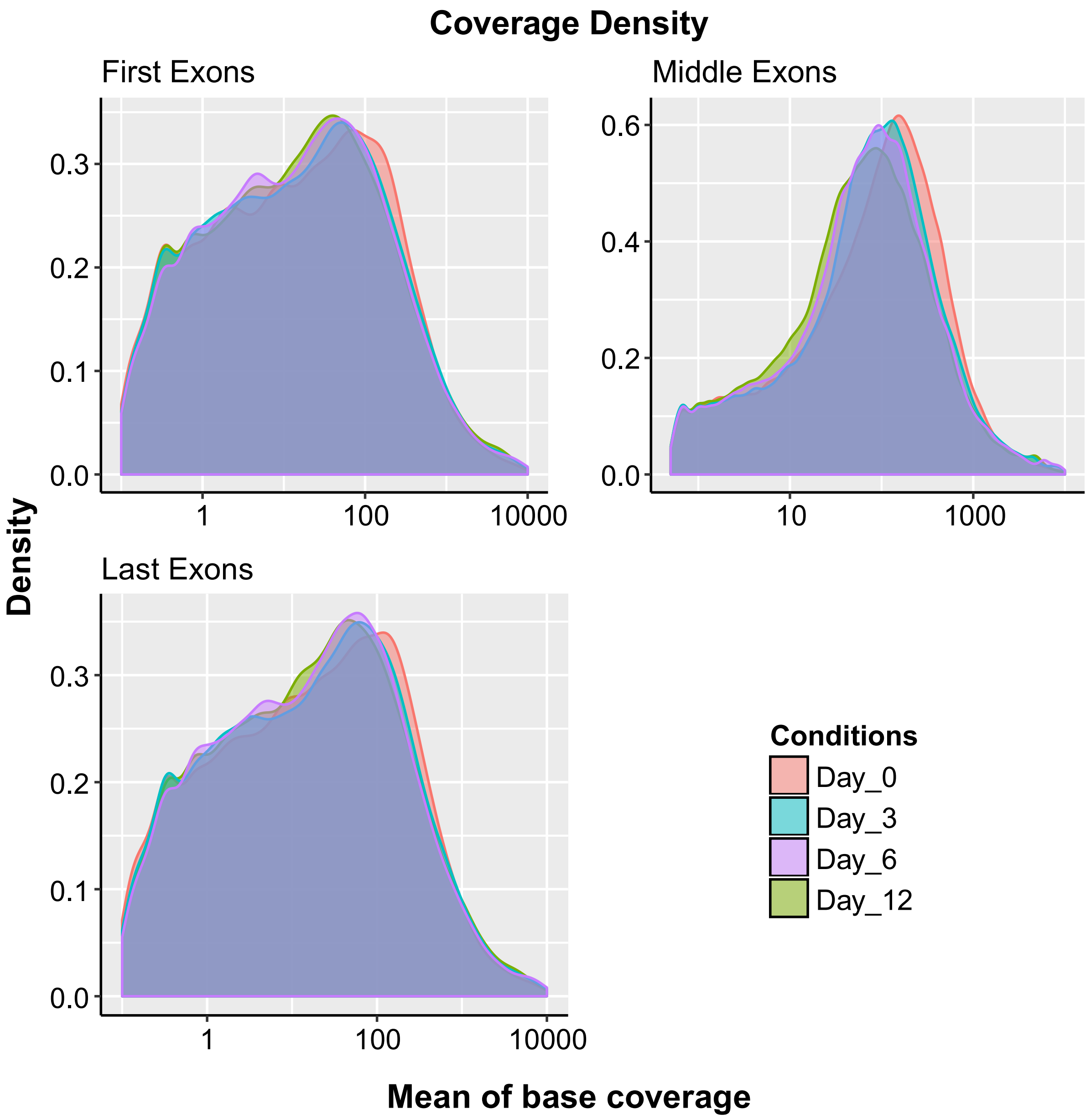

Figure 3.9: Coverage density for RNA-seq reads features. The plot shows three coverage density distribution of the base mean in the first exons (5' UTR exons), middle exons (within translation regions) and the last exons (3' UTR exons). The base mean is shown on the x-axis on a log10 scale for each of these features.

5. Segment alignments from steps 3 and 4 are stitched together to form whole read alignments.
6. Reads that extended a few bases into an intron in step 2 are realigned to exons using the new splice site information.
7. In order to decide which alignments to report for multimapping reads, TopHat2 recalculates their alignment score taking into account how many reads support the splice junctions, indels, etc.

To perform alignment by TopHat 2, first we need to prepare the reference genome index. Bowtie2 reference genome indexes are available for many organisms at the Bowtie2 website [119] and the Illumina iGenomes website [120]. However, it is better to build the reference index to be sure that genome index/FASTA files are from the same provider as GTF files and the same version. It is easy to build the index using bowtie2-build command.

When calling bowtie2-build command, we need to take into account the following notes: call the option -f to indicate that the reference is FASTA file with “.fa” extension (or .mfa , .fna), and the fasta file must be unzipped. The Code-box 3.12 shows the command to build reference index.

Code-box 3.12 Building genome reference index

```
#!/bin/bash
# build the index of GRCm38 from Ensembl, the fasta file must be unzipped
# option -f The reference input files are FASTA files (having extension.fa)
# unzip the genome reference
gunzip -k Mus_musculus.GRCm38.dna.toplevel.fa.gz.
#Run Bowtie2-build command
bowtie2-build -f /home/layal/OSTEO/Bowtie2_index/Mus_musculus.GRCm38.fa
/home/layal/OSTEO/Bowtie2_index/Mus_musculus.GRCm38
```

TopHat2 accepts both FASTQ and FASTA files as input. Read files can be compressed (.gz), but tarballs (.tgz or .tar.gz) need to be opened to separate files. TopHat2 can also combine single-end reads in a paired-end alignment if needed [121]. Our Osteoblast samples are paired-end reads, so we used TopHat syntax of paired-end reads as stated in Code-box 3.13.

we used the default syntax of TopHat, but I added few options assuming their necessity in our experiment.

- `-p 30`to used 30 server-processors simultaneously to speed up the process.
- `-G` to provide GTF/GFF genes model annotation file to TopHat to build the transcriptome index .
- `--b2-sensitive` it is a Bowtie option to perform the alignment more accurate and sensitive but slower.
- `--keep-fasta-order`to sort alignments in the same order in the genome fasta file.

TopHat produces several result files [121]:

- accepted_hits.bam contains the alignmentsin BAM format. The alignments are sorted according to chromosomal coordinates.
- junctions.bed contains the discovered exon junctions in BED format [122]. A junction consists of two blocks, where each block is as long as the longest overhang of any read spanning the junction. The score is the number of alignments spanning the junction.
- insertions.bed contains the discovered insertions, `-chromLeft`refers to the last genomic base before the insertion.
- deletions.bed contains the discovered deletions `-chromLeft` refers to the first genomic base of the deletion.
- align_summary.txt reports the alignment rate and how many reads and pairs had multiple alignments. For example, the following summary is produced to osteoblast Day0 replicate1 sample:

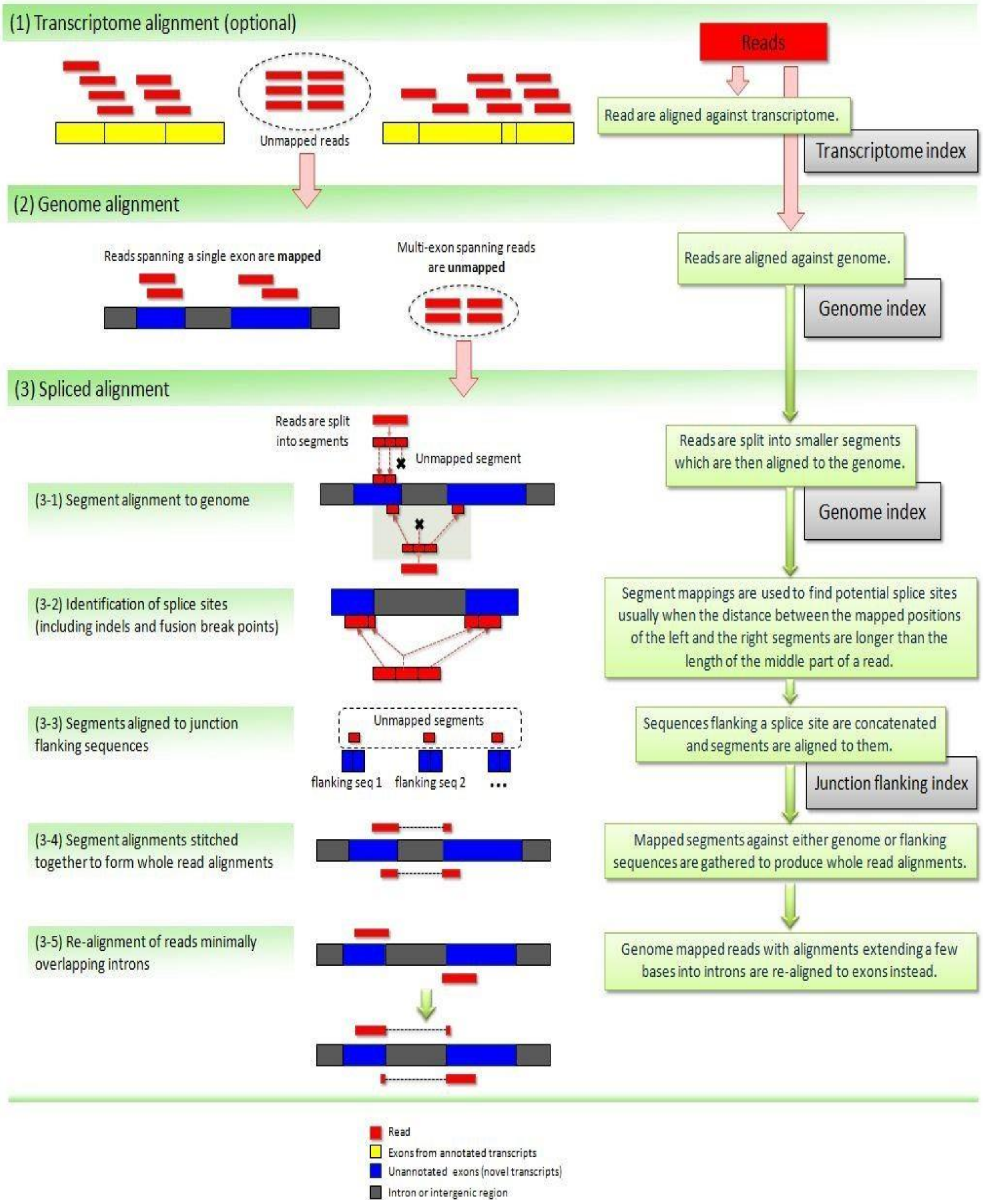
(1) Transcriptome alignment (optional)
Reads
Unmapped reads
Read are aligned against transcriptome.
Transcriptome index
(2) Genome alignment
Reads spanning a single exon are mapped
Multi-exon spanning reads are unmapped
Reads are aligned against genome.
Genome index
(3) Spliced alignment
Reads are split into segments
Unmapped segment
Reads are split into smaller segments which are then aligned to the genome.
(3-1) Segment alignment to genome
Genome index
(3-2) Identification of splice sites (including indels and fusion break points)
Segment mappings are used to find potential splice sites usually when the distance between the mapped positions of the left and the right segments are longer than the length of the middle part of a read.
(3-3) Segments aligned to junction flanking sequences
Unmapped segments
flanking seq 1
flanking seq 2
...
Sequences flanking a splice site are concatenated and segments are aligned to them.
Junction flanking index
(3-4) Segment alignments stitched together to form whole read alignments
Mapped segments against either genome or flanking sequences are gathered to produce whole read alignments.
(3-5) Re-alignment of reads minimally overlapping introns
Genome mapped reads with alignments extending a few bases into introns are re-aligned to exons instead.
Read
Exons from annotated transcripts
Unannotated exons (novel transcripts)
Intron or intergenic region

Figure 3.10 TopHat2 pipeline [85]

```
Left reads:
          Input     :  59034966
           Mapped   :  58038832 (98.3% of input)
            of these:   2109266 ( 3.6%) have multiple alignments (39454 have >20)
Right reads:
          Input     :  59034966
Mapped   :  57881402 (98.0% of input)
of these:   2100416 ( 3.6%) have multiple alignments (39342 have >20)

98.2% overall read mapping rate.

Aligned pairs:  57151426
     of these:   2059667 ( 3.6%) have multiple alignments
                  725517 ( 1.3%) are discordant alignments
95.6% concordant pair alignment rate.
```

In the following is the bash script code for calling TopHat2 on a dataset without concerning about inserting the correct name of each single sample.

Code-box 3.13 TopHat inserting function

```
!/bin/bash

# The purpose of this script is to run TopHat2 on a dataset
# Input: the trimmed RNAseq files
# Input: the annotation file
# Input: bowtie2 index files
# Output: Mapped reads
tophat_insert (){

    # First argument is the path to the trimmed RNA-seq files
    RNAseq_data$1
    #Second argument is the path to the gene-annotation file
    geneAnnot$2
    # Third argument is the path to bowtie2 index files
    genomeIndex$3
    # fourth argument is the path to TopHat output
    tophat_out$4
    # the path to TopHat tool, it should be the last stable version
    TOPHAT=/path/tophat2

    if[ ! -d${tophat_out}];then
    mkdir${tophat_out}
    fi
    #Map the reads for each sample to the reference genome:
    for Dir in $(find$RNAseq_data -mindepth 1 -maxdepth 1 -type d );
    do
    i=$(basename $Dir); ## is the name of the experiment, e.g., 666004
        echo "Calling tophat on $Dir for experiment $i"
        mkdir $tophat_out/${i}
    ${TOPHAT}-p 30 -G $geneAnnot --b2-sensitive --keep-fasta-order -o
    $tophat_out/${i}$genomeIndex $RNAseq_data/${i}/${i}_1.fq.gz
    $RNAseq_data/${i}/${i}_2.fq.gz
    done
}
```

```
...

# example for calling the tophat_insert function on osteoblast dataset

BASE=/home/layal/OSTEO
geneAnnot=$BASE/GeneAnnotation/Mus_musculus.GRCm38.86.gtf
genomeIndex=$BASE/Bowtie2_index/Mus_musculus.GRCm38
tophat_out=$BASE/Align_Tophat_GRCm38
RNAseq_data=$BASE/RNA_afterTrimming

tophat_insert$RNAseq_data $geneAnnot $genomeIndex $tophat_out
```

3.4. DIFFERENTIAL GENE EXPRESSION ANALYSIS

3.4.1 METHOD

The fundamental process in RNA-Seq data analysis and transcriptome characterization, is to define a set of genes that have significant expression variance between conditions in an experiment. AS we described briefly in the section “General concepts of statistical distributions in RNA-Seq data”, there are various algorithms for differential gene expression analysis.

The comparative analysis of transcriptomic data in our research based on DESeq2 method [123], that takes a count matrix as an initial input. The count matrix composed of *n* rows; one row for each gene *I*, and *m* columns; one column for each sample j. The matrix elements K_{ij} indicate the number of sequencing reads that have been unambiguously mapped to a gene in a sample. As described in the section “Assigning sequence reads to genomic features”.

The read count K_{ij} for gene *i* in sample *j* is modeled with a generalized linear model (GLM) [124] of the negative binomial family with a logarithmic link. Read counts K_{ij} follow a negative binomial distribution (a gamma-Poisson distribution) with mean μ_{ij}, the variance σ_{ij}^2, and dispersion α_i.

$$K_{ij} \sim NB\left(\text{mean} = \mu_{ij}\text{dispersion} = \alpha_i\right) \text{ [123]}$$

The mean parameter μ_{ij} is the expectation value of the observed counts for gene *i* in sample *j*, it is the product of a quantity q_{ij}, proportional to the concentration of cDNA fragments from the gene in the sample, and a normalization factor s_{ij}:

$$\mu_{ij} = s_{ij}.q_{ij}$$

The GLM fit returns coefficients indicating the overall expression strength of the gene and the log2 fold change between the conditions. DESeq2 uses GLMs with a logarithmic link:

$$\log_2 q_{ij} = \sum_r x_{jr}.\beta_{ir}$$

Where x_{jr} are design matrix elements, and β_{ir} coefficients.

Using linear models provides the flexibility to analyze more complex experimental designs, to serve the genomic studies.

To estimate the size factors, that are used for count normalization, we take the median of the ratios of observed counts. As the following equation shows. normalization constants s_{ij} may differ from gene to gene.

$$s_{ij} = \underset{i}{\text{median}} \frac{K_{ij}}{\left(\prod_{j=1}^{m} K_{ij}\right)^{1/m}}$$

Usually in RNA-Seq experiment the sample sizes tend to be quite small (in most case 2 or 3 replicates for each condition), this cause highly variable dispersion estimates for each gene, which is considered as noisy estimates “shot noise”, and would compromise the accuracy of differential expression testing. The variability between replicates, is modeled by the dispersion parameter α_i, which describes the variance of counts by:

$$\text{Var}\, K_{ij} = \mu_{ij} + \alpha_i \mu_{ij}^2$$

The dispersion parameter α_i follows a log-normal prior distribution that t s centered around a trend which depends on the gene’s mean normalized read count:

$$\bar{\mu_i} = \frac{1}{m} \sum_j \frac{K_{ij}}{s_{ij}}$$

The Estimation of dispersions performed by DESeq2 in three steps, which implemented based on empirical Bayes approach. First each gene is treated separately to get gene-wise dispersion estimates by using maximum likelihood (MLE) of the read count of each individual gene. Next determine the trend parameter of the dispersion distribution of these estimates; to allow for dependence on average expression strength. The last step is to get the final dispersion estimation by combining the likelihood with the trended prior to get maximum a posteriori (MAP) values as final dispersion estimates. Details for the dispersion estimation can be found in DESeq2 paper and supplements [123]. DESeq2 estimates the width of the prior distribution from the data, and therefore automatically controls the amount of shrinkage based on the observed properties of the data. The shrinkage procedure helps avoid potential false positives, which can result from underestimates of dispersion, or because some genes may show much higher variability than others for biological or technical reasons.

To get the logarithm fold change LFC of each gene between conditions, DESeq2 employs an empirical Bayes procedure too; first it performs ordinary GLM fits to obtain maximum-likelihood estimates (MLEs) for the LFCs and then fit a zero-centered normal distribution to the observed distribution of MLEs over all genes. This distribution is used as a prior on LFCs in a second round of GLM fits, and the MAP estimates are kept as final estimates of LFC. Furthermore, a standard error for each estimate is reported, which is derived from the posterior’s curvature at its maximum. These shrunken LFCs and their standard errors are used in the Wald tests for differential expression significance testing.

The Wald test allows testing of individual coefficients, or contrasts of coefficients, without the need to fit a reduced model as with the likelihood ratio test. The Wald test *P* values from the subset of genes that pass an independent filtering step are adjusted for multiple testing using the procedure of Benjamini and Hochberg [125] as default. However, in our research we used Bonniforoni correction[126][127], because we have big number of significant differential expressed genes, as I describe in results.

3.4.2 RESULTS:

3.4.2.1 DESeqDataSet Object Design

After we got the count matrix (its rows correspond to genes and the columns to samples) using *featureCounts* function from *Rsubread* package as I described previously in the section “**Error! Reference source not found.**”, we create DESeq-DataSet object, using *D*

ESeqDataSetFromMatrix function, by providing the count matrix, and sample information table which is the experiment design.

For osteoblast, we designed the experiment with one conditions level (time points). The row names are the columns name of count matrix (the samples), the columns are the conditions, here we have one condition level with four variables (Day_Zero, Day_Three, Day_Six, Day_Twelve). The sample information table looks like the following

Samples	Condition
Day_0_R1	Day_Zero
Day_0_R2	Day_Zero
Day_0_R3	Day_Zero
Day_3_R1	Day_Three
Day_3_R2	Day_Three
Day_3_R3	Day_Three
Day_6_R1	Day_Six
Day_6_R2	Day_Six
Day_6_R3	Day_Six
Day_12_R1	Day_Twelve
Day_12_R2	Day_Twelve
Day_12_R3	Day_Twelve

Table 3.1: Table of osteoblast experimental samples

DESeqDataSetFromMatrix syntax is as the following:

```
dds <- DESeqDataSetFromMatrix(countData = “count matrix”,
colData = “sample info table”,
design = ~ condition)
```

3.4.2.2 Performing Differential expression analysis:

As described in methods we need first to estimate the size factors normalization, then estimate the count dispersion, and final step performing the negative binomial GLM fitting and Wald test statistics.

```
dds <- estimateSizeFactors(dds)
dds <- estimateDispersions(dds)
dds <- nbinomWaldTest(dds)
```

To test the gene expression differences between the group conditions we have 6 comparisons.

- Day_12 vs Day_6.
- Day_12 vs Day_3.
- Day_12 vs Day_0.
- Day_6 vs Day_3
- Day_6 vs Day_0

- Day_3 vs Day_0

To figure out the number of comparisons between experiment conditions, we got to the conclusion that for n conditions, the number of comparisons is given by taking the Combination of number of conditions n and $n-2$:

$$C^{n}_{n-2} = \binom{n}{n-2} = \frac{n!}{2!\cdot\ (n-2)!}$$

The function "*results*" from DESeq Bioconductor package gives the results of these comparisons including log2 fold changes of expression values, p-values and adjusted p-values which calculated based on the used statistical test. To determine the comparison between two conditions we define the *contrast* argument, that is an experimental design containing a factor with three levels, the design attribute and the both conditions we need to find log2 fold change of gene expression between them.

