

University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

5-2015

An island-based approach for RNA-SEQ differential expression analysis.

Abdallah Eteleeb
University of Louisville

Follow this and additional works at: <https://ir.library.louisville.edu/etd>



Part of the [Computer Engineering Commons](#)

Recommended Citation

Eteleeb, Abdallah, "An island-based approach for RNA-SEQ differential expression analysis." (2015).
Electronic Theses and Dissertations. Paper 2072.
<https://doi.org/10.18297/etd/2072>

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

AN ISLAND-BASED APPROACH FOR RNA-SEQ DIFFERENTIAL EXPRESSION ANALYSIS

By

Abdallah Eteleeb

B.S., University of Al-Jabal Al-Gharbi, Libya, 1996

M.S., HAN University of Applied Science, The Netherlands, 2005

A Dissertation

Submitted to the Faculty of the

J.B. Speed School of Engineering of the University of Louisville

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science and Engineering

Department of Computer Engineering and Computer Science

University of Louisville

Louisville, Kentucky

May 2015

© Copyright 2015 by Abdallah Eteleeb

All Rights Reserved

AN ISLAND-BASED APPROACH FOR RNA-SEQ DIFFERENTIAL EXPRESSION ANALYSIS

By

Abdallah Eteleeb

B.S., University of Al-Jabal Al-Gharbi, Libya, 1996

M.S., HAN University of Applied Science, The Netherlands, 2005

A Dissertation Approved On

April 6, 2015

Date

by the following Dissertation Committee:

Eric C. Rouchka, D.Sc. Advisor

Dar-jen Chang, Ph.D.

Hunter N. Moseley, Ph.D.

Jeffrey C. Petruska, Ph.D.

Roman V. Yampolskiy, Ph.D.

DEDICATION

This dissertation is dedicated to
my family
for their love, endless support, and encouragement

ACKNOWLEDGEMENTS

I would never been able to finish my dissertation without the guidance, help, support, and encouragement from several people. It is to them that I owe my deepest gratitude.

First and foremost, I would like to express my deepest gratitude to my advisor Dr. Eric Rouchka, for his inspirational guidance, continuous support, and patience throughout my Ph.D. journey. His knowledge and expertise made it possible for me to dive in the world of life science and made bioinformatics an interesting realm for me. He has guided me and helped me in so many different directions including dissertation research, paper writing, research talks, poster presentations, conferences and workshops attendance, and job documents writing and job interviews preparations. Through his guidance, support, and encouragement, Dr. Rouchka has played a vital role for me to develop my skills and expertise in computational biology and bioinformatics areas and become a good Bioinformatician. I am blessed to have had such a wonderful supervisor-working with him has always been a delightful experience. Without his guidance and persistent help this dissertation would not have been possible.

I am very grateful to Dr. Robert Flight for his support, guidance, and encouragement. His insightful ideas, suggestions, and experience were crucial to my work and helped me to develop the approach my dissertation is based on. He was always supportive and available to provide solutions to every single problem I had in my work.

I am also thankful to our collaborators, Dr. Jeffrey Petruska and Dr. Benjamin Harrison for their support and help through this work. They were the ones who allowed us to use their datasets and answered most of our biological questions. I sincerely hope that we will remain both collaborators and friends for many years to come.

I would especially like to thank Dr. Hunter Moseley for supporting my dissertation work with his clever ideas and suggestions. He contributed interesting discussions and provided excellent solutions to the second part of my dissertation work on the comparison of combined p -value methods. He was always available answering all type of questions whether biological or statistical and computational questions. Working with him has been an extremely delightful experience. I would like also to thank him for his acceptance to be part of my dissertation committee despite the long distance he had to travel.

My thanks to Dr. Dar-jen Chang and Dr. Roman V. Yampolskiy for voluntarily accepting to serve on my dissertation committee. I greatly appreciate their valuable time to review my dissertation and provide suggestions and feedback that help me refine my work. I would also like to thank Dr. Ming Ouyang who served as a committee member during my Ph.D. proposal.

I would like to thank my former