We use Bonferroni correction for the p-value of Wald test, where we use the estimated standard error of a log2 fold change to test if it is equal to zero. For the FDR cutoff, we set the argument alpha in *results* function to initial value of 0.01. This independent filtering based on the mean of normalized counts for each gene, optimizing the number of genes which will have an adjusted p value below a given FDR cutoff.

Furthermore, we activate count outliers detecting mood argument *cooks-Cutoff* in *results* function that allows us to determine extreme counts in individual samples that are apparently unrelated to the experimental or study design, and which are considered outliers. A diagnostic test for outliers called Cook's distance, which is a measure of how much a single sample is influencing the fitted coefficients for a gene, and a large value of Cook's distance is intended to indicate an outlier count. We wrote the *results* function syntax as the following to get the desirable results table (example Day_12 vs Day_0)

```
resTwelve_Zero <- results(ddseq,
                          contrast=c("condition", "DayTwelve", "DayZero"),
                          pAdjustMethod = "bonferroni",
alpha = 0.01,
                          cooksCutoff = TRUE,
independentFiltering = TRUE)
```

The common used cutoff of the adjusted p_value is (0.01), and fold change of 4, so the log2 fold change (LFC) is 2. Applying these cutoffs, we got the following number of significant differentially expressed genes in 6 comparisons:

Comparisons	Significance p_value < 0.01	Differentially expressed FC >4, *|LFC| > log2(4)*	Upregulated LFC >2	Downregulated LFC < -2
Day_12 vs Day_0	10058	2299	1231	1068
Day_6 vs Day_0	10026	2834	1702	1132
Day_3 vs Day_0	9695	2378	1466	912
Day_12 vs Day_3	5551	655	233	422
Day_12 vs Day_6	5427	626	214	412
Day_6 vs Day_3	1788	155	92	63

Table 3.2: Number of differential expressed genes with adj. p_value <0.01 and |LFC| > 2.

Using default threshold of 0.01 for adjusted p-value by Bonferroni correction, we got 12932 genes significantly differentially expressed at least in one of the comparisons out of 29148 genes analyzed. We got (4012) genes as a union of genes that significantly expressed at least in one of the comparisons, with adjusted *p*-value less than 0.01 and absolute logarithm fold change greater than 2. From the BioMart data mining tool in Ensembl, we could define the DE protein coding genes, and the noncoding ones, writing a code in R to define the bio type of each gene in DGE data set. In the following table (Table 3.3) is the statistics of protein coding genes, non-coding and lnc_RNA:

	All genes	Protein coding	Non-coding	Linc-RNA
All genes in DESeq	29148	17948	11200	2008
Significant Adj.P <0.01	12932	11773	1159	287
DE genes union Adj.P <0.01 & LFC >2	4012	3308	704	184

Table 3.3: Union of DE genes; upregulated and downregulated, protein coding and noncoding.

Although we used stricter p-value cutoff commonly used by researchers (alpha = 0.001), we still got huge number of significantly differentially expressed genes as it is illustrated in Table 3.4 and MA-plot Figure 3.11.

The MA-plot is originally used to visualize DNA microarray gene expression data, however, it is also used to visualize high-throughput sequencing analysis. It plots the distribution of differences between normalized counts taken in two samples. M refers to log ratio (log2 Fold change) and A refers to mean average scales (mean of normalized counts).

Comparisons	Significance p_value < 0.001	Upregulated LFC >2	Downregulated LFC < -2
Day_12 vs Day_0	9609	1170	1130
Day_6 vs Day_0	9555	1628	1077
Day_3 vs Day_0	9201	1399	869
Day_12 vs Day_3	5069	219	393
Day_12 vs Day_6	4922	197	387
Day_6 vs Day_3	1506	90	59

Table 3.4: Number of DGE with adj. p_value <0.001 and |LFC| > 2

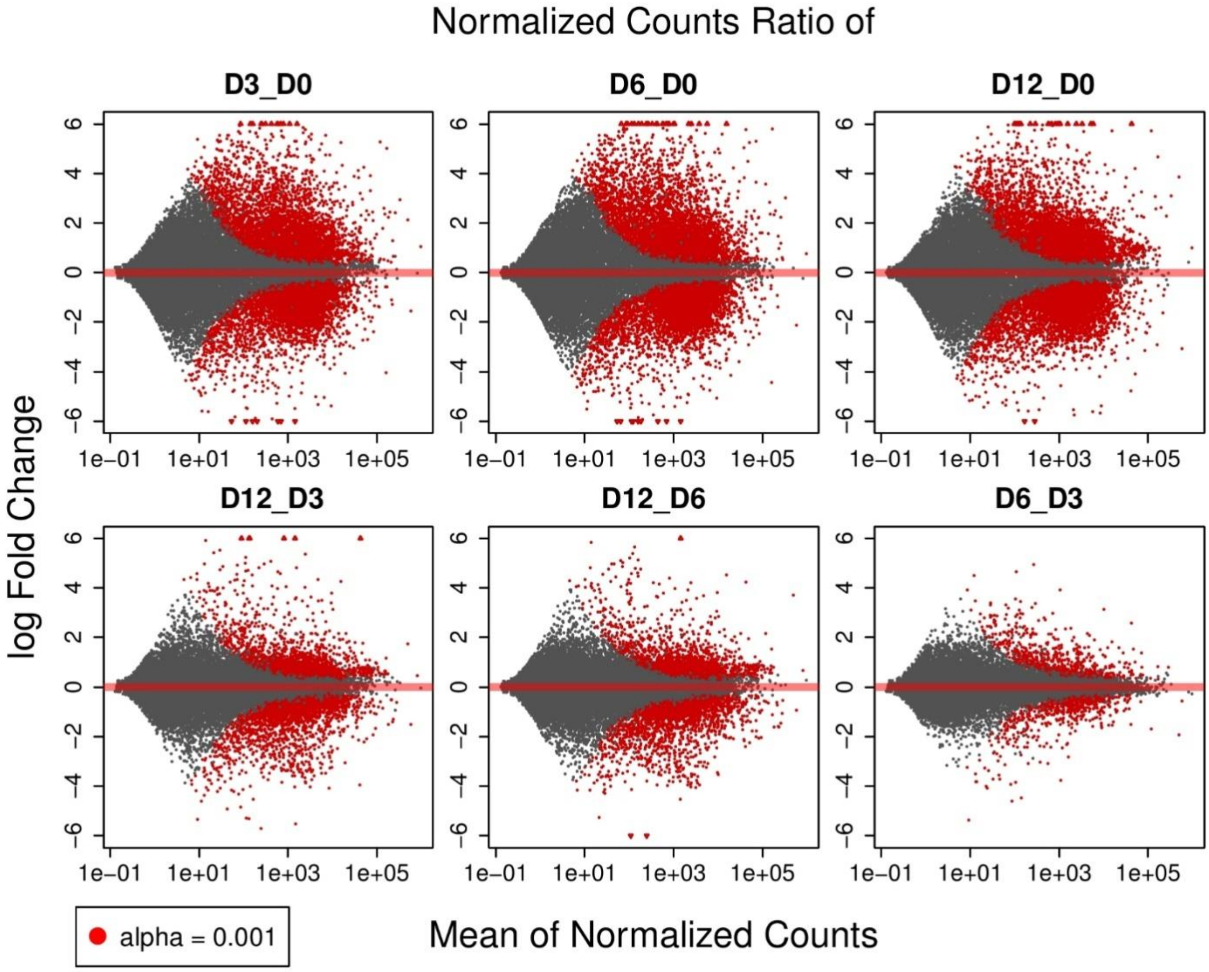

Figure 3.11 MA-plot of 6 comparisons between osteoblast differentiation time points. Red points indicate the genes if the adjusted p-value less than 0.001

In DESeq2, MA-plot of log2 fold change shows how the shrinkage of fold changes works for genes with low counts. Whereas in High Throughput Sequencing (HTS) data, there is huge variance of LFC estimates for genes with low read count. This means, weakly expressed genes seem to show much stronger differences between the compared conditions than strongly expressed genes. This heteroskedasticity (variance of LFCs depending on mean count) complicates downstream analysis and data interpretation, DESeq2 overcomes this issue by shrinking LFC estimates toward zero [123], as it is described previously in the methods. In the Figure 3.12 MA-plots show the difference of LFC shrinkage, in osteoblast comparison day_12 vs day_0, with cutoffs; adj.p-value < 0.01 and |LFC|>2.

3.4.2.3 Setting Thresholds:

Using the common used cutoffs of adjusted p-value and LFC, gives us huge number of genes not suitable for further Gene Ontology enrichment analysis or clusters visualization. Therefore, to choose the appropriate thresholds for our data, we use volcano plot. Usually we cut when the volcano arms start to open as the following Figure 3.13 illustrates.

In the panel (A) the cutoff of the adjusted *p*-value is 0.01, we can see most of the volcano dots are blue, which are the significant DE genes that have adjusted p-value less than 0.01, the small black line in the base of the volcano is the non-significant. In (B) we added the cutoff for the fold change of 4: $abs(log_2FC)> log_2(4)$. We chose the cutoff adjusted *p*-value 10^{-50} according to the volcano plot, as it is illustrated in (C). In (D) applying both cutoffs on the genes set of $|LFC| > \log_2 5$ and $-\log_{10} adjP < 50$. The following graph in Figure 3.14 of volcano plots of the six comparisons, proves that choosing the cutoffs was adequate decision for all the samples

By using cutoff of adjusted p-value less than 10^$^{-50}$ and absolute value of logarithm fold change |LFC| greater than $\log_2 5$, we got 1441 genes including 1386 protein coding, 55 non-coding and 19 lincRNA (Table 3.5).

3.4.2.4 Count data transformation:

For samples similarity analysis and visualization as clustering, it is important to use homoscedastic data (all random variables in the sequence have the same finite variance). Heteroscedasticity in RNA-Seq data causes a problem, when the original count scale is used in clustering or ordination algorithm, the result will be dominated by highly expressed, highly variable genes; if logarithm-transformed data are used, undue weight will be given to weakly expressed genes, which show exaggerated LFCs.

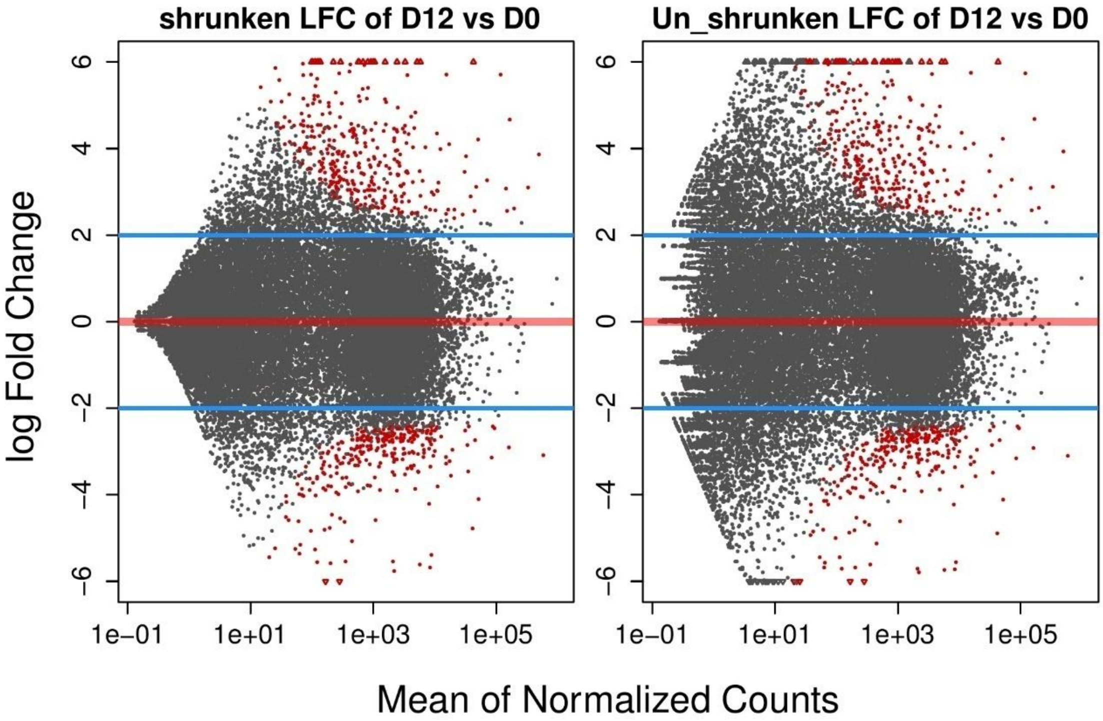

Figure 3.12 LFC shrinkage in day_12 vs day_0, adj.p-value < 0.01 and |LFC| >2

adjP< 10^$^{-50}$ |LFC|> $\log_2(5)$	All genes	Protein coding	Non-coding	lincRNA
DE genes union	1441	1386	55	19

Table 3.5 Significant differential expressed genes subsets with cutoff adjP< 10^$^{-50}$

The purpose behind data transformation is to remove the dependence of the variance on the mean, particularly the high variance of the logarithm of count data when the mean is low. It produces transformed data on the $\log_2$ scale which has been normalized with respect to library size. In order to transform the data to remove the experiment-wide trend. The aim of this transformation is not that all the genes have exactly the same variance after transformation. However, after the transformations, the genes with the same mean do not have exactly the same standard deviations, but that the experiment-wide trend has flattened (Figure 3.15). It is those genes with row variance above the trend which will allow us to cluster samples into interesting groups.

One method for counts transformation is a *regularized logarithm transformation (rlog)*, that is used by DESeq2. It behaves similarly to a log2 transformation for gene with high counts, while shrinking together the values for different samples for genes with low counts, it incorporates a prior on the sample differences. This method considers the variance of each gene, computed across samples, these variances are stabilized – i.e., approximately the same, or homoscedastic – after the rlog transformation. It thus facilitates multivariate visualization and ordinations such as clustering or principal component analysis that tend to work best when the variables have similar dynamic range.

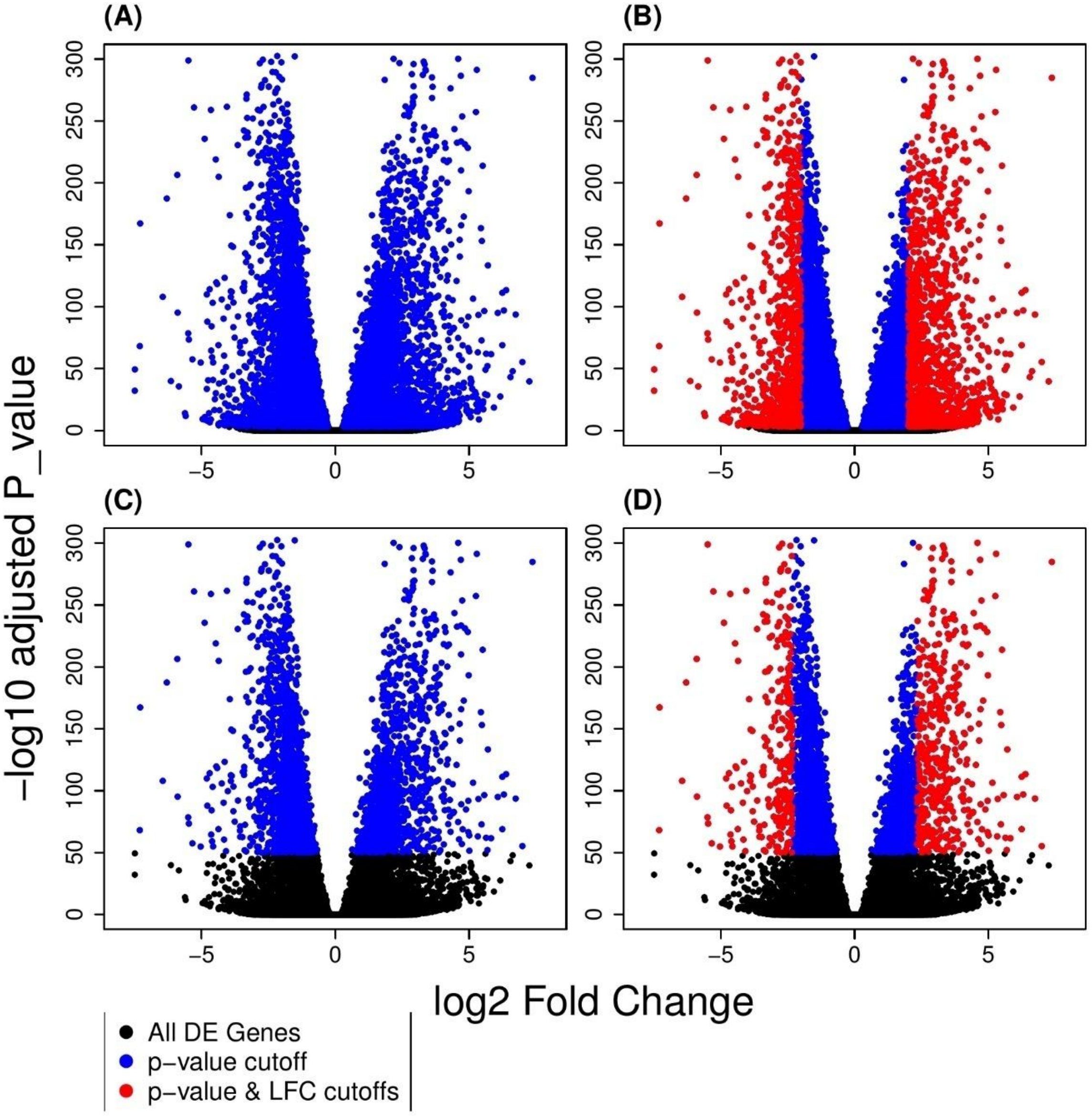

Figure 3.13 Example of volcano plots of DGE in day3 vs day 0, illustrates making the decision of choosing adjusted p-value cutoff = $10^{\wedge-50}$.

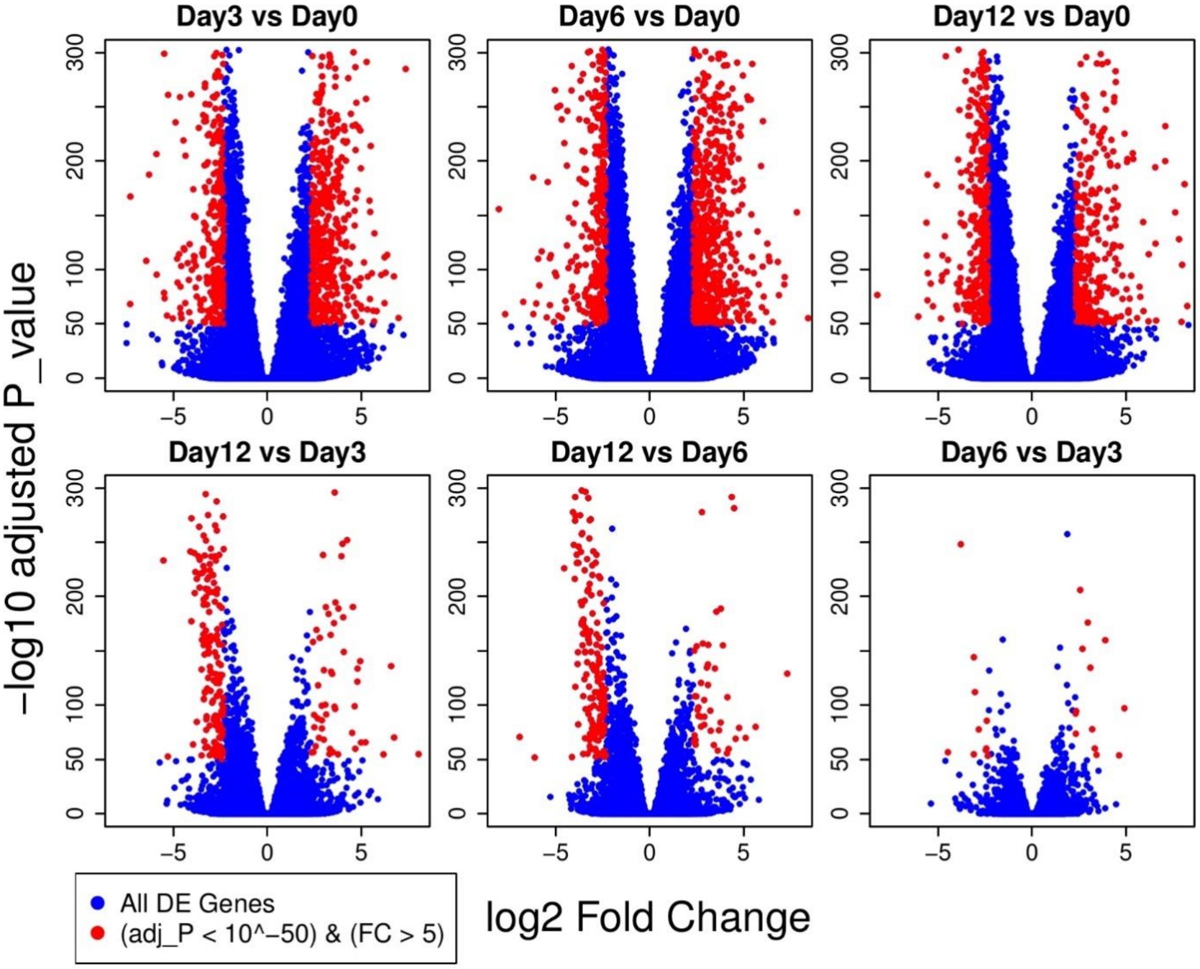

Figure 3.14 Volcano plots of the differential gene expression in osteoblast differentiation days.

The regularized log transformation, transforms the original count data to the log2 scale by fitting a model with a term for each sample and a prior distribution on the coefficients which is estimated from the data. This is the same kind of shrinkage of log fold changes used by the negative binomial Wald Test. The resulting transformed data is given as the following equation:

$$\log_2(q_{ij}) = \beta_{i0} + \beta_{ij}$$

where $q_{ij}$is a parameter proportional to the expected true concentration of fragmentsfor gene i and sample j (see method section), $\beta_{i0}$is an intercept which does notundergo shrinkage, and $\beta_{ij}$is the sample-specific effect which is shrunk toward zerobased on the dispersion-mean trend over the entire dataset. The trend typically captures high dispersions for low counts, and therefore these genes exhibit higher shrinkage from the rlog. Without priors, the design matrix would lead to anon-unique solution, however the addition of a prior on non-intercept betas allows for a unique solution to be found.

The variance stabilizing transformation is another method for data transformation, it calculates the global dispersion trend on a subset of the genes, However, in datasets with large variation in sequencing depth, it was observed undesirable artifacts in its performance, see DESeq2 article [123].

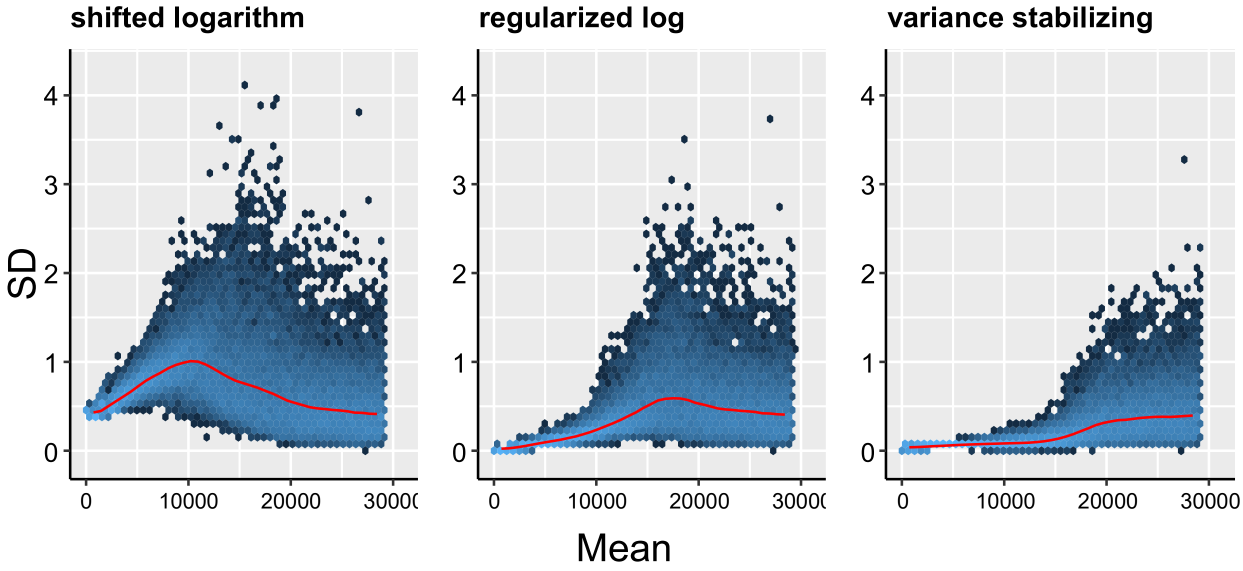

Figure 3.15 Comparisons of data transformation methods

3.4.2.5 Principal Component Analysis:

Principal component analysis (PCA) is a statistical procedure that can be used for exploratory data analysis. PCA uses linear combinations of the original data (gene expression values) to define a new set of unrelated variables (principal components). These new variables are orthogonal to each other, avoiding redundant information. PCA was invented in 1901 by Karl Pearson[128].

Thus, PCA can be used to reduce the dimensions of a data set, allowing the description of data sets and their variance with a reduced number of variables. It is often sufficient to look at the first two components, as these describe the largest variability.

PCA plot is useful for visualizing the overall effect of experimental covariates and batch effects (technical sources of variation), it is used to get an impression on the similarity of RNA-sequencing samples. The variance in RNA-Seq data usually grows with the expression mean, using PCA on the transformed data matrix by regularized logarithm transformation will often lead to principal components that are dominated by the variance of a few highly expressed genes, and avoid the high random noise of low count data

The following graph in Figure 3.16 is the PCA plot of osteoblast data set that has 12 samples for 4 biological conditions. The replicates in each condition show similarity in the variances which proves that the experimental samples did not suffer from an abnormality in variance between biological replicated.

3.4.2.6 Sample to sample distance:

We computed the Euclidean distance between the samples, using the regularized logarithm transformed data count. From the distance matrix, we created dendrogram of the samples as Figure 3.17 illustrates, the count data in day_0 of osteoblast differentiation has greater variance to the other days, while day_3 and day_6 are closer, this means the genes in both time points have similar expression patterns.

A heatmap of the distance matrix gives an overview over similarities and dissimilarities between samples, so we used the Euclidean distance of rlog transformed data likewise to create a distance heatmap as the Figure 3.18.

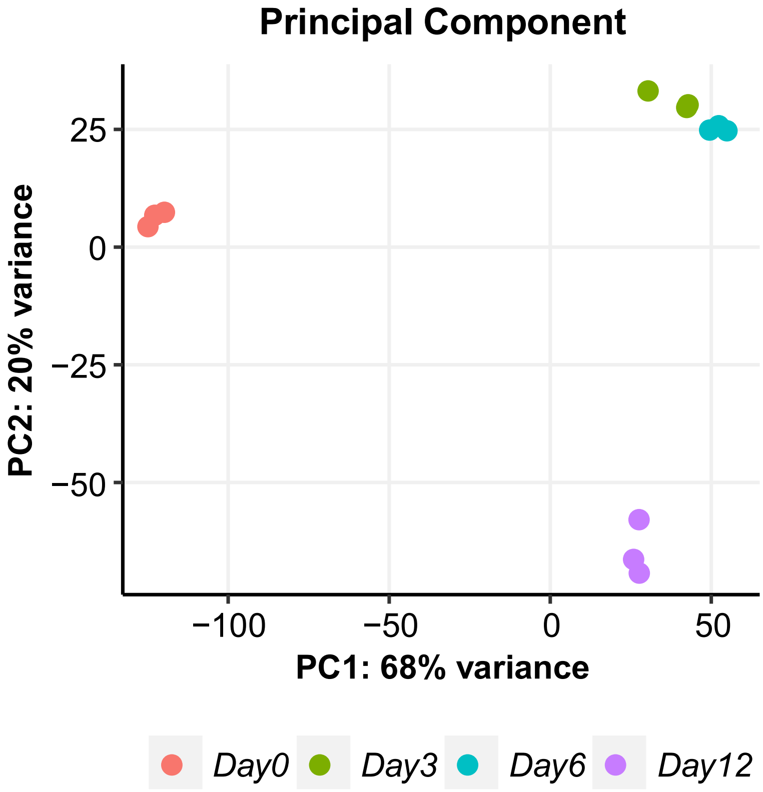

Figure 3.16 PCA of osteoblast dataset. The replicates in each condition show similarity in the variances

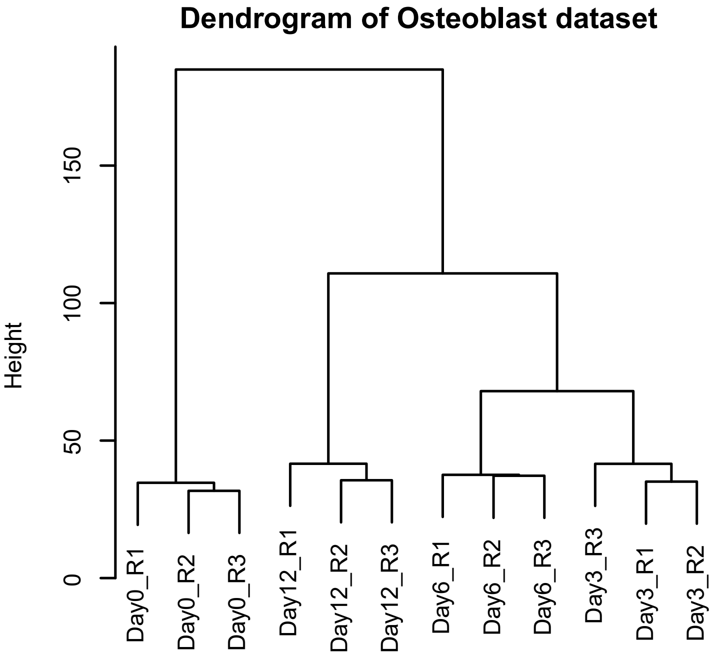

Figure 3.17 Hierarchical clustering dendrogram of osteoblast dataset

3.4.2.7 On / off genes subset analysis

The overall distribution of the fold change differences between the conditions was almost symmetric (Figure 3.19).

However, we found groups of genes with on/off expression as in Table 3.6. We research each gene of them and defined a group of genes which have significant biological role in osteoblast differentiation and ossification. In addition, we got the union of those genes (237 genes) and define their expression patterns in 4 time points (see the appendix I)

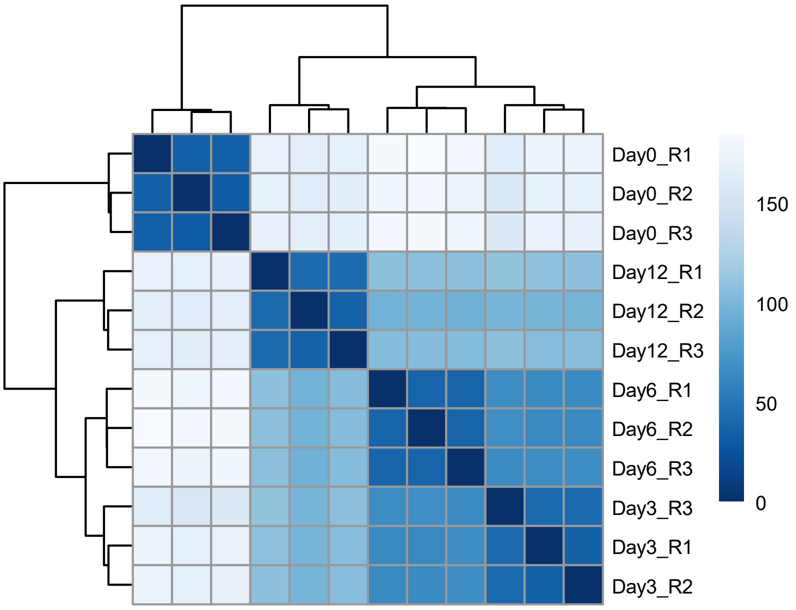

Figure 3.18 Heatmap of samples distance

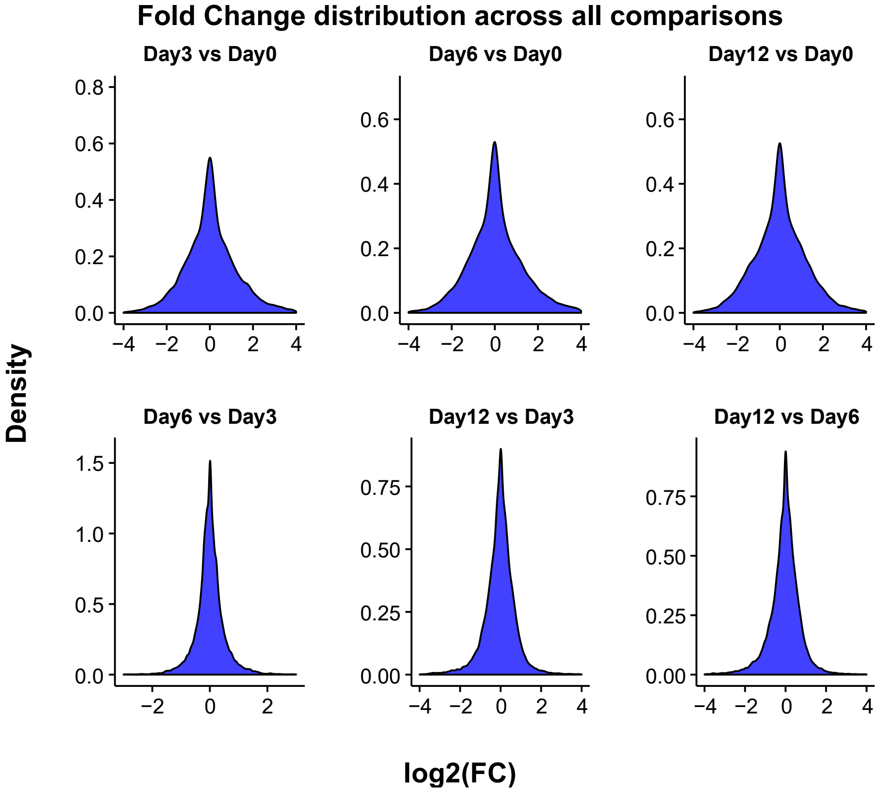

Figure 3.19 The distribution of overall fold change differences across all the comparisons

Comparison **X** vs **Y**	On gene expression on in **X**& off in **Y**	Off gene expression off in **X**& on in **Y**
Day 12 vs Day 0	81	19
Day 12 vs Day 3	12	11
Day 12 vs Day 6	15	5
Day 6 vs Day 0	97	33
Day 6 vs Day 3	1	3
Day 3 vs Day 0	90	27

Table 3.6 ON/OFF genes in osteoblast dataset

We define if a gene is silent during an osteoblast differentiation time point, by applying the following algorithm on the comparisons respectively. For example, the comparison of DGE between Day_3 of osteoblast differentiation against Day_0:

First, we take the normalized counts of significant differentially expressed genes in Day3 vs Day_0 with adjusted p-value less than 0.01 and FC greater than 4.

```
pd = 0.01
lfc = log2(4)
res3_0 <- as.data.frame(resThree_Zero)
# the significant DGE in Day_3 vs Day_0, adjp < 0.01 , |LFC|>2
Diff3_0 <- res3_0 [which(res3_0$padj<pd& abs(res3_0$log2FoldChange)>lfc),]
# all the normalized count in DESeq object
NormSF_Counts <- counts(ddseq, normalized=TRUE)
#normalized count of significant DGE in Day_3 vs Day_0
NC3_0 <- as.data.frame( NormSF_Counts[which(rownames(NormSF_Counts)
%in% rownames(Diff3_0)), ])
```

Then we calculate the mean of the normalized count of the replicates in each condition.

```
NC3_0$Mean0 <-rowMeans(cbind(NC3_0$Day0_R1,NC3_0$Day0_R2,NC3_0$Day0_R3))

NC3_0$Mean3 <-rowMeans(cbind(NC3_0$Day3_R1,NC3_0$Day3_R2,NC3_0$Day3_R3))

NC3_0$Mean6 <-rowMeans(cbind(NC3_0$Day6_R1,NC3_0$Day6_R2,NC3_0$Day6_R3))

NC3_0$Mean12 <-rowMeans(cbind(NC3_0$Day12_R1,NC3_0$Day12_R2,NC3_0$Day12_R3))
```

After we got the mean of the normalized count, we check whether DE genes are off in Day_0, if at least in two replicates the normalized count is 0, and the mean of normalized count in 3 replicates is less than 1.5. We apply the same conditional statements on Day_3 to get the silent genes.

```
On3off0 <-  NC3_0[which(
             (NC3_0$Day0_R1 ==0 & NC3_0$Day0_R2 == 0 & NC3_0$Mean0 < 1.5) |

             (NC3_0$Day0_R1 ==0 & NC3_0$Day0_R3 == 0 & NC3_0$Mean0 < 1.5) |

            (NC3_0$Day0_R2 ==0 & NC3_0$Day0_R3 == 0 & NC3_0$Mean0 < 1.5) ), ]

On0off3 <-  NC3_0[which(
             (NC3_0$Day3_R1 ==0 & NC3_0$Day3_R2 == 0 & NC3_0$Mean3 < 1.5) |

             (NC3_0$Day3_R1 ==0 & NC3_0$Day3_R3 == 0 & NC3_0$Mean3 < 1.5) |

        (NC3_0$Day3_R2 ==0 & NC3_0$Day3_R3 == 0 & NC3_0$Mean3 < 1.5) ), ]
```

Why did we repeat this procedure for every comparison separately? Because we need to avoid annotating a gene as turned off during a differentiation time point, while it is not significant differentially expressed between those conditions, therefore we take merely the differentially expressed genes that achieve the threshold criteria in each individual comparison.

3.5. GENE ONTOLOGY ANALYSIS ENRICHMENT

3.5.1 METHOD

For clustering the differential expressed genes, we plot a heatmap that creates a similarity matrix of the values and groups the genes with similar pattern together, then highlights them in different colors. We got the normalized count of the DE genes we desire to cluster, from DESeqDataSet object(dds) by applying the following statement:

```
DSC <- as.data.frame(counts(dds, normalized = TRUE)[genesCluster,])
```

Then we calculate the count mean of the replicates in each condition, we create new mean count matrix with all the genes names.

```
DSCM <- data.frame(row.names = rownames(DSC))

DSCM$Day0_Mean <- rowMeans(cbind(DSC$DayZero_R1, DSC$DayZero_R2,
                            DSC$DayZero_R3), na.rm=TRUE)
DSCM$Day3_Mean <- rowMeans(cbind(DSC$DayThree_R1 , DSC$DayThree_R2,
                           DSC$DayThree_R3) , na.rm = TRUE)

DSCM$Day6_Mean <- rowMeans(cbind(DSC$DaySix_R1 , DSC$DaySix_R2,
                           DSC$DaySix_R3) , na.rm = TRUE)

DSCM$Day12_Mean <- rowMeans(cbind(DSC$DayTwelve_R1, DSC$DayTwelve_R2,
                            DSC$DayTwelve_R3) , na.rm = TRUE)
```

The fundamental step to generate clusters heatmap, is to scale the normalized count with mean centering as the following:

$$x_{new} = \frac{x - \mu}{\sigma}$$

Where x is the value of normalized counts, μ is the mean of the rows, σ is the standard deviation, however the mean of the new row values is 0 and the standard deviation is 1. We need to transpose the matrix to cluster the genes as the following:

```
DSCM_scale <- apply(DSCM, 1, scale)

myclusters <-t(DSCM_scale)

colnames(myclusters) <- colnames(DSCM)
```

The heatmap function in R calculate the Euclidean distances and calculate the variance to cluster the genes and reorder the values according to its dendrogram. We could identify 9 clusters of the genes by visual inspection, whereas the genes were plotted in the heatmap according to their expression patterns see Figure 3.20.

Then we uploaded the differentially expressed genes clusters to Ontologizer [129], using the model-based gene set analysis method “MGSA”[130] we got the top annotation-enriched GO terms of the differential gene expression group. Then we applied parent-child intersection approach [131]on the clusters, to get the enriched GO terms of each cluster. For correcting the *p*-value we used Benjamini Hochberg correction.

3.5.2 RESULTS

For biological evaluation of the function of the protein coding genes, we performed gene ontology enrichment analysis. There are 1386 protein coding genes, differentially expressed with more stringent cutoff values (absolute fold change of fivefold or greater and adjusted *P*-value < 10^{-50}). These genes were plotted within a heatmap to identify clusters of genes according to their expression pattern. In total, nine clusters were identified by visual inspection, as illustrated in protein coding genes clusters heatmap (Figure 3.20). Cluster A comprises a total of 303 genes that display high expression at day 0 and low expression levels at later time points. 301 of them are annotated to a GO term. 38/301 genes have “*nucleic acid binding transcription factor activity*”. Several genes are involved in “response to ER stress” including *Atf3, Atf4* and *Ddit3.* This cluster also contains *Bmpr1b* which is well known to be involved in osteoblast differentiation.

Genes in Cluster B display a very similar expression pattern in comparison to cluster A. In this cluster we also identified a large number of genes involved in “*transcription from RNA polymerase II promoter*” and “*transcription factor activity, protein binding*” including *Actn4* which is involved in regulation of transcription as well as substrate adhesion-dependent cell spreading. Almost 10 percent of these genes are coding for proteins localized in the extracellular matrix, and nine of them are structural constituents of collagen trimers. About 18 percent of these genes are involved in “*movement of cell or subcellular component*”.

Genes that show increasing expression levels of time reaching highest expression at day 12 are localized in cluster C. 23/103 genes within this cluster are annotated to terms such as “*ossification*”, “*biomineral tissue development*”, and “*skeletal system development*” including well known genes such as *Bglap*, *Bglap2*, *Dmp1*, *Ibsp*, *Mef2c*, *Mepe*, *Ostn*, *Phex*, *Spp1, Vdr*. Of note, *Grem1* is also located within this cluster which is a known BMP antagonist, and its increasing expression is associated with decrease in proliferation and migration of cells. Another 24/103 of the genes in Cluster C, are annotated to “*immune system process*”.

Cluster D contains genes that have very low expression levels at day 0 but higher expression levels at the later time points. Although GO analysis did not reveal statistical significance, more than 10 percent of these genes are annotated to “*oxidoreductase activity*”. Significance was obtained for a subset of these genes annotated to “*respiratory chain*” and “*oxidoreductase complex*”. Interestingly, *Tnfsf11* (aka RANKL) is located within this cluster as well as *Mmp13* and *Arrb1.* The latter has been described to protect against ER stress and its expression is kind of inverse to the genes in cluster A where several genes are annotated to response to ER stress.

There are 22 genes in cluster E which show highest expression levels at day 3. They are all annotated to the rather unspecific GO term "*single-organism process*". However, this cluster contains *Col9a3* and *Ucma*. Both genes are involved in chondrogenic differentiation, and *Ucma* has been speculated to be involved in negative regulation of osteoblast differentiation.

Cluster F contains five genes. These genes display moderate expression levels from day 0 on with peak expression at day 6 and very low expression levels at day 12. Gene products of the four genes *Fbn2*, *Hmcn1*, *Mal2*, and *Serpini1* are localized within the extracellular matrix. The fifth gene, *Lrrc75b* has been reported to be a negative regulator of myogenic differentiation [132].

Genes in cluster G display low expression levels at day 0 and peak expression at day 3. About one third of these genes are involved in lipid metabolic processes and 19/101 genes are annotated to have oxidoreductase activity. Several of these genes such as *Cyp51*, *Dhcr7*, *Dhcr24*, *Fdps*, *Hmgcs1*, *Idi1*, *Msmo1*, *Mvd*, *Nsdhl*, *Pmvk*, *Sc5d*, *Sqle,* and *Tm7sf2* encode enzymes of the cholesterol biosynthetic pathway, and several of the genes in this cluster play a known role in bone development as shown by knockout phenotypes. For example, developing *Cyp51* knockout embryos exhibit shortened and bowed limbs and synostosis of femur and tibia, *Cyp7b1* knockout animals have decreased bone mineral density and decreased trabecular number and thickness, mutations in *DHCR24* cause severe developmental anomalies including short limbs, and *Elovl6* plays a role during growth plate development.

In cluster H, we observed an incremental increase of expression levels with a peak at day 6 and moderate expression at day 12. Products of 22/116 genes are localized within the extracellular matrix and several of the genes in this cluster are annotated to "*biomineral tissue development*", "*collagen binding*", and "*Collagen trimer*". *Cst3* encoding Cystatin C has been suggested to promote osteoblast differentiation via BMP signaling. Although not annotated to "*collagen binding*", epiphycan encoded by *Epyc* is a regulator fibrillo-genesis by interacting with collagen fibrils and other ECM proteins. This cluster also contains *Srfp2,* a negative regulator of Wnt-signaling, *Meox2* which is involved in limb morphogenesis and the genes osteoglycin (*Ogn*) and osteomodulin (*Omd*). Osteoglycin-deficient mice have collagen fibril abnormalities, and raising osteomodulin expression levels have been reported to mediate the switch from osteoblast proliferation to differentiation.

Cluster I is the largest cluster containing 373 protein-coding genes annotated to a GO term. Expression levels are comparably high at day 3 and day 6, but low at day 0 and day 12. About one third of these genes (125/373) are annotated to "*cell cycle*" including cyclins, cell division cycle (associated) genes, cyclin dependent kinases and -kinase inhibitors, centromer proteins, E2F transcription factors, and numerous genes encoding kinesin-like proteins. Genes in this cluster with known roles in bone biology include *Birc5* which is a *Comp* target gene, *Clu* which is a marker of zonal articular chondrocytes, *Dkk2, Fam111a* where mutations cause Gracile Bone Dysplasia (OMIM 602361), *Gdf10, Gdf11, Id2, Itm2a, Kazald1* aka *Bono1, Lgas9, Mmp9, Nell1, Postn, Ptn, Sfrp1,* and *S100a11* which accelerates chondrocyte hypertrophy.

Our analysis approved the genes that are known to be associated with osteoblast differentiation and play role in bone forming and mineralization, those genes have been studied inclusively, as I described in the gene ontology analysis. While we found other new candidates known with their function in cell proliferation and differentiation, in addition to ions and minerals binding, we could build our theory about their roles in osteoblast differentiation and ossification based on their expression pattern (our paper). Our On/Off method highlight few group of gene that are silent or turned off during an osteoblast differentiation time point while it is on in others (See appendix II).

To confirm the biological validity of our data, we analyzed differential expression of selected marker genes that are well known to be differentially expressed during osteoblast differentiation.

As shown in Figure 3.21, Bmp2 which is a marker gene for osteogenic induction displayed highest expression levels at day 0. Runx2 which encodes one of the key transcription factors of bone formation is highly expressed with slight downregulation over time. Col1a1 shows peak expression at day 6. One of the marker genes for osteoblast matrix remodeling Mmp13 is highly expressed at day 3 and day 12 with an intermittent drop at day 6. Marker genes for osteoblast differentiation such as Bglap2 showed rising expression levels over time starting at day 3, and Sost which is a marker for the onset of mineralization is basically turned off until day 6, but expressed at day 12. In contrast, Pparg, a marker gene for adipogenic differentiation displayed decreasing expression and Lep which is expressed by differentiated fat cells shows constantly low expression (Figure 3.22). The chondrogenic marker genes Comp and Col9a1 are downregulated over time. Finally, marker genes for myogenic differentiation such as Myod and mature muscle cells such as Ckm displayed marginal expression levels. Taken together, expression of marker genes confirmed osteogenic differentiation.

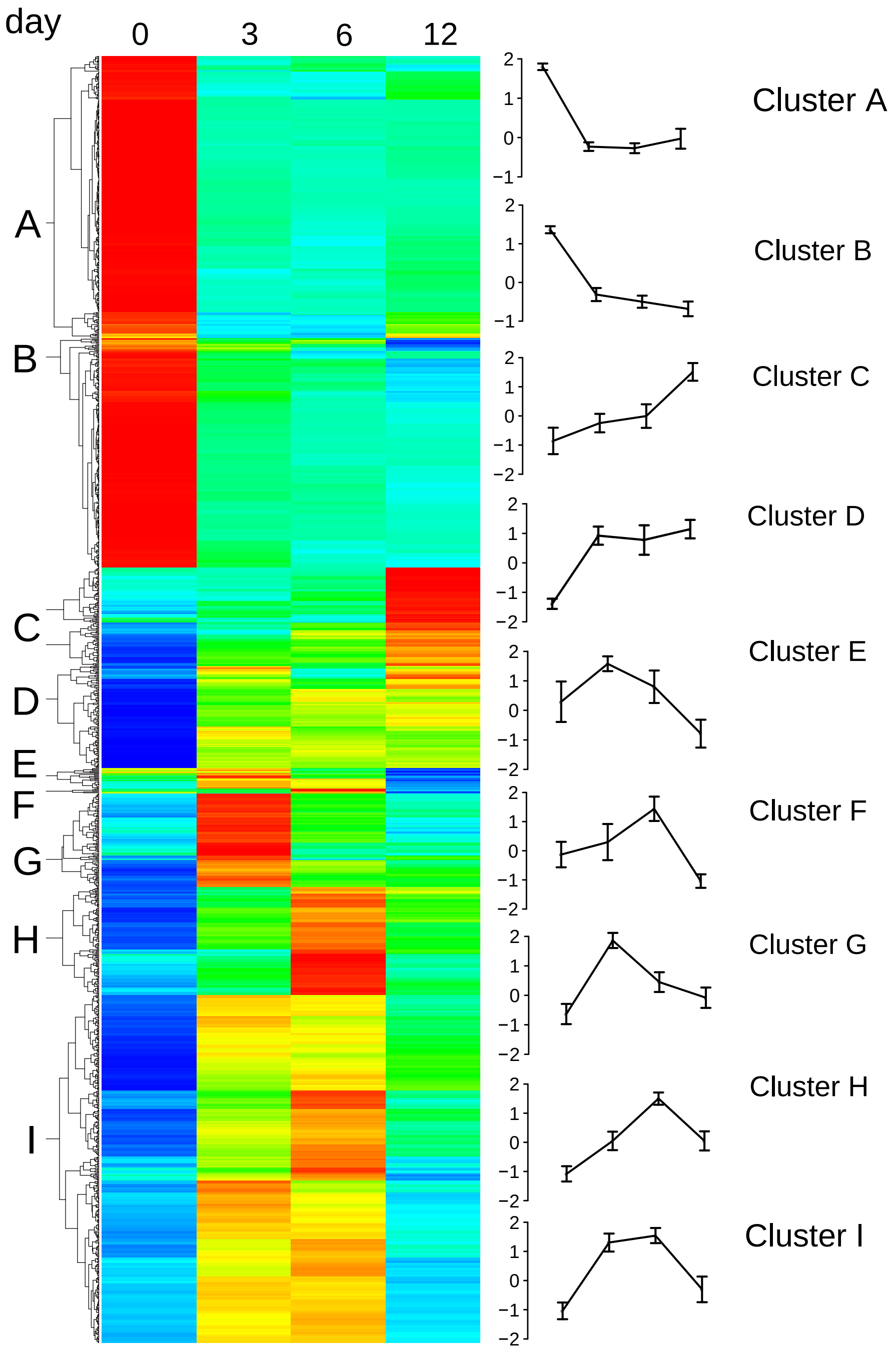

Figure 3.20 Heatmap of genes cluster with their expression patterns and significant GO terms

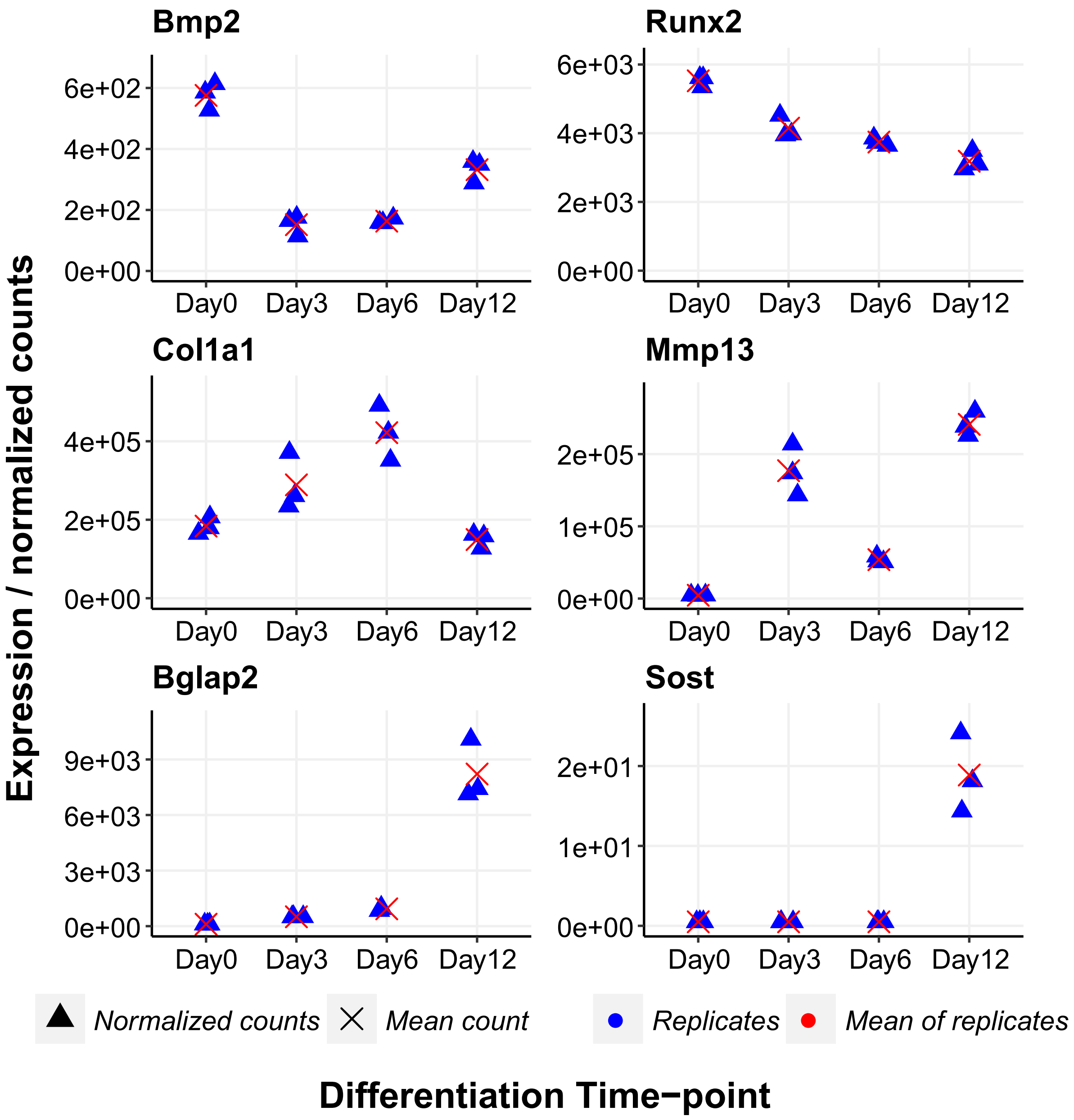

Figure 3.21 Expression pattern of few gene markers well known to be differentially expressed during osteoblast differentiation.

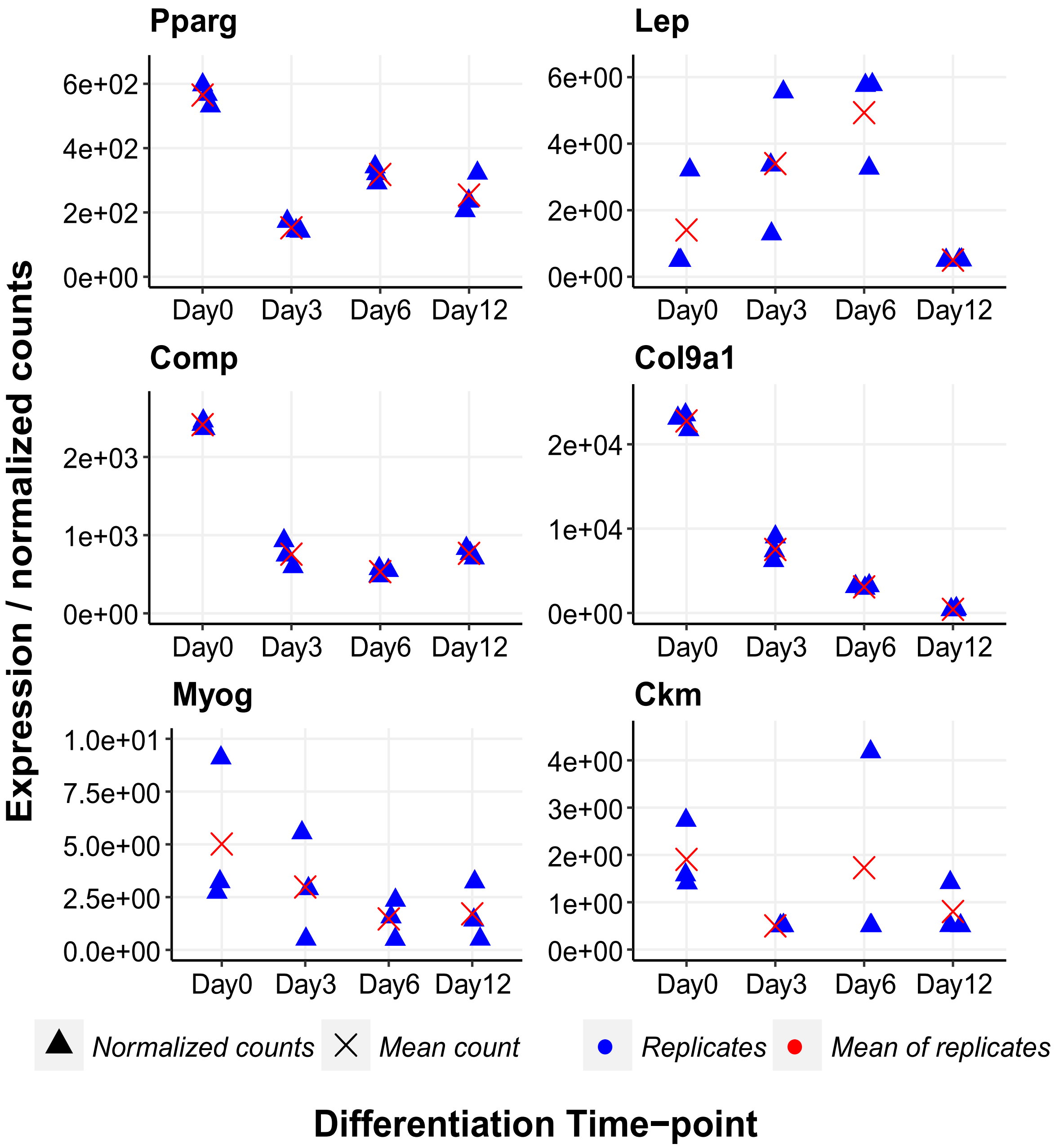

Figure 3.22 Expression of genes known to be downregulated in osteoblast cells

3.6. DIFFERENTIAL EXONS USAGE

Alternative transcription start-sites lead to differences in the beginning of mRNA, whereas alternative splicing causes some of the exons to be skipped and not translated at all. RNA-seq offers exciting possibilities for studying the expression and regulation of isoforms on the whole genome level.

Most of the current RNA-seq methods produce short reads which do not cover full transcripts. Instead, transcripts need to be assembled from sequenced fragments. The assembly and the subsequent abundance estimation can be challenging, because isoforms typically have common or overlapping exons. Furthermore, the coverage along transcripts is not uniform because of biases introduced in sequencing and library preparation. In order to avoid uncertainties in the assembly, one approach for studying alternative isoform regulation of it is to look at differences in the usage of individual exons. RNA-seq reads can also be mapped to exons so that the differences in exon-specific counts can be compared between certain conditions, groups, or treatments.

3.6.1 METHODS

We used a statistical method to test for differential exon usage in RNA-seq data, by applying Bioconductor package DEXSeq [133], which uses generalized linear models, taking into the account the biological variability and looks for differences across conditions of the relative usage of each exon. Using HTSeq library in python, we generated a reference annotation genes model (flattened GFF file) contains one entry for each exon or exonic part, which is cut from the exon if the exon's boundary differs between transcripts. Then we got the count of reads that overlap with each of the exon counting bins defined in the flattened GFF file in each sample. The DEXSeq function normalizes these counts by the library size factor s_j, which accounts for the depth that sample j was sequenced. The number of the reads N_{ijk} overlapping counting bin (exonic part) k of gene i in sample j, follows the negative binomial (NB) distribution and modeled by GLMs, where the dispersion parameter is estimated by, firstly performing an IRLS (iteratively reweighted least square) fits for each gene, then, insert these fitted values in the log likelihood function with Smyth's Cox-Reid [134][135] term and find its maximum using Brent's line search. So, the gene expression variability is absorbed by the model parameters, while the model increase the power of the test for differential exon usage.

3.6.2 RESULTS

We used DEXseq method to test for differential exon usage in comparative RNA-Seq experiments. We mean by differential exon usage (DEU), changes in the relative usage of exons caused by the experimental condition. The relative usage of an exon is defined as:

$$\frac{\text{number of transcripts from the gene that contain this exon}}{\text{number of all transcripts from the gene}}$$

We test the differential exon usage in each comparison, applying merely the DEXSeq method with thresholds; adjusted p-value less than 0.01 and FC greater than 4, we got the following number of entries, which are exons or exonic part differentially used between the conditions, many of those exons belong to more than one transcript, and some of them overlapped with few genes. The results presented in Table 3.7

Comparisons	Adjusted p-value < 0.01	adjp < 0.01 & |logFC| > 2
Day12 vs Day0	17229	291
Day12 vs Day3	3216	71
Day12 vs Day6	2917	35
Day3 vs Day0	17283	293
Day6 vs Day0	18408	373
Day6 vs Day3	480	8

Table 3.7 Differential Exon Usage (DEU) in osteoblast comparisons; adjp <0.01, |logFC|>2

The results from DEXSeq are not adequate for further alternative splicing analysis. Therefore, the genes name and gene type associated with the entries must be provided, using Biomart data mining tool from Ensemble, I built the function code in R (Code-box 3. 14).

It has been found in previous studies that almost 73% of human genes are alternatively spliced [136]. To figure out the featured biological role that a gene can play when alternative transcripts expressed in different conditions, we need to determine if the exonic part that differentially used between those conditions is protein-coding or within the open reading frame ORF. To achieve this task, first I coded a function to provide the bio-type of transcripts that include the differential used exon, as it is illustrated in Code-box 3.15. The output of this function provides us with a data matrix includes the DEU, logFC, adjP-value, Gene name, Gene type, Transcript type, Genomic start and end of the exon part, Genomic width, and genomic strand.

This data matrix is the basis of ExonORF function that define if the exonic part is within ORF. I coded ExonORF Code-box 3.16 function in the following steps:

1- Get the table exons features from BioMart in Ensemble, of all transcripts that differentially used in all comparisons. The exon feature table wouldh look like this:

Exon.Chr.Start	Exon.Chr.End	En.Transcript.ID	G.coding.start	G.coding.end	Strand
15356	15422	ENSMUST00000082423	NA	NA	-1
14145	15288	ENSMUST00000082421	14145	15288	+1

The exons that are non-coding the Genomic.coding.start and Genomic.coding.end not available, while some exons are partially within ORF so they have either Genomic coding start or end.

2- Check the exonic bin is within the range of a defined exon in Ensemble. S_E , E_E are the start and end of an exon, and S_B, E_B are the start and end of an exon part.

3- Check if this exon part is within the ORF. S_c , E_c are the start and end of coding frame.

$$S_B, E_B \in [S_E, E_E] \cap S_B, E_B \in [S_c, E_c]$$

Code-box 3.14 function to add the type and name of genes in DEXSeq data matrix

```
biomart <- read.table(paste(inDirG,"BioMart_GRCm38R86.txt",sep="/"), sep = "\t",
header = TRUE)

library(hash)
gene_name_hash <- hash(keys= biomart$Ensembl.Gene.ID,
values= biomart$Associated.Gene.Name)
biotype_hash <- hash(keys= biomart$Ensembl.Gene.ID,
values= biomart$Gene.type)
DxGeneType <- function(DEU){
    DEUx <- DEU
    for (i in 1:length(DEUx$groupID))
    {
      if (grepl("\\+",DEUx$groupID[i]))
      {
        z<- unlist(strsplit(DEUx$groupID[i], "\\+"))
        x<- vector(mode = "character")
        y<- vector(mode = "character")
        for(k in 1:length(z))
        {
          if (is.null(gene_name_hash[[ z[k] ]])){
            x[k]<- "Not Available"
            y[k] <- "NA"
          }
          else{
            x[k] <- as.character(gene_name_hash[[ z[k] ]])
            y[k] <-as.character(biotype_hash[[ z[k] ]])
          }
        }
        DEUx$gene_name[i] <- paste(x ,sep = " + ", collapse = "+")
        DEUx$gene_t[i] <- paste(y ,sep = " + ", collapse = "+")
      }
      else
      {
        if (is.null(gene_name_hash[[ DEU$groupID[i] ]])){
          DEUx$gene_name[i] <- "Not Available"
    DEUx$gene_t[i] <- "NA"
    }
        else{
          DEUx$gene_name[i] <- as.character(gene_name_hash[[ DEUx$groupID[i] ]])
          DEUx$gene_t[i] <- as.character(biotype_hash[[ DEUx$groupID[i] ]])
        }
     }
    }
}
```

Code-box 3.15 function to define bio-type of transcripts

```
trans_biomart <- read.table(paste(inDirG,"transcriptsBiotype.txt",sep="/"), sep =
"\t", header = TRUE)

transName_hash <- hash(keys= trans_biomart$Ensembl.Transcript.ID, values=
trans_biomart$Associated.Transcript.Name)
transBiotype_hash <- hash(keys= trans_biomart$Ensembl.Transcript.ID, values=
trans_biomart$Transcript.type)

trans_ProCod <- function(DX, TX){
  t <- as.character(DX$transcripts)
for (i in 1:length(t))
  {
if (grepl("c",t[i]))
    {
      c <-gsub( 'c','', (gsub("[[:punct:]]",'', t[i])))
      z<- unlist(strsplit(c, " "))
      x<- vector(mode = "character")
      y<- vector(mode = "character")
for(k in 1:length(z))
      {
        if (is.null(transName_hash[[ z[k] ]])){
          x[k]<- "Not Available"
          y[k] <- "NA"
        }
else{
          x[k] <-as.character(transName_hash[[ z[k] ]])
          y[k] <- as.character(transBiotype_hash[[ z[k] ]])
        }
}
      DX$transName[i] <- paste(x ,sep = "  +  ", collapse = "+")
      DX$transBiotype[i] <-  paste(y ,sep = "  +  ", collapse = "+")
      TX <- append(TX, z)
    }
else{
      TX <- append(TX, t[i])
      if (is.null(transName_hash[[ t[i] ]])){
        DX$transName[i] <- "Not Available"
        DX$transBiotype[i] <- "NA"
      }
else{
        DX$transName[i] <- as.character(transName_hash[[ t[i] ]])
        DX$transBiotype[i] <- as.character(transBiotype_hash[[ t[i] ]])
      }
    }
  }

  DXPro <- DX[ which(grepl("protein_coding", DX$transBiotype))
                     ,c(1,23,24,22,25,26,2,3,7,8,9,10,12,13,14,15)]
  resPro = list(DX = DXPro, Trans= TX)
  return(resPro)
}
```

Code-box 3.16 function to define the coding exon (within ORF)

```
ExonsCor <-  read.table(paste(inDirG,"exonsCor.txt",sep="/"), sep = "\t",
header = TRUE)

Trans <- Reduce(union, list(Trans12_0,Trans12_3,Trans12_6,
Trans3_0,Trans6_0,Trans6_3))

exonCor <- ExonsCor[which(ExonsCor$Ensembl.Transcript.ID %in% Trans),]

CodStart_hash <- hash(keys= exonCor$Exon.Chr.Start..bp. ,
values=exonCor$Genomic.coding.start)
CodEnd_hash <- hash(keys= exonCor$Exon.Chr.End..bp.,
values=exonCor$Genomic.coding.end)

ExonORF <- function(DX, trans){
  BMexon <- ExonsCor[which(ExonsCor$Ensembl.Transcript.ID %in% trans),]

for(j in 1:nrow(DX)){
    Sb <- DX$genomicData.start[j]
    Eb <- DX$genomicData.end[j]
for(i in 1:nrow(BMexon)){
      S <- BMexon$Exon.Chr.Start..bp.[i]
      E <- BMexon$Exon.Chr.End..bp.[i]
      Sc <- BMexon$Genomic.coding.start[i]
      Ec <- BMexon$Genomic.coding.end[i]

if(Sb>=S  && Eb<=E){
        cat("\n start of bin Sb and the end is inside S,E ")
if(!is.na(Sc)  && !is.na(Ec)){
          cat("\n start coding is integer ")
if(Sb>=Sc && Eb<=Ec){
            cat("\n  exonic bin is coding  ")
            DX$ExonType[j] <- "coding_Exon"
          }
else{
            DX$ExonType[j] <- "out_ORF"
          }
        }
else{
          DX$ExonType[j] <- "NOT_Coding"
        }
      }
    }
  }

return(DX)
}
```

Table 3.8 Table 3.8 number of the differential used coding exon (within ORF) in Osteoblast datasetshows the number of differential used exons in each comparison, that are within the coding frame.

Comparisons	Exons within ORF adjp < 0.01 & |logFC| > 2
Day12 vs Day0	32
Day12 vs Day3	13
Day12 vs Day6	7
Day3 vs Day0	35
Day6 vs Day0	50
Day6 vs Day3	0

Table 3.8 number of the differential used coding exon (within ORF) in Osteoblast dataset

3.7. LONG NON-CODING RNA IDENTIFICATION

3.7.1 METHOD

We innovated an algorithm to predict the potential functions of lncRNA genes which are differentially expressed, by their correlation with protein coding genes. The algorithm is based on two concepts (Figure 3.23); the first concept is the spatial interaction of the lncRNA and the protein coding genes or what is called ***topologically associating domain*** (TAD)[137]. And the second concept is the co-expression correlation which was used in previous study for lncRNA functions characterization[138]. Then define the enriched functional terms among the protein coding genes that are significantly correlated with lncRNAs using a gene ontology tool.

Since the lncRNA expressed in lower level, so we used more tolerant cutoffs to get the significant differential expression, whereas the adjusted *p*_value less than 0.01 and the binary logarithm of fold change between the conditions is greater than 1.5 (adj.Pvalue ≤ 0.01 , FC ≥ 1.5). The algorithm workflow is concise in the following steps:

1- Mapping the reads and the default differential gene expression pipeline using DESeq2.

2- Creating a DGE data matrix for protein-coding genes and another data matrix for lncRNA.

3- For computing the correlation, we generate two matrices for lncRNA and protein-coding genes, contain normalized counts (expression values) among the different Osteoblast differentiation time point.

4- Getting the TAD annotation for mouse genome mm10.

5- Selecting the protein coding gene and lncRNA pairs based on two criteria; firstly, the protein-coding gene and the lncRNA must be within the same topological associated domain. Secondly, the absolute correlation value must be greater than 0.9, and the *p*_value of the correlation test less than 0.01.

6- Most of the lncRNA correlated to more than protein-coding genes. lncRNA have positive correlation with protein coding genes, when they have similar expression pattern, and negative correlation when their expression patterns are contradicted.

3.7.2 RESULTS

Applying this algorithm on osteoblast data, we got throughputs needed to be biologically verified, which was not possible due to flaws in wet-lab organization, however, according to the logical methodology we used, based on published literature [137][138], the algorithm is applicable and need RNA-seq data to confirm it.

As statistical overview of results I got from osteoblast data; there were 10760 protein coding genes differentially expressed with thresholds (adj.Pvalue ≤ 0.01 , FC ≥ 1.5) and within the Topological Associated Domains (TAD) , 299 of those protein coding genes are correlated with 126 lncRNA. However, we have 285 differentially expressed lncRNA with the same cutoffs, 158 of them were not correlated (Table 3.9 number of differential expressed and correlated Pro.Cod genes and lncRNA)

	DGE adjp < 0.01 & FC >1.5	Within TADs	Correlated	Not correlated
Protein coding	10775	10760	299	10461
lncRNA	285	284	126	158

Table 3.9 number of differential expressed and correlated Pro.Cod genes and lncRNA

The distinct result in our analysis, that our algorithm can define when the lncRNA is positively correlated with the protein coding gene, this guide us to the hypothesis that the lncRNA enhancing the expression of the protein coding gene, in other words, it plays positive regulation role to that Pro.Cod gene. While the lncRNA negatively correlated with the protein coding gene, it plays suppressor role. (Appendix II)

Some of the lncRNA correlated with more than one Pro.Cod gene, therefore we have chosen the maximum correlated Pro.Cod gene (positively and negatively). However, for the analysis, we needed to study the full set of Pro.Cod genes that are correlated with one lncRNA. Moreover, many of lncRNA correlated positively with some Pro.Cod genes and negatively with others. We found few biological meaningful examples of such lncRNA behavior served our research on osteoblast differentiation (See our submitted paper).

	All pairs	Maximum correlated
Positive correlation	217	91
Negative correlation	131	66

Table 3.10 positive and negative correlated Pro.Cod and lncRNA pairs

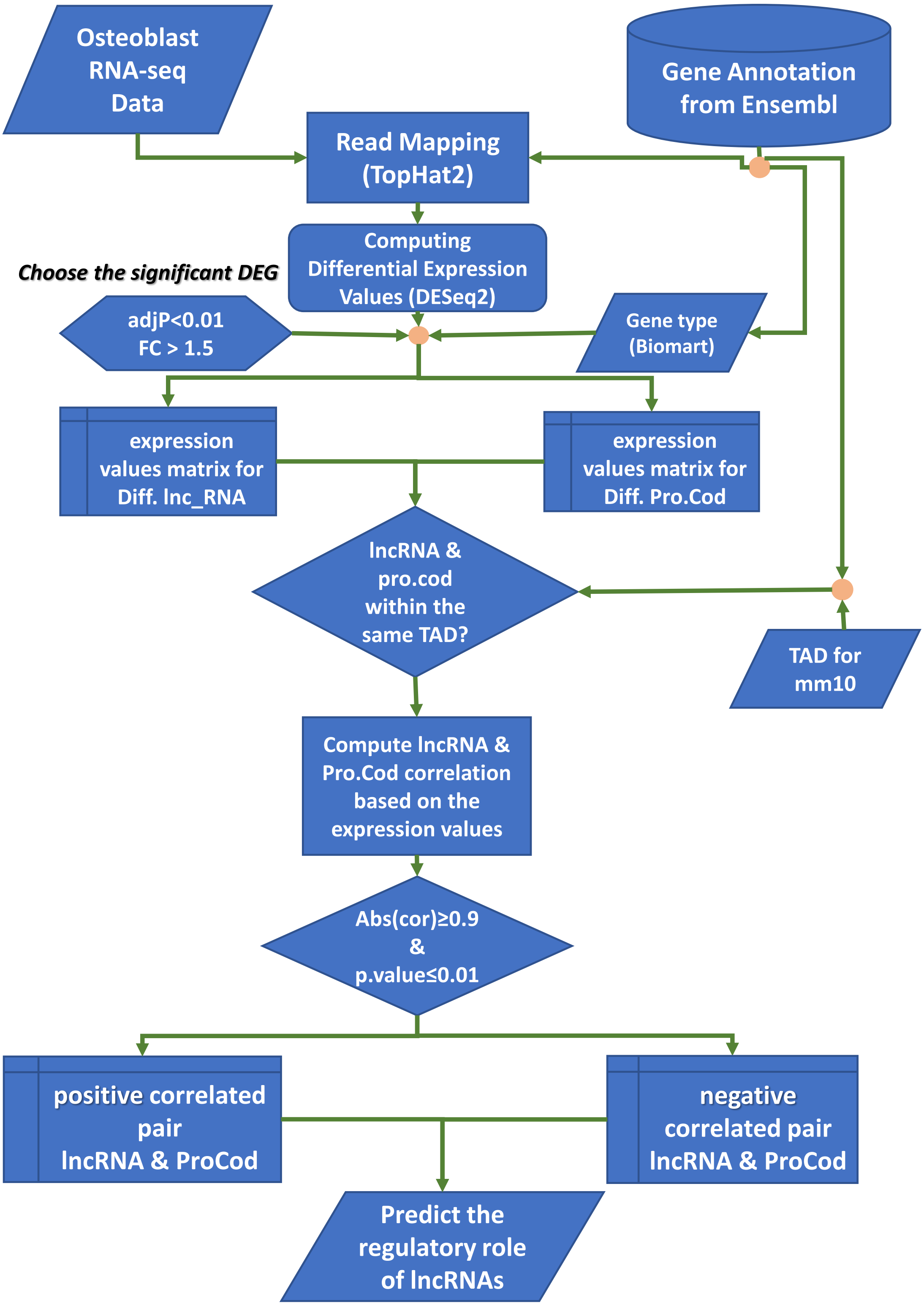
Osteoblast RNA-seq Data
Gene Annotation from Ensembl
Read Mapping (TopHat2)
Computing Differential Expression Values (DESeq2)
Choose the significant DEG
adjP<0.01
FC > 1.5
Gene type (Biomart)
expression values matrix for Diff. lnc_RNA
expression values matrix for Diff. Pro.Cod
lncRNA & pro.cod within the same TAD?
TAD for mm10
Compute lncRNA & Pro.Cod correlation based on the expression values
Abs(cor)≥0.9
&
p.value≤0.01
positive correlated pair lncRNA & ProCod
negative correlated pair lncRNA & ProCod
Predict the regulatory role of lncRNAs

Figure 3.23 lncRNA correlation algorithm

4. DISCUSSION

Despite the availability of the web-based platform for RNA-Seq data analysis as Galaxy [139], our pipeline analysis in this doctoral research still carries novel and auxiliary approaches, which provide integrity, and improve the performance of the existing tools. It provides a solution to input a set of samples into a tool, by means of a single function command. Furthermore, our pipeline provides informative results, allowing straightforward interpretation by the biologists. To discuss the novelty and the benefits of our research, I will go through the pipeline's procedures one by one as follows:

1- We coded three functions (fastqc_insert, trimo_insert, tophat_insert) to input a set of samples to FastQC, Trimmomatic and TopHat tools respectively. Despite the simplicity of the codes, they serve a good purpose by facilitating the input of samples, where the user simply needs to apply accessible paths, where the input data is stored, and where the outputs of the tools are needed to be stored. The user doesn't need to have any skills in bash script or shell command-line to run those tools. These inserting functions can be useful cores when we design an integrated web-platform for RNA-Seq data analysis, since we as the operators don't need huge data-storage to analyze the data or worry about server maintenance, and on the other hand the users don't need to upload their big fastq files elsewhere and spend hours and interrupt their network connection.

2- The quality control tool FastQC returns two sets of files for each sample (one for the forward reads and the other for reverse reads), this means for a small experiment, it will produce a range of 20 output sets. Retrieving quality information from the FastQC files will be a burden to go through the plain text files for each sample. Therefore, we coded two functions; "BasicStatistic" to retrieve the length of reads and the number of reads, before and after trimming. And "QualityScore" to calculate the mean quality of all bases in a fastq file. Furthermore, we coded "read_qual" function to get average quality in each base along reads positions, and to plot the mean quality of bases across the reads positions.

3- The coverage uniformity of the gene features is one of the concerns when using libraries based on polyadenylated RNA. Although there are tools to plot graphs and heatmap represent the coverage along the gene body as RSeQC[140], our method to check the coverage uniformity gives a detailed vision of the coverage distribution along the gene's features, by plotting the density of mean of base coverage after separating the first exons in 5′UTR, the last exon in 3′URT, and the middle exons.

4- In the differential gene expression pipeline, we defined the significant differential expressed genes using the statistical model of DESeq [123]. However, we improved the default pipeline, the purpose being to obtain informative outputs from the differential gene expression matrix. We proposed a procedure to set the suitable cutoffs of the *P*-value and logarithm fold change as described in the "Setting Thresholds:" section. Apart from this, we defined the biotype of each significant differentially expressed gene, so we could do the gene enrichment for the protein-coding genes and separate the long non-coding RNA for our further analysis. Furthermore, based on the differential gene expression matrix, we proposed the analyzing procedure for the multiple conditions experiment, we called it "ON/OFF genes". This procedure suggests defining the genes that are completely silent during a biological condition while it is expressed in the others. Using this analysis, we can get a set of genes that are uniquely regulated between biological conditions.

5- For alternative splicing analysis, we improved the performance of the differential exon usage tool DEXSeq[133], by defining whether the differential used exon (or part of the exon) is a coding exon within the ORF. Followed by the comparison of the domains of the transcripts that contain the differentially expressed exon, to determine the functions or the products that are affected by the alternative splicing of the gene.

6- long non-coding RNA (lncRNA) species have been identified whose loci locate both within and between protein coding genes. While lncRNAs remain the most enigmatic ncRNA species in terms of function, there is now much effort centered on their functional characterization and their molecular mechanisms in different cell types[141]. However, our method is concentrated on finding the potential interaction between the lncRNA and the protein-coding genes, by finding the expression correlation between the lncRNA and protein coding genes that are within the same Topological Associated Domain (TAD). Although we interduce our method as a novel approach, it is based on existing and approved researches. TAD is a known genome architecture, it is a self-interacting genomic region, meaning that DNA sequences within a TAD physically interact with each other more frequently than with sequences outside the TAD [137]. And defining the gene ontology terms of lncRNA by finding the expression correlation with protein-coding genes [138]. However, LncRNA2Function tool defines the co-expression without taking in consideration the topological associated domains. Furthermore, this database is available for GO terms in human genome, where our method can be applicable to any RNA-seq experiment.

REFERENCES

[1] M. L. Metzker, “Sequencing technologies - the next generation.,” *Nat. Rev. Genet.*, vol. 11, no. 1, pp. 31–46, 2010.

[2] Z. Wang, M. Gerstein, and M. Snyder, “RNA-Seq: a revolutionary tool for transcriptomics.,” *Nat. Rev. Genet.*, vol. 10, no. 1, pp. 57–63, 2009.

[3] Korpelainen Eija and Tuimala Jarno, *RNA-seq Data Analysis: A Practical Approach - Eija Korpelainen, Jarno Tuimala, Panu Somervuo, Mikael Huss, Garry Wong - Google Books*. CRC Press, 2015.

[4] S. N. Peirson and J. N. Butler, “RNA Extraction From Mammalian Tissues BT - Circadian Rhythms: Methods and Protocols,” E. Rosato, Ed. Totowa, NJ: Humana Press, 2007, pp. 315–327.

[5] S. R. Gallagher and P. R. Desjardins, “Quantitation of DNA and RNA with absorption and fluorescence spectroscopy.,” *Curr. Protoc. Mol. Biol.*, vol. Appendix 3, p. Appendix 3D, 2006.

[6] A. Masotti and T. Preckel, “Analysis of small RNAs with the Agilent 2100 Bioanalyzer,” *Nat. Methods*, vol. 3, no. 8, p. 62507, 2006.

[7] J. Pease and R. Sooknanan, “A rapid, directional RNA-seq library preparation workflow for Illumina[reg] sequencing,” *Nat Meth*, vol. 9, no. 3, Mar. 2012.

[8] J. Pease and R. Sooknanan, “A rapid, directional RNA-seq library preparation workflow for Illumina sequencing,” *Nat. Publ. Gr.*, vol. 9, 2012.

[9] Illumina, “Illumina Sequencing Technology.”

[10] E. R. Mardis, “Next-Generation DNA Sequencing Methods,” *Annu. Rev. Genomics Hum. Genet.*, vol. 9, no. 1, pp. 387–402, 2008.

[11] “An introduction to Next-Generation Sequencing Technology I. Welcome to Next-Generation Sequencing.”

[12] S. Goodwin, J. D. McPherson, and W. R. McCombie, “Coming of age: ten years of next-generation sequencing technologies,” *Nat Rev Genet*, vol. 17, no. 6, pp. 333–351, 2016.

[13] E. R. Mardis, “The impact of next-generation sequencing technology on genetics,” *Trends in Genetics*, vol. 24, no. 3. pp. 133–141, 2008.

[14] A. Diekstra *et al.*, “Translating sanger-based routine DNA diagnostics into generic massive parallel ion semiconductor sequencing,” *Clin. Chem.*, vol. 61, no. 1, pp. 154–162, 2015.

[15] A. Rhoads and K. F. Au, “PacBio Sequencing and Its Applications,” *Genomics, Proteomics and Bioinformatics*, vol. 13, no. 5. pp. 278–289, 2015.

[16] K. J. Travers, C. S. Chin, D. R. Rank, J. S. Eid, and S. W. Turner, “A flexible and efficient template format for circular consensus sequencing and SNP detection,” *Nucleic Acids Res.*, vol. 38, no. 15, p. e159, 2010.

[17] A. Mccarthy, “Third Generation DNA Sequencing: Pacific Biosciences’ Single Molecule Real Time Technology,” *Chem. Biol.*, vol. 17, pp. 675–676, 2010.

[18] J. Eid *et al.*, “Real-time DNA sequencing from single polymerase molecules.,” *Science*, vol. 323, no. 5910, pp. 133–8, 2009.

[19] BGI Platform, “Ion Torrent Sequencing Workflow.” [Online]. Available: http://www.genomics.cn/en/navigation/show_navigation?nid=2640. [Accessed: 24-Aug-2017].

[20] S. Koren and A. M. Phillippy, “One chromosome, one contig: Complete microbial genomes from long-read sequencing and assembly,” *Current Opinion in Microbiology*, vol. 23. pp. 110–120, 2015.

[21] J. J. Goetz and J. M. Trimarchi, “Transcriptome sequencing of single cells with Smart-Seq,” *Nat Biotech*, vol. 30, no. 8, pp. 763–765, Aug. 2012.

[22] Y. Feng, Y. Zhang, C. Ying, D. Wang, and C. Du, “Nanopore-based fourth-generation DNA sequencing technology,” *Genomics, Proteomics and Bioinformatics*, vol. 13, no. 1. pp. 4–16, 2015.

[23] “Nanopore Sequencing - MIT Technology Review.” [Online]. Available: http://www2.technologyreview.com/news/427677/nanopore-sequencing/. [Accessed: 24-Aug-2017].

[24] P. Svoboda and A. Di Cara, “Hairpin RNA: A secondary structure of primary importance,” *Cellular and Molecular Life Sciences*, vol. 63, no. 7–8. pp. 901–918, 2006.

[25] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, “Mapping and quantifying mammalian transcriptomes by RNA-Seq,” *Nat. Methods*, vol. 5, no. 7, pp. 621–628, 2008.

[26] Shanrong Zhao and Baohong Zhang, *Next Generation Sequencing - Advances, Applications and Challenges , Chapter 16*, vol. 3. InTech.

[27] “gene_structure.” [Online]. Available: http://disi.unitn.it/~teso/courses/sciprog/_images/genestructure.jpg. [Accessed: 02-Sep-2017].

[28] M. Chen *et al.*, “Improvement of genome assembly completeness and identification of novel full-length protein-coding genes by RNA-seq in the giant panda genome.,” *Sci. Rep.*, vol. 5, no. November, p. 18019, 2015.

[29] C. Trapnell *et al.*, “Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation,” *Nat. Biotechnol.*, vol. 28, no. 5, pp. 511–515, 2010.

[30] J. Peltonen, V. Aarnio, L. Heikkinen, M. Lakso, and G. Wong, “Chronic ethanol exposure increases cytochrome P-450 and decreases activated in blocked unfolded protein response gene family transcripts in caenorhabditis elegans,” *J. Biochem. Mol. Toxicol.*, vol. 27, no. 3, pp. 219–228, 2013.

[31] A. C. Nica and E. T. Dermitzakis, “Expression quantitative trait loci: present and future.,” *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, vol. 368, no. 1620, p. 20120362, 2013.

[32] J. Majewski and T. Pastinen, “The study of eQTL variations by RNA-seq: From SNPs to phenotypes,” *Trends in Genetics*, vol. 27, no. 2. pp. 72–79, 2011.

[33] E. Lalonde *et al.*, “RNA sequencing reveals the role of splicing polymorphisms in regulating human gene expression,” *Genome Res.*, vol. 21, no. 4, pp. 545–554, 2011.

[34] F. Tang *et al.*, “mRNA-Seq whole-transcriptome analysis of a single cell,” *Nat. Methods*, vol. 6, no. 5, pp. 377–382, 2009.

[35] T. Hashimshony, F. Wagner, N. Sher, and I. Yanai, “CEL-Seq: Single-Cell RNA-Seq by Multiplexed Linear Amplification,” *Cell Rep.*, vol. 2, no. 3, pp. 666–673, 2012.

[36] H. Edgren *et al.*, “Identification of fusion genes in breast cancer by paired-end RNA-sequencing.,” *Genome Biol.*, vol. 12, no. 1, p. R6, 2011.

[37] H. Edgren *et al.*, “Identification of fusion genes in breast cancer by paired-end RNA-sequencing,” *Genome Biol.*, vol. 12, no. 1, 2011.

[38] R. Fani, M. Brilli, M. Fondi, and P. Lió, “The role of gene fusions in the evolution of metabolic pathways: the histidine biosynthesis case,” *BMC Evol. Biol.*, vol. 7, no. 2, 2006.

[39] E. M. Quinn *et al.*, “Development of Strategies for SNP Detection in RNA-Seq Data: Application to Lymphoblastoid Cell Lines and Evaluation Using 1000 Genomes Data,” *PLoS One*, vol. 8, no. 3, 2013.

[40] A. Djari *et al.*, “Gene-based single nucleotide polymorphism discovery in bovine muscle using next-generation transcriptomic sequencing,” *BMC Genomics*, vol. 14, p. 17, 2013.

[41] C. Ku *et al.*, “Exome versus transcriptome sequencing in identifying coding region variants.,” *Expert Rev. Mol. Diagn.*, vol. 12, no. 3, pp. 241–51, 2012.

[42] N. E. Ilott and C. P. Ponting, “Predicting long non-coding RNAs using RNA sequencing,” *Methods*, vol. 63, no. 1, pp. 50–59, 2013.

[43] M. A. Faghihi *et al.*, “Expression of a noncoding RNA is elevated in Alzheimer’s disease and drives rapid feed-forward regulation of β-secretase,” *Nat. Med.*, vol. 14, no. 7, pp. 723–730, 2008.

[44] J. S. Mattick and I. V. Makunin, “Non-coding RNA.,” *Human molecular genetics*, vol. 15 Spec No. 2006.

[45] J. Srinivasan *et al.*, “The draft genome and transcriptome of Panagrellus redivivus are shaped by the harsh demands of a free-living lifestyle,” *Genetics*, vol. 193, no. 4, pp. 1279–1295, 2013.

[46] T. R. Mercer *et al.*, “Targeted RNA sequencing reveals the deep complexity of the human transcriptome,” *Nat. Biotechnol.*, vol. 30, no. 1, pp. 99–104, 2011.

[47] T. Komori, “Regulation of osteoblast differentiation by transcription factors,” *Journal of Cellular Biochemistry*, vol. 99, no. 5. pp. 1233–1239, 2006.

[48] G. D. Roodman, “Cell biology of the osteoclast,” *Experimental Hematology*, vol. 27, no. 8. pp. 1229–1241, 1999.

[49] K. K. Papachroni, D. N. Karatzas, K. A. Papavassiliou, E. K. Basdra, and A. G. Papavassiliou, “Mechanotransduction in osteoblast regulation and bone disease,” *Trends in Molecular Medicine*, vol. 15, no. 5. pp. 208–216, 2009.

[50] V. Kumar, A. K. Abbas, J. C. Aster, and N. Fausto, *Robbins & Cotran pathologic basis of disease*, 8th ed. Philadelphia: Saunders, 2009.

[51] J. F. Charles and A. O. Aliprantis, “Osteoclasts: More than ‘bone eaters,’” *Trends in Molecular Medicine*, vol. 20, no. 8. pp. 449–459, 2014.

[52] J. Caetano-Lopes, H. Canhão, and J. E. Fonseca, “Osteoblasts and bone formation.,” *Acta Reum. Port.*, vol. 32, no. 2, pp. 103–110, 2007.

[53] A. Rutkovskiy, K.-O. Stensløkken, and I. J. Vaage, “Osteoblast Differentiation at a Glance.,” *Med. Sci. Monit. Basic Res.*, vol. 22, pp. 95–106, 2016.

[54] B. F. Boyce, “Advances in the Regulation of Osteoclasts and Osteoclast Functions,” *J. Dent. Res.*, vol. 92, no. 10, pp. 860–867, 2013.

[55] Jensen, Gopalakrishnan, and J. J. Westendorf, “Regulation of Gene Expression in Osteoblasts,” *Biofactors*, vol. 36, no. 1, pp. 25–32, 2010.

[56] R. Quarto *et al.*, “Repair of large bone defects with the use of autologous bone marrow stromal cells.,” *N. Engl. J. Med.*, vol. 344, no. 5, pp. 385–386, 2001.

[57] E. Canalis, “Notch signaling in osteoblasts,” *Sci. Signal.*, vol. 1, no. 17, p. pe17, 2008.

[58] F. Long, “Building strong bones: molecular regulation of the osteoblast lineage,” *Nat. Rev. Mol. Cell Biol.*, vol. 13, no. 1, pp. 27–38, 2011.

[59] T. Komori *et al.*, “Targeted Disruption of Cbfa1 Results in a Complete Lack of Bone Formation owing to Maturational Arrest of Osteoblasts,” *Cell*, vol. 89, no. 5, pp. 755–764, 1997.

[60] D. L. Black, “Mechanisms of Alternative Pre-Messenger RNA Splicing,” *Annu. Rev. Biochem.*, vol. 72, no. 1, pp. 291–336, 2003.

[61] J. F. Cáceres and A. R. Kornblihtt, “Alternative splicing: Multiple control mechanisms and involvement in human disease,” *Trends in Genetics*, vol. 18, no. 4. pp. 186–193, 2002.

[62] A. J. Matlin, F. Clark, and C. W. J. Smith, “Understanding alternative splicing: towards a cellular code,” *Nat. Rev. Mol. Cell Biol.*, vol. 6, no. 5, pp. 386–398, 2005.

[63] “Alternative RNA splicing.” [Online]. Available: http://www.exonhit.com/technology/alternative-rna-splicing. [Accessed: 21-Sep-2017].

[64] M. Sammeth, S. Foissac, and R. Guigó, “A General Definition and Nomenclature for Alternative Splicing Events,” *PLoS Comput Biol*, vol. 4, no. 8, 2008.

[65] P. J. A. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice, “The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants,” *Nucleic Acids Res.*, vol. 38, no. 6, pp. 1767–1771, 2009.

[66] Simon Andrews, “Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data.” [Online]. Available: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/. [Accessed: 10-Aug-2017].

[67] L. Wang, S. Wang, and W. Li, “RSeQC: quality control of RNA-seq experiments.,” *Bioinformatics*, vol. 28, no. 16, pp. 2184–5, 2012.

[68] “Trimmomatic Manual: V0.30.”

[69] S. Anders, P. T. Pyl, and W. Huber, “HTSeq-A Python framework to work with high-throughput sequencing data,” *Bioinformatics*, vol. 31, no. 2, pp. 166–169, 2015.

[70] B. Ewing, L. Hillier, M. C. Wendl, and P. Green, “Base-calling of automated sequencer traces using

phred. I. Accuracy assessment.," *Genome Res.*, vol. 8, no. 3, pp. 175–85, Mar. 1998.

[71] B. Ewing and P. Green, "Base-Calling of Automated Sequencer Traces Using Phred. II. Error Probabilities," *Genome Res.*, vol. 8, no. 3, pp. 186–194, Mar. 1998.

[72] A. E. Men, P. Wilson, K. Siemering, and S. Forrest, "Sanger DNA Sequencing," in *Next Generation Genome Sequencing: Towards Personalized Medicine*, 2008, pp. 1–11.

[73] S. Bennett, "Solexa Ltd," *Pharmacogenomics*, vol. 5, no. 4, pp. 433–438, Jun. 2004.

[74] I. Illumina, "Sequencing Analysis Software User Guide," *Illumina Inc, San Diego, CA, USA.*, 2008.

[75] J.-W. Li *et al.*, "The NGS WikiBook: a dynamic collaborative online training effort with long-term sustainability," *Brief. Bioinform.*, vol. 14, 2013.

[76] A. M. Bolger, M. Lohse, and B. Usadel, "Trimmomatic: a flexible trimmer for Illumina sequence data," vol. 30, no. 15, pp. 2114–2120, 2014.

[77] H. Li and N. Homer, "A survey of sequence alignment algorithms for next-generation sequencing."

[78] H. Li *et al.*, "The Sequence Alignment/Map format and SAMtools," *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, 2009.

[79] 1000 Genome Project Data and P. Subgroup, "Sequence Alignment/Map Format Specification," *github/SamTools*, no. June, pp. 1–16, 2017.

[80] SamTools, "SAM Format Specification 0.1.2-draft," *SamTools*, no. 20090820, pp. 1–13, 2009.

[81] T. F. Smite and M. S. Waterman, "Identification of Common Molecular Subsequences," *Repr. from J . Mol. Biol. J. Mol. Bwl*, vol. 147, no. 147, pp. 195–197, 1981.

[82] N. A. Fonseca, J. Rung, A. Brazma, and J. C. Marioni, "HTS Mappers." [Online]. Available: http://wwwdev.ebi.ac.uk/fg/hts_mappers/. [Accessed: 27-Aug-2017].

[83] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with Bowtie 2."

[84] H. Li and R. Durbin, "Fast and accurate long-read alignment with Burrows–Wheeler transform," *Bioinforma. Orig. Pap.*, vol. 26, no. 5, pp. 589–59510, 2010.

[85] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg, "TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions," *Genome Biol.*, vol. 14, no. 4, p. R36, 2013.

[86] A. Dobin *et al.*, "STAR: ultrafast universal RNA-seq aligner," vol. 29, no. 1, pp. 15–21, 2013.

[87] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome," *Genome Biol.*, vol. 10, 2009.

[88] C. Trapnell, L. Pachter, and S. L. Salzberg, "TopHat: Discovering splice junctions with RNA-Seq," *Bioinformatics*, vol. 25, no. 9, pp. 1105–1111, 2009.

[89] U. Pozzoli, G. Menozzi, G. P. Comi, R. Cagliani, N. Bresolin, and M. Sironi, "Intron size in mammals: complexity comes to terms with economy," *Trends Genet.*, vol. 23, no. 1, pp. 20–24, Aug. 2017.

[90] B. Li and C. N. Dewey, "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome."

[91] C. Trapnell *et al.*, "Transcript assembly and abundance estimation from RNA-Seq reveals thousands of new transcripts and switching among isoforms."

[92] A. Oshlack, M. D. Robinson, and M. D. Young, "From RNA-seq reads to differential expression results," *Genome Biol.*, 2010.

[93] S. Anders, A. Reyes, and W. Huber, "Detecting differential usage of exons from RNA-seq data."

[94] Y. Liao, G. K. Smyth, and W. Shi, "FeatureCounts: An efficient general purpose program for assigning sequence reads to genomic features," *Bioinformatics*, vol. 30, no. 7, pp. 923–930, 2014.

[95] S. Anders and W. Huber, "Differential expression analysis for sequence count data," *Genome Biol.*, vol. 11, no. R106, pp. 1–12, 2010.

[96] P. L. Auer and R. W. Doerge, "A Two-Stage Poisson Model for Testing RNA-Seq Data," *Stat. Appl. Genet. Mol. Biol.*, vol. 10, 2011.

[97] T. J. Hardcastle and K. A. Kelly, “baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data,” 2010.

[98] I. Nookaew *et al.*, “A comprehensive comparison of RNA-Seq-based transcriptome analysis from reads to differential gene expression and cross-comparison with microarrays: a case study in Saccharomyces cerevisiae ,” *Nucleic Acids Res.*, vol. 40, no. 20, pp. 10084–10097, Nov. 2012.

[99] F. Rapaport *et al.*, “Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data,” *Genome Biol.*, vol. 14, p. R95, 2013.

[100] M. Lawrence *et al.*, “Software for Computing and Annotating Genomic Ranges,” *PLoS Comput. Biol.*, vol. 9, no. 8, 2013.

[101] R. C. Gentleman *et al.*, “Bioconductor: open software development for computational biology and bioinformatics,” *Genome Biol.*, vol. 5, no. 10, 2004.

[102] S. Anders, P. T. Pyl, and W. Huber, “HTSeq—a Python framework to work with high-throughput sequencing data,” vol. 31, no. 2, pp. 166–169, 2015.

[103] A. R. Quinlan and I. M. Hall, “BEDTools: a flexible suite of utilities for comparing genomic features,” *Bioinforma. Appl. NOTE*, vol. 26, no. 6, pp. 841–84210, 2010.

[104] Y. Liao, G. K. Smyth, and W. Shi, “The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote.”

[105] A. Wei Shi, Y. Liao with contributions from Gordon Smyth, J. Dai, T. Triche, and J. Maintainer Wei Shi, “Package ‘Rsubread.’” Bioconductor, 2017.

[106] 1890-1962. Fisher, Ronald Aylmer, Sir, *The design of experiments*, 1st ed. Edinburgh : Oliver and Boyd, 1935., 1935.

[107] P. L. Auer and R. W. Doerge, “Statistical Design and Analysis of RNA Sequencing Data.”

[108] M. Bengtsson, A. Ståhlberg, P. Rorsman, and M. Kubista, “Gene expression profiling in single cells from the pancreatic islets of Langerhans reveals lognormal distribution of mRNA levels.”

[109] J. C. Marioni, C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad, “RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays,” *Genome Res.*, vol. 18, no. 9, pp. 1509–1517, 2008.

[110] L. Wang, Z. Feng, X. Wang, X. Wang, and X. Zhang, “DEGseq: An R package for identifying differentially expressed genes from RNA-seq data,” *Bioinformatics*, vol. 26, no. 1, pp. 136–138, 2009.

[111] U. Nagalakshmi *et al.*, “The Transcriptional Landscape of the Yeast Genome Defined by RNA Sequencing,” *Science (80-.).*, vol. 320, no. 5881, pp. 1344–1349, 2008.

[112] M. D. Robinson *et al.*, “Moderated statistical tests for assessing differences in tag abundance,” *Bioinformatics*, vol. 23, no. 21, pp. 2881–2887, 2007.

[113] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, “edgeR: A Bioconductor package for differential expression analysis of digital gene expression data,” *Bioinformatics*, vol. 26, no. 1, pp. 139–140, 2009.

[114] Y. H. Zhou, K. Xia, and F. A. Wright, “A powerful and flexible approach to the analysis of RNA sequence count data,” *Bioinformatics*, vol. 27, no. 19, pp. 2672–2678, 2011.

[115] H. Wu, C. Wang, and Z. Wu, “A new shrinkage estimator for dispersion improves differential expression detection in RNA-seq data,” *Biostatistics*, vol. 14, no. 2, pp. 232–243, 2013.

[116] T. J. Hardcastle and K. A. Kelly, “baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data,” *BMC Bioinformatics*, vol. 11, no. 1, p. 422, 2010.

[117] M. A. Van De Wiel, G. G. R. Leday, L. Pardo, H. Rue, A. W. Van Der Vaart, and W. N. Van Wieringen, “Bayesian analysis of RNA sequencing data by estimating multiple shrinkage priors,” *Biostatistics*, vol. 14, no. 1, pp. 113–128, 2013.

[118] C.-E. Ott *et al.*, “Promiscuous and Depolarization-Induced Immediate-Early Response Genes Are Induced by Mechanical Strain of Osteoblasts,” *J. Bone Miner. Res.*, vol. 24, no. 7, pp. 1247–1262, 2009.

[119] “Bowtie 2: fast and sensitive read alignment.” [Online]. Available: http://bowtie-bio.sourceforge.net/bowtie2/index.shtml. [Accessed: 28-Aug-2017].

[120] “iGenomes.” [Online]. Available: https://support.illumina.com/sequencing/sequencing_software/igenome.html. [Accessed: 28-Aug-2017].

[121] C. et al Trapnell, “TopHat2 Manual.” [Online]. Available: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btp120.

[122] “BED format / Genome Browser.” [Online]. Available: http://genome.ucsc.edu/FAQ/FAQformat.html#format1. [Accessed: 28-Aug-2017].

[123] M. I. Love, W. Huber, and S. Anders, “Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2.,” *Genome Biol.*, vol. 15, no. 12, p. 550, 2014.

[124] J. A. Nelder and R. W. M. Wedderburn, “Generalized Linear Models,” *J. R. Stat. Soc. Ser. A*, vol. 135, no. 3, p. 370, 1972.

[125] Y. Benjamini and Y. Hochberg, “Controlling the false discovery rate: a practical and powerful approach to multiple testing,” *Journal of the Royal Statistical Society B*, vol. 57, no. 1. pp. 289–300, 1995.

[126] O. J. Dunn, “Estimation of the Means of Dependent Variables,” *Ann. Math. Stat.*, vol. 29, no. 4, pp. 1095–1111, 1958.

[127] O. J. Dunn, “Multiple Comparisons Among Means,” *J. Am. Stat. Assoc.*, vol. 56, no. 293, p. 52, 1961.

[128] K. Pearson, “On Lines and Planes of Closest Fit to Systems of Points in Space,” *Philos. Mag.*, vol. 2, no. 11, pp. 559–572, 1905.

[129] S. Bauer, S. Grossmann, M. Vingron, and P. N. Robinson, “Ontologizer 2.0 - A multifunctional tool for GO term enrichment analysis and data exploration,” *Bioinformatics*, vol. 24, no. 14, pp. 1650–1651, 2008.

[130] S. Bauer, J. Gagneur, and P. N. Robinson, “Going Bayesian: Model-based gene set analysis of genome-scale data,” *Nucleic Acids Res.*, vol. 38, no. 11, pp. 3523–3532, 2010.

[131] S. Grossmann, S. Bauer, P. N. Robinson, and M. Vingron, “Improved detection of overrepresentation of Gene-Ontology annotations with parent-child analysis,” *Bioinformatics*, vol. 23, no. 22, pp. 3024–3031, 2007.

[132] Z. YUECHUN *et al.*, “Lrrc75b is a novel negative regulator of C2C12 myogenic differentiation,” *Int. J. Mol. Med.*, vol. 38, pp. 1411–1418, 2016.

[133] S. Anders, A. Reyes, and W. Huber, “Detecting diferential usage of exons from RNA-seq data,” *Genome Res*, vol. 22, no. 10, pp. 2008–2017, 2012.

[134] D. R. Cox and N. Reid, “Parameter orthogonality and approximate conditional inference,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 49. pp. 1–39, 1987.

[135] G. K. Smyth and A. P. Verbyla, “A Conditional Likelihood Approach to Residual Maximum Likelihood Estimation in Generalized Linear Models,” *J. R. Stat. Soc. Ser. B*, vol. 58, no. 3, pp. 565–572, 1996.

[136] J. M. Johnson, “Genome-Wide Survey of Human Alternative Pre-mRNA Splicing with Exon Junction Microarrays,” *Science (80-.).*, vol. 302, no. 5653, pp. 2141–2144, 2003.

[137] A. Pombo and N. Dillon, “Three-dimensional genome architecture: players and mechanisms,” *Nat. Rev. Mol. Cell Biol.*, vol. 16, no. 4, pp. 245–257, 2015.

[138] Q. Jiang *et al.*, “LncRNA2Function: a comprehensive resource for functional investigation of human lncRNAs based on RNA-seq data,” *BMC Genomics*, vol. 16, no. Suppl 3, p. S2, 2015.

[139] “Galaxy.” [Online]. Available: https://usegalaxy.org/. [Accessed: 11-Oct-2017].

[140] L. Wang, S. Wang, and W. Li, “RSeQC: quality control of RNA-seq experiments,” *Bioinformatics*, vol. 28, no. 16, pp. 2184–2185, Aug. 2012.

[141] M. Cabili *et al.*, “Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses,” *Genes Dev.*, vol. 25, no. 18, pp. 1915–1927, 2011.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DGE:	Differential Gene Expression	**adj.p:**	Adjusted p-value
DE:	Differentially Expressed	**ORF:**	Open Read Frame
DEU:	Differential Exons Usage	**PCA:**	Principle Component Analysis
FC:	Fold change	**GO:**	Gene Ontology
LFC:	Logarithm Fold Chnage	**lncRNA:**	Long Non-Coding RNA
bp:	Base pair	**Pro.Cod**	Protein Coding

APPENDIX I : ON-OFF GENES

Gene	Day 0	Day3	Day6	Day12	Note
LRRC66	OFF	OFF	ON	ON	bone resorption and osteoclastic acid secretion
IL20RA	OFF	ON	OFF	ON	regulation of bone resorption and in skeleton phenotype
LCN2	ON	ON	OFF	ON	anti-osteoclastogenic, suppressing the proliferation and differentiation of osteoclast
MEPE	OFF	ON	ON	ON	expressed by osteocytes, within mineralized bone
IFNE	ON	OFF	OFF	OFF	cell proliferation and activation of the immune system
HRK	ON	OFF	OFF	OFF	encode apoptosis regulatory proteins
Hmx3	ON	OFF	OFF	OFF	cell differentiation
FA2H	ON	OFF	OFF	OFF	cell proliferation, lipid modification
CHRND	ON	OFF	OFF	OFF	inhibits osteoclastogenesis
NEURL3	ON	OFF	OFF	OFF	zinc ion binding, upregulated in first 24 hours of osteoblast differentiation
Dll4	ON	OFF	OFF	OFF	Skull defect, Finger syndactyly, Abnormality of the metacarpal bones
GBP4	ON	OFF	OFF	OFF	differentially expressed in osteoclasts from patients of Autosomal dominant osteopetrosis type II
Cntnap3	ON	OFF	OFF	OFF	related to BMP signaling and chondrogenic differentiation
CACNA1I	ON	OFF	OFF	OFF	control the rapid entry of Ca(2+)
Gpr132	ON	OFF	Low	Low	Runx2 directly regulates a unique set of cell cycle genes Gpr132,Runx2 is expressed in both chondrocytes and osteoblasts
DCC	ON	OFF	Low	Low	involved in the pathogenesis of skeletal diseases including osteoporosis and arthritis
MCTP1	ON	OFF	Low	Low	calcium-dependent phospholipid binding
ADORA2A	ON	Low	Low	OFF	Adenosine regulates bone metabolism, differentiation of mesenchymal stem cell to osteoblasts and adipocytes
PPP1R1B	ON	Low	Low	OFF	Midbrain dopaminergic neurons
ECEL1	ON	OFF	OFF	ON	important role for these enzymes in bone metabolism and osteoblasts
DTHD1	ON	OFF	OFF	ON	mediates apoptosis in human osteoblast
RASAL1	ON	OFF	OFF	Low	ossifying fibromas compared with fibrous dysplasia
PIK3AP1	ON	OFF	OFF	Low	in the protection of macrophages from apoptosis induced by endoplasmic reticulum stress
Rbm24	ON	OFF	OFF	Low	regulates myogenic differentiation
KCNK3	ON	ON	OFF	OFF	control the resting membrane potential in many cell types
LRRC4	ON	ON	OFF	OFF	overexpress in brain and glioblastoma cells
NRG3	ON	OFF	ON	OFF	regulation of cell growth

Gene	Day 0	Day3	Day6	Day12	Note
Ifi205	OFF	ON	ON	ON	plays a role in adipogenic differentiation of mouse adipose-derived stem cells
ArhGAP15	OFF	ON	ON	ON	controls neutrophil mobilization from the bone marrow
BMP15	OFF	ON	ON	ON	expressed in oocytes , osteoblastic cell
KLHL4	OFF	ON	ON	ON	GO : actin binding
FRMPD4	OFF	ON	ON	ON	regulates dendritic spine morphogenesis
Il12a	OFF	ON	ON	ON	In AIA, PRL treatment lowered osteoclast density
Hist2h3c2	OFF	ON	ON	ON	encodes a replication-dependent histone
Gjb5	OFF	ON	ON	ON	suppressed cell migration and invasion
Ppbp	OFF	ON	ON	ON	platelet basic protein
FABP7	OFF	ON	ON	ON	FABP7 and FABP5 are differentially expressed in oligodendrocytel cells and regulate their proliferation and/or differentiation.
NR1H4 (FXR)	OFF	ON	ON	ON	new role for FXR in the modulation of osteoblast/adipocyte balance
GALNT15	OFF	ON	ON	ON	metal ion binding.
PTPN20	OFF	ON	ON	ON	protein tyrosine phosphatase
ATP12A	OFF	ON	ON	ON	Oxidative phosphorylation
MMP27	OFF	ON	ON	ON	calcium ion binding , zinc ion binding, collagen catabolic process
Slfn2	OFF	ON	ON	ON	involved in hematopoietic and immune processes
MYH13	OFF	ON	ON	ON	actin binding, calmodulin binding , muscle myosin complex
Ccl11	OFF	ON	ON	ON	it plays an important role in cartilage degradation in osteoarthritis
Sprr3	OFF	ON	ON	ON	keratinocyte differentiation, keratinization
CCDC27	OFF	ON	ON	ON	is a positional candidate genes for low bone mineral density
Lingo2	OFF	ON	ON	ON	ssociated with essential tremor and Parkinson disease.
ADCYAP1R1	OFF	ON	ON	ON	(pituitary),regulation of calcium ion , cell death, inositol phosphate biosynthetic process, cell differentiation
Tinag	OFF	ON	ON	ON	GO immune response, cell adhesion, polysaccharide binding.
Il13ra2	OFF	ON	ON	ON	differentiation and activation of osteoblasts

APPENDIX II: lnc_RNA CORRELATION

LincRNA	Protein_cod	Correlation	Pvalue	TAD_id
Gm37233	Dst	-0.96121575	6.47508E-07	chr1_33663155_34463155
Pantr1	Mrps9	-0.934765657	8.33333E-06	chr1_40823155_43103155
Gm29157	1500015O10Rik	0.973851527	9.21426E-08	chr1_43103155_43743155
Gm28818	Atic	-0.962644304	5.38024E-07	chr1_71553426_72233426
Gm28818	Mreg	-0.903268137	5.65931E-05	chr1_71553426_72233426
2810414N06Rik	Fam129a	-0.957655318	9.98494E-07	chr1_151432870_152352870
Gm16701	Fam78b	0.910379006	3.91111E-05	chr1_166389869_167349869
Gm17275	Lin9	-0.92915666	1.24671E-05	chr1_180229869_180829869
Gm17275	Acbd3	0.935325129	7.98969E-06	chr1_180229869_180829869
Gm17275	H3f3a	-0.909034732	4.2036E-05	chr1_180229869_180829869
Gm34342	Marc2	0.941843912	4.74963E-06	chr1_183503931_184816121
Gm29233	Thbs1	0.937151432	6.94519E-06	chr2_117494264_119014264
Gm29233	Bub1b	0.93289764	9.56619E-06	chr2_117494264_119014264
Gm29233	Knstrn	0.929139794	1.24816E-05	chr2_117494264_119014264
Gm26899	Rpap1	0.973021857	1.07568E-07	chr2_119654264_121494264
Gm26899	Tyro3	0.945248932	3.53317E-06	chr2_119654264_121494264
Gm26899	Pla2g4b	0.933644451	9.05714E-06	chr2_119654264_121494264
Gm26899	Vps39	0.904408504	5.34403E-05	chr2_119654264_121494264
Gm26899	Ganc	-0.914667635	3.08341E-05	chr2_119654264_121494264
Gm26899	Zfp106	0.957993781	9.59771E-07	chr2_119654264_121494264
Gm26899	Snap23	-0.922628823	1.9156E-05	chr2_119654264_121494264
Gm26899	Stard9	0.904999012	5.18628E-05	chr2_119654264_121494264
Gm26899	Cdan1	0.954733039	1.3871E-06	chr2_119654264_121494264
Gm26899	Zscan29	0.92185492	2.01066E-05	chr2_119654264_121494264
Gm26899	Ppip5k1	0.936840642	7.11484E-06	chr2_119654264_121494264
Gm14230	Mybl1	0.909835188	4.02744E-05	chr1_9569919_10129919
A530013C23Rik	Mybl1	-0.935632771	7.80557E-06	chr1_9569919_10129919
Gm26617	Apoo	-0.965734257	3.51225E-07	chrX_94194661_95964661
Gm26617	Zxdb	0.995181022	2.03018E-11	chrX_94194661_95964661
Gm26617	Spin4	-0.930210904	1.15875E-05	chrX_94194661_95964661
Gm26617	Zc4h2	-0.915596799	2.92382E-05	chrX_94194661_95964661
Gm17308	Pex2	0.919235405	2.36037E-05	chr3_3000000_6320000
Gm30074	Naa15	-0.9226284	1.91565E-05	chr3_51196078_52236078
4921539H07Rik	Pfn2	0.948249014	2.67924E-06	chr3_57436078_58476078
Gm37933	Fhdc1	-0.921859835	2.01005E-05	chr3_84316078_86156078
Gm37933	Glt28d2	0.929057701	1.25523E-05	chr3_84316078_86156078
Gm4349	Pip5k1a	0.929325602	1.23227E-05	chr3_95036078_96476077
Gm4349	Vps72	0.901813801	6.0824E-05	chr3_95036078_96476077
Gm4349	Lysmd1	0.94840478	2.63986E-06	chr3_95036078_96476077
Gm4349	Prune	0.975938189	6.10092E-08	chr3_95036078_96476077
Gm4349	Setdb1	0.97830312	3.65146E-08	chr3_95036078_96476077
Gm4349	Arnt	0.983191581	1.02726E-08	chr3_95036078_96476077
Gm4349	Golph3l	-0.923014206	1.8696E-05	chr3_95036078_96476077
Gm4349	Ensa	-0.937362648	6.83173E-06	chr3_95036078_96476077
Gm4349	Rprd2	0.933944969	8.85844E-06	chr3_95036078_96476077
Gm4349	Mrps21	-0.956477972	1.14296E-06	chr3_95036078_96476077
Gm4349	Aph1a	0.948278129	2.67184E-06	chr3_95036078_96476077
Gm4349	Anp32e	-0.918741938	2.4313E-05	chr3_95036078_96476077
Gm4349	Plekho1	0.907273173	4.61264E-05	chr3_95036078_96476077
Gm4349	Vps45	-0.959306968	8.20665E-07	chr3_95036078_96476077
Gm4349	Otud7b	0.958770403	8.7542E-07	chr3_95036078_96476077
Gm26530	S1pr1	-0.960656863	6.94869E-07	chr3_115657082_116897082
Gm26530	Agl	0.975138339	7.17497E-08	chr3_115657082_116897082
A730020M07Rik	Cnn3	0.966003039	3.37818E-07	chr3_121217082_121777082
A730020M07Rik	Slc44a3	0.961632267	6.13911E-07	chr3_121217082_121777082
Gm43254	Tet2	-0.958088602	9.4914E-07	chr3_132697036_134657036
Gm17501	Ddah1	0.958183172	9.3863E-07	chr3_145377036_145937036
AI427809	Abca1	-0.964130354	4.40295E-07	chr4_53027128_53667128
Gm12415	Cntln	0.964055756	4.44837E-07	chr4_83594096_85114096
Gm26516	Cops5	-0.904920465	5.20705E-05	chr1_9569919_10129919
Gm17300	Sgk3	0.952978854	1.67253E-06	chr1_9569919_10129919
Gm20707	Sgk3	0.955251851	1.31056E-06	chr1_9569919_10129919
Gm13067	Spsb1	0.981633816	1.5959E-08	chr4_149745891_150145891
Gm13067	Gpr157	0.973488209	9.8664E-08	chr4_149745891_150145891
Gm26840	Tmem88b	0.902132401	5.98765E-05	chr4_154865891_156256011
Gm26840	Ccnl2	-0.954903402	1.36158E-06	chr4_154865891_156256011
6030443J06Rik	Orc5	0.954698571	1.3923E-06	chr5_21774182_23694182

LincRNA	Protein_cod	Correlation	Pvalue	TAD_id
1700096K18Rik	Mybl1	-0.903865942	5.49226E-05	chr1_9569919_10129919
1700096K18Rik	Sgk3	0.920777201	2.1492E-05	chr1_9569919_10129919
Gm4961	Hadhb	0.919282796	2.35365E-05	chr5_29553460_31057627
Gm4961	Slc5a6	-0.925452486	1.59837E-05	chr5_29553460_31057627
C130083M11Rik	Sod3	0.923907619	1.76631E-05	chr5_52088761_52848761
Gm3716	Pgm1	0.944681835	3.71639E-06	chr5_63888761_65328761
Gm3716	Tlr1	0.914815061	3.05764E-05	chr5_63888761_65328761
Gm3716	Fam114a1	0.952108913	1.83037E-06	chr5_63888761_65328761
Gm3716	Wdr19	0.914626248	3.09067E-05	chr5_63888761_65328761
1700010H22Rik	Bmp3	0.962115618	5.76674E-07	chr5_98330981_99170981
D930016D06Rik	BC005561	0.957958665	9.63733E-07	chr5_103930981_105370981
Miat	Mybl1	-0.944081859	3.91835E-06	chr1_9569919_10129919
1110006O24Rik	Wsb2	-0.91275864	3.43278E-05	chr5_115629991_117469991
4933404O12Rik	Hspb1	0.953922996	1.51365E-06	chr5_135884130_136924130
Gm8066	Cops5	-0.950978012	2.05299E-06	chr1_9569919_10129919
B230303O12Rik	Cpsf4	0.993750417	7.4298E-11	chr5_145159131_146668424
B230303O12Rik	Atp5j2	-0.962284187	5.64119E-07	chr5_145159131_146668424
B230303O12Rik	Zkscan5	-0.954158588	1.47593E-06	chr5_145159131_146668424
B230303O12Rik	Zscan25	0.983053594	1.06987E-08	chr5_145159131_146668424
B230303O12Rik	1700001J03Rik	0.954528908	1.41816E-06	chr5_145159131_146668424
B230303O12Rik	Cdk8	0.986019393	4.10898E-09	chr5_145159131_146668424
Gm43625	Cpsf4	0.971895766	1.31722E-07	chr5_145159131_146668424
Gm43625	Atp5j2	-0.923649987	1.79562E-05	chr5_145159131_146668424
Gm43625	Zkscan5	-0.927757663	1.37147E-05	chr5_145159131_146668424
Gm43625	Zscan25	0.950158002	2.22743E-06	chr5_145159131_146668424
Gm43625	1700001J03Rik	0.949664589	2.33793E-06	chr5_145159131_146668424
Gm43625	Cdk8	0.971287935	1.46445E-07	chr5_145159131_146668424
8430423G03Rik	Medag	0.901486566	6.18093E-05	chr5_148308424_149597425
5930430L01Rik	Slc7a1	-0.950991184	2.05028E-06	chr5_148308424_149597425
5930430L01Rik	Ubl3	-0.94718285	2.96146E-06	chr5_148308424_149597425
5930430L01Rik	Wdr95	-0.900549182	6.47014E-05	chr5_148308424_149597425
Gm15408	Katnal1	-0.925966405	1.54539E-05	chr5_148308424_149597425
Gm15408	Medag	0.935364435	7.96598E-06	chr5_148308424_149597425
Gm15408	Wdr95	-0.927132534	1.43031E-05	chr5_148308424_149597425
Gm42788	Ubl3	-0.906094425	4.90337E-05	chr5_148308424_149597425
Gm42788	Hmgb1	0.930978272	1.09788E-05	chr5_148308424_149597425
Gm20186	Lsm8	0.909491761	4.10229E-05	chr6_18050000_18850000
2310069B03Rik	Mrpl19	0.972474928	1.18813E-07	chr6_81930006_83010006
Gm42810	Itfg2	-0.939556573	5.73766E-06	chr6_127449982_128489982
9330102E08Rik	Foxm1	-0.980254192	2.28707E-08	chr6_127449982_128489982
Gm10010	Foxm1	-0.9291718	1.24541E-05	chr6_127449982_128489982
Gm44275	Pbp2	0.963921157	4.53125E-07	chr6_134969982_136411479
Gm44275	Emp1	0.919697025	2.29551E-05	chr6_134969982_136411479
Gm15706	Rassf8	-0.925088657	1.63674E-05	chr6_145011480_146531478
Gm15706	Itpr2	-0.945841634	3.34941E-06	chr6_145011480_146531478
Gm26890	Snrpd2	-0.928946126	1.26489E-05	chr7_19134651_20094651
Gm26890	Opa3	0.96855959	2.29508E-07	chr7_19134651_20094651
Gm26890	Vasp	0.963977997	4.4961E-07	chr7_19134651_20094651
Gm26890	Ppp1r13l	0.908666472	4.28666E-05	chr7_19134651_20094651
Gm26890	Ercc2	0.90189554	6.05798E-05	chr7_19134651_20094651
Gm26890	Mark4	0.928905924	1.26838E-05	chr7_19134651_20094651
Gm26890	Clasrp	0.906672477	4.75905E-05	chr7_19134651_20094651
Gm26890	Nectin2	0.905368412	5.08948E-05	chr7_19134651_20094651
Gm26762	Pdcd2l	0.937149571	6.94619E-06	chr7_34134981_34774981
Gm26762	Kctd15	0.957772814	9.84913E-07	chr7_34134981_34774981
Gm37494	AI987944	0.965820987	3.46854E-07	chr7_39424630_42624630
Gm37494	Zfp976	0.982285431	1.33364E-08	chr7_39424630_42624630
RP23-54G8.4	Ndn	0.908836386	4.24818E-05	chr7_59895114_62455114
RP23-478L20.2	Tnrc6a	0.944444393	3.79531E-06	chr7_122976486_123496486
RP23-478L20.2	Arhgap17	0.96958667	1.94728E-07	chr7_122976486_123496486
RP23-478L20.2	Lcmt1	-0.943275571	4.20333E-06	chr7_122976486_123496486
H19	Tnnt3	0.975464663	6.71998E-08	chr7_142294095_142734095
Gm26860	Sec63	0.98707892	2.77568E-09	chr10_42280194_43560194
Gm26860	Bend3	0.992770085	1.53701E-10	chr10_42280194_43560194
Gm26789	Sirt1	0.946196721	3.243E-06	chr10_63137252_63417252
Gm26789	Dnajc12	-0.968758947	2.22398E-07	chr10_63137252_63417252
Rmst	Nedd1	-0.9174445	2.62584E-05	chr10_91217255_93017255
Cep83os	Ndufa12	0.929282996	1.2359E-05	chr10_93817255_94977366
Gm26853	Cops5	-0.944868689	3.65521E-06	chr1_9569919_10129919
Gm26768	Cops5	-0.928613976	1.29399E-05	chr1_9569919_10129919

LincRNA	Protein_cod	Correlation	Pvalue	TAD_id
RP23-93F3.3	Ppp1r3b	0.949937219	2.27635E-06	chr8_34856946_35496946
Gm26584	Slc7a2	0.941492275	4.89203E-06	chr8_40514646_41434646
Gm26584	Pcm1	0.917652301	2.59388E-05	chr8_40514646_41434646
2010320M18Rik	Ifi30	0.92978478	1.19369E-05	chr8_70676101_72592747
2010320M18Rik	Mast3	-0.956250684	1.17266E-06	chr8_70676101_72592747
2010320M18Rik	Slc5a5	-0.90610962	4.89954E-05	chr8_70676101_72592747
2010320M18Rik	Myo9b	-0.952516581	1.755E-06	chr8_70676101_72592747
2010320M18Rik	Babam1	0.918200043	2.51112E-05	chr8_70676101_72592747
2010320M18Rik	Pgls	0.960530376	7.05959E-07	chr8_70676101_72592747
2010320M18Rik	Zfp709	0.957843075	9.76863E-07	chr8_70676101_72592747
2010320M18Rik	Zfp882	0.935048065	8.15844E-06	chr8_70676101_72592747
2010320M18Rik	Zfp961	0.922822685	1.89235E-05	chr8_70676101_72592747
2010320M18Rik	Fam32a	0.976995592	4.88182E-08	chr8_70676101_72592747
2010320M18Rik	Eps15l1	-0.900931776	6.35084E-05	chr8_70676101_72592747
2010320M18Rik	Med26	-0.90569578	5.00489E-05	chr8_70676101_72592747
RP23-399J5.1	Mast3	0.937364544	6.83072E-06	chr8_70676101_72592747
RP23-399J5.1	Gtpbp3	0.941518491	4.8813E-06	chr8_70676101_72592747
RP23-399J5.1	Slc27a1	0.907693107	4.51241E-05	chr8_70676101_72592747
RP23-399J5.1	Pgls	-0.916390221	2.79277E-05	chr8_70676101_72592747
RP23-399J5.1	Fam129c	0.950957179	2.05728E-06	chr8_70676101_72592747
RP23-399J5.1	Zfp882	-0.959090773	8.4239E-07	chr8_70676101_72592747
RP23-399J5.1	Zfp961	-0.948683702	2.57049E-06	chr8_70676101_72592747
RP23-399J5.1	Fam32a	-0.927823938	1.36535E-05	chr8_70676101_72592747
RP23-399J5.1	Cherp	0.953888699	1.5192E-06	chr8_70676101_72592747
RP23-399J5.1	Med26	0.964225017	4.34585E-07	chr8_70676101_72592747
Gm26532	Adgrl1	-0.914498854	3.11312E-05	chr8_83636101_84636101
Gm26532	Cacna1a	-0.939837018	5.60847E-06	chr8_83636101_84636101
A330074K22Rik	Gins2	0.950069034	2.24704E-06	chr8_119396100_120636100
RP23-115O21.3	Fanca	0.983697618	8.82407E-09	chr8_121836100_123556100
Gm26772	Zmiz1	0.939628916	5.70411E-06	chr14_24980514_25860514
Gm26772	Ppif	0.924672352	1.68152E-05	chr14_24980514_25860514
Gm10101	Wdhd1	0.937652791	6.67827E-06	chr14_46820325_47260325
Gm26782	Tep1	-0.926331163	1.50864E-05	chr14_50820325_51860325
Gm26782	Osgep	0.961231213	6.46236E-07	chr14_50820325_51860325
Gm26782	Rnase4	0.922692691	1.90791E-05	chr14_50820325_51860325
Gm26782	Ang	0.949729482	2.32315E-06	chr14_50820325_51860325
Gm16617	Arhgef40	0.917244496	2.65689E-05	chr14_51860325_53700325
Gm16617	Zfp219	0.964255192	4.32778E-07	chr14_51860325_53700325
Gm16617	Chd8	0.977885734	4.01364E-08	chr14_51860325_53700325
Gm16973	Ipo4	0.950478906	2.15782E-06	chr14_55621163_57101163
Gm16973	Nedd8	-0.966306363	3.2318E-07	chr14_55621163_57101163
Gm16973	Nop9	0.969089336	2.11002E-07	chr14_55621163_57101163
Gm16973	Adcy4	0.94261881	4.44738E-06	chr14_55621163_57101163
Gm16973	Ripk3	-0.941450837	4.90903E-06	chr14_55621163_57101163
Gm16973	Khnyn	0.972001889	1.29277E-07	chr14_55621163_57101163
Gm16973	Parp4	0.970009241	1.817E-07	chr14_55621163_57101163
4930480K23Rik	Bin3	0.926047232	1.53719E-05	chr14_69560193_70160193
C030014I23Rik	Kmt2a	-0.935571895	7.84174E-06	chr9_44431917_45151917
B930082K07Rik	Sgk3	0.942978523	4.31236E-06	chr1_9569919_10129919
B930082K07Rik	Cops5	0.917475281	2.62109E-05	chr1_9569919_10129919
Gm17477	Cops5	-0.916432569	2.78591E-05	chr1_9569919_10129919
9430037G07Rik	Snx14	0.923672846	1.79301E-05	chr9_87425165_88905162
4833445I07Rik	Sgk3	0.909503206	4.09978E-05	chr1_9569919_10129919
Gm26797	Entpd3	0.913939464	3.2132E-05	chr9_120130882_124161429
Gm26563	Cops5	-0.916349979	2.7993E-05	chr1_9569919_10129919
2010300F17Rik	Kat7	-0.938809473	6.09336E-06	chr11_94618686_97618686
2010300F17Rik	Spop	0.935206769	8.06144E-06	chr11_94618686_97618686
2010300F17Rik	B4galnt2	-0.960508654	7.07878E-07	chr11_94618686_97618686
2010300F17Rik	Copz2	0.946862165	3.0508E-06	chr11_94618686_97618686
2810433D01Rik	Gpatch8	-0.961120497	6.55393E-07	chr11_102458686_103098686
2810433D01Rik	Gm11627	0.950237325	2.21006E-06	chr11_102458686_103098686
2810433D01Rik	Gjc1	-0.937903692	6.54777E-06	chr11_102458686_103098686
2810433D01Rik	Plcd3	-0.902953179	5.74891E-05	chr11_102458686_103098686
Snhg20	St6galnac2	-0.91346794	3.29951E-05	chr11_116618686_117818686
Snhg20	Tk1	-0.923023487	1.8685E-05	chr11_116618686_117818686
Gpr137b-ps	Ero1lb	0.968095358	2.46766E-07	chr13_12467733_13427733
Gm26514	Slc35b3	-0.946384396	3.18785E-06	chr13_37988131_40784631
5033403F01Rik	Bloc1s5	0.908798052	4.25683E-05	chr13_37988131_40784631
5033430I15Rik	Atxn1	0.956177268	1.18239E-06	chr13_45544631_46664631
5033430I15Rik	Rbm24	0.905778096	4.98379E-05	chr13_45544631_46664631

LincRNA	Protein_cod	Correlation	Pvalue	TAD_id
6720427I07Rik	Phf2	-0.954510352	1.42102E-06	chr13_48504631_49744631
6720427I07Rik	Fam120a	-0.938896391	6.05111E-06	chr13_48504631_49744631
6720427I07Rik	Wnk2	-0.956611979	1.12573E-06	chr13_48504631_49744631
6720427I07Rik	Fgd3	-0.91502641	3.021E-05	chr13_48504631_49744631
6720427I07Rik	Bicd2	-0.950784618	2.09312E-06	chr13_48504631_49744631
Fam120aos	Phf2	-0.9853245	5.23074E-09	chr13_48504631_49744631
Fam120aos	Fam120a	-0.968113313	2.4608E-07	chr13_48504631_49744631
Fam120aos	Wnk2	-0.973029616	1.07415E-07	chr13_48504631_49744631
Fam120aos	Card19	-0.943312748	4.18984E-06	chr13_48504631_49744631
Fam120aos	Fgd3	-0.949082866	2.47374E-06	chr13_48504631_49744631
Fam120aos	Bicd2	-0.992028603	2.50117E-10	chr13_48504631_49744631
Fam120aos	Iars	-0.951758978	1.8971E-06	chr13_48504631_49744631
Gm26651	Sema4d	-0.959817395	7.71136E-07	chr13_51664631_52224631
Gm26819	Ice1	0.991658688	3.13599E-10	chr13_69661122_70621122
2310015A10Rik	Zfyve26	-0.944799355	3.67782E-06	chr12_79179013_80139013
Gm28933	Ptpn21	0.995882881	9.25199E-12	chr12_98641790_100121790
Gm28933	Eml5	0.98816687	1.79128E-09	chr12_98641790_100121790
Gm28933	Foxn3	0.998034811	2.30065E-13	chr12_98641790_100121790
4930478K11Rik	Hhipl1	0.985317193	5.24371E-09	chr12_108281790_108521790
Meg3	Evl	0.900965816	6.34031E-05	chr12_108521790_110401790
Meg3	Dio3	-0.920626386	2.16918E-05	chr12_108521790_110401790
Gm10425	Rcor1	-0.910213199	3.94629E-05	chr12_110881790_113241789
Gm10425	Cdc42bpb	-0.900132954	6.60191E-05	chr12_110881790_113241789
Gm10425	Gpr132	-0.906303832	4.8507E-05	chr12_110881790_113241789
Gm10425	Mta1	-0.906930372	4.69575E-05	chr12_110881790_113241789
2810029C07Rik	Rcor1	-0.975386335	6.82703E-08	chr12_110881790_113241789
2810029C07Rik	Cdc42bpb	-0.959584294	7.93451E-07	chr12_110881790_113241789
2810029C07Rik	Eif5	-0.960920997	6.72155E-07	chr12_110881790_113241789
2810029C07Rik	Mark3	-0.974430365	8.24659E-08	chr12_110881790_113241789
2810029C07Rik	Zfyve21	0.945440897	3.4728E-06	chr12_110881790_113241789
2810029C07Rik	Ppp1r13b	-0.977765641	4.12298E-08	chr12_110881790_113241789
2810029C07Rik	2010107E04Rik	0.948632584	2.58309E-06	chr12_110881790_113241789
2810029C07Rik	Zbtb42	-0.971101096	1.51225E-07	chr12_110881790_113241789
2810029C07Rik	Cep170b	-0.963761857	4.63092E-07	chr12_110881790_113241789
2810029C07Rik	Ahnak2	-0.974885167	7.54459E-08	chr12_110881790_113241789
2810029C07Rik	BC022687	-0.902952474	5.74911E-05	chr12_110881790_113241789
2810029C07Rik	Gpr132	-0.969375535	2.0151E-07	chr12_110881790_113241789
2810029C07Rik	Pacs2	-0.970703652	1.61806E-07	chr12_110881790_113241789
2810029C07Rik	Mta1	-0.96305805	5.09238E-07	chr12_110881790_113241789
Gm17638	Myh9	0.914125464	3.17964E-05	chr15_77449570_77969570
Gm16576	Pick1	0.966567071	3.11005E-07	chr15_79209570_80089570
Gm16576	Maff	0.977164025	4.70703E-08	chr15_79209570_80089570
Gm16576	Csnk1e	0.993092357	1.22432E-10	chr15_79209570_80089570
Gm16576	Kcnj4	0.987713595	2.16003E-09	chr15_79209570_80089570
Gm16576	Kdelr3	-0.975703827	6.40135E-08	chr15_79209570_80089570
Gm16576	Ddx17	0.970914367	1.56125E-07	chr15_79209570_80089570
Gm16576	Josd1	0.994830874	2.88132E-11	chr15_79209570_80089570
Gm16576	Gtpbp1	0.947721164	2.81617E-06	chr15_79209570_80089570
Gm16576	Sun2	0.990537172	5.88157E-10	chr15_79209570_80089570
Gm16576	Cbx7	0.904021175	5.44952E-05	chr15_79209570_80089570
Gm26518	Zfp740	0.970300053	1.73145E-07	chr15_102089569_102689569
Gm26518	Rarg	0.90593587	4.94355E-05	chr15_102089569_102689569
Gm26518	Mfsd5	0.938058238	6.4684E-06	chr15_102089569_102689569
Gm26518	Sp1	0.97662301	5.28685E-08	chr15_102089569_102689569
Gm26518	Tarbp2	0.918826512	2.41903E-05	chr15_102089569_102689569
Gm26518	Atf7	0.981167692	1.80754E-08	chr15_102089569_102689569
Gm16861	Ubald1	0.93479082	8.31763E-06	chr16_4680000_5239907
Gm16861	Ubn1	0.928989583	1.26112E-05	chr16_4680000_5239907
Gm16861	Nagpa	0.947023098	3.0057E-06	chr16_4680000_5239907
Gm4262	Litaf	0.932046887	1.01734E-05	chr16_10559907_13079907
2610020C07Rik	Txndc11	0.918436842	2.47599E-05	chr16_10559907_13079907
2610020C07Rik	Cpped1	-0.914889538	3.04469E-05	chr16_10559907_13079907
1300002E11Rik	Igf2bp2	-0.920238189	2.22126E-05	chr16_21479927_22239927
Gm26838	Sec22a	-0.969064891	2.11828E-07	chr16_35079914_35839914
3300005D01Rik	Serac1	-0.91716226	2.66974E-05	chr17_5400000_6440852
Gm26873	Lnpep	0.927237065	1.42034E-05	chr17_17263036_17663036
9330136K24Rik	Zfp677	0.951206028	2.00646E-06	chr17_21183036_21623033
9330136K24Rik	Zfp51	0.935434409	7.9239E-06	chr17_21183036_21623033
9330136K24Rik	Zfp53	0.940722353	5.21564E-06	chr17_21183036_21623033
Gm16386	Zfp947	0.916167093	2.82915E-05	chr17_22063033_23743033

LincRNA	Protein_cod	Correlation	Pvalue	TAD_id
Gm16386	Zfp944	0.903459968	5.60528E-05	chr17_22063033_23743033
Gm16386	Zfp758	0.91974591	2.28872E-05	chr17_22063033_23743033
Gm16386	Zfp946	0.931084198	1.08967E-05	chr17_22063033_23743033
9530082P21Rik	Sgk3	-0.905698442	5.00421E-05	chr1_9569919_10129919
9530082P21Rik	Cops5	-0.912812153	3.42258E-05	chr1_9569919_10129919
Gm26724	Ip6k3	0.98310084	1.05513E-08	chr17_27143055_27543055
Gm26724	Lemd2	0.988347823	1.65895E-09	chr17_27143055_27543055
Gm26785	Xpo5	0.977064992	4.80918E-08	chr17_46183051_46743051
Gm26785	Yipf3	0.971050308	1.52546E-07	chr17_46183051_46743051
Gm26785	Tjap1	0.972385213	1.20744E-07	chr17_46183051_46743051
Gm26785	Zfp318	0.978245012	3.70026E-08	chr17_46183051_46743051
Gm26785	Srf	0.989449594	1.01145E-09	chr17_46183051_46743051
Gm26785	Ptk7	0.989211906	1.13018E-09	chr17_46183051_46743051
Gm26785	Klc4	0.967517568	2.69672E-07	chr17_46183051_46743051
Gm26785	Cnpy3	0.940389785	5.36056E-06	chr17_46183051_46743051
Gm26749	Spast	0.964456577	4.20866E-07	chr17_74320660_75560660
Gm26749	Slc30a6	-0.915993913	2.85764E-05	chr17_74320660_75560660
Gm26749	Birc6	0.944699478	3.71058E-06	chr17_74320660_75560660
4833418N02Rik	Pigf	0.941634929	4.83386E-06	chr17_86480660_87280660
4833418N02Rik	Mcfd2	0.921722646	2.02727E-05	chr17_86480660_87280660
Gm26682	Zeb1	0.903455641	5.60649E-05	chr18_4200002_6600002
2010110K18Rik	Kif20a	0.974238058	8.55863E-08	chr18_33800346_34800346
2010110K18Rik	Cdc25c	0.983283095	9.99749E-09	chr18_33800346_34800346
2010110K18Rik	Gm3550	0.922375936	1.94626E-05	chr18_33800346_34800346
3222401L13Rik	Pcdhb11	0.944376644	3.81806E-06	chr18_37280346_37960346
3222401L13Rik	Pcdhb14	0.949512612	2.37283E-06	chr18_37280346_37960346
3222401L13Rik	Pcdhb19	0.966332963	3.2192E-07	chr18_37280346_37960346
3222401L13Rik	Pcdhb21	0.936964258	7.04697E-06	chr18_37280346_37960346
3222401L13Rik	Pcdhb22	0.924988599	1.64742E-05	chr18_37280346_37960346
3222401L13Rik	Pcdhga1	-0.943300898	4.19414E-06	chr18_37280346_37960346
3222401L13Rik	Pcdhga2	-0.907294964	4.6074E-05	chr18_37280346_37960346
3222401L13Rik	Pcdhga3	-0.911816304	3.61634E-05	chr18_37280346_37960346
3222401L13Rik	Pcdhga5	-0.901597893	6.14727E-05	chr18_37280346_37960346
3222401L13Rik	Pcdhgb4	-0.960852723	6.77968E-07	chr18_37280346_37960346
3222401L13Rik	Pcdhga8	-0.967304253	2.78543E-07	chr18_37280346_37960346
3222401L13Rik	Pcdhgb5	-0.945972359	3.30991E-06	chr18_37280346_37960346
3222401L13Rik	Pcdhgb6	-0.903598188	5.56659E-05	chr18_37280346_37960346
3222401L13Rik	Pcdhga11	-0.907167075	4.63824E-05	chr18_37280346_37960346
3222401L13Rik	Pcdhgc3	-0.924403856	1.71092E-05	chr18_37280346_37960346
3222401L13Rik	Diaph1	-0.912489623	3.48441E-05	chr18_37280346_37960346
3222401L13Rik	Fchsd1	-0.955259051	1.30952E-06	chr18_37280346_37960346
Carmn	Cd74	0.986289628	3.72861E-09	chr18_60800346_61720346
Carmn	Tcof1	0.991615103	3.21855E-10	chr18_60800346_61720346
Carmn	Pdgfrb	0.936256189	7.44275E-06	chr18_60800346_61720346
Carmn	Hmgxb3	0.991474799	3.49617E-10	chr18_60800346_61720346
Carmn	Ppargc1b	0.987335139	2.51225E-09	chr18_60800346_61720346
Malat1	Cfl1	-0.919103819	2.37913E-05	chr19_5480000_6400000
Malat1	Kat5	0.943619041	4.07999E-06	chr19_5480000_6400000
Malat1	Rela	0.950146153	2.23004E-06	chr19_5480000_6400000
Malat1	Pcnx3	0.925440519	1.59962E-05	chr19_5480000_6400000
Malat1	Ehbp1l1	0.937137173	6.9529E-06	chr19_5480000_6400000
Malat1	Ltbp3	0.931298694	1.07322E-05	chr19_5480000_6400000
Malat1	Scyl1	0.940404144	5.35424E-06	chr19_5480000_6400000
Malat1	Dpf2	0.909921304	4.00884E-05	chr19_5480000_6400000
Malat1	Syvn1	0.9519198	1.8662E-06	chr19_5480000_6400000
Malat1	Vps51	0.90300446	5.73425E-05	chr19_5480000_6400000
Malat1	Sf1	0.944839342	3.66477E-06	chr19_5480000_6400000
Neat1	Cfl1	-0.971428829	1.42921E-07	chr19_5480000_6400000
Neat1	Kat5	0.95174895	1.89904E-06	chr19_5480000_6400000
Neat1	Rela	0.933558678	9.11449E-06	chr19_5480000_6400000
Neat1	Pcnx3	0.931258493	1.07629E-05	chr19_5480000_6400000
Neat1	Ehbp1l1	0.935750603	7.73594E-06	chr19_5480000_6400000
Neat1	Frmd8	0.938660348	6.16641E-06	chr19_5480000_6400000
Neat1	Dpf2	0.922935322	1.87894E-05	chr19_5480000_6400000
Neat1	Syvn1	0.935728534	7.74895E-06	chr19_5480000_6400000
Neat1	Vps51	0.900787926	6.39549E-05	chr19_5480000_6400000
Neat1	Zfpl1	-0.914733727	3.07183E-05	chr19_5480000_6400000
Neat1	Sf1	0.907111996	4.65157E-05	chr19_5480000_6400000
Gm26792	Nfkb2	0.941433956	4.91597E-06	chr19_46045510_46525510
Gm26792	Cuedc2	-0.910880023	3.80629E-05	chr19_46045510_46525510

LincRNA	Protein_cod	Correlation	Pvalue	TAD_id
Gm26792	Trim8	0.930973727	1.09823E-05	chr19_46045510_46525510

Publications:

- *Differential Analysis of Neurodegenerative Aging-Related Mitochondrial Genes of Long-Lived Naked Mole-Rat.* L.A. Khayal, I. Provaznik and E. Tkacz. International Journal of Bioscience, Biochemistry and Bioinformatics **(IJBBB)**, Vol. 3(2), pp. 75 – 79, 19th February **2013**, ISSN 2010-3638. Presented in 3rd International Conference on Bioscience, Biochemistry and Bioinformatics ICBBB 2013, 25th February 2013, Rome Italy.
- *Improvement of Electrophoresis Performance by Spectral Analysis*. L.A. Khayal, R. Kizek and I. Provaznik. African Journal of Biotechnology **(AJB)**, Vol. 11(51), pp. 11329-11332, 26th June **2012**, ISSN 1684–5315.
- *Using Spectral Analysis in Electrophoresis of Nucleic Acids.* L.A. Khayal and I. Provaznik. The 5th International Symposium on Bio- and Medical Informatics and Cybernetics: **(BMIC -WMSCI 2011)**, ISBN-978-1-936338-30-6, 20th July 2011, Orlando, Florida, USA.
- *The progeroid disorder gerodermia osteodysplastica is caused by defective decorin glycanation leading to extracellular matrix disorganization and elevated TGF-β signaling.* W.L. Chan, M. Steiner, L.A. Khayal, S. Mundlos, and U. Kornak. **PLoS Genetics** (Under Review)
- *Transcriptional Profiling of Murine Osteoblast Differentiation based on RNA-seq Expression Analyses.* L.A. Khayal, J. Grünhagen, Ivo Provazník, S. Mundlos, U. Kornak, P.N. Robionson, C.E. Ott. **BONE** (Under Review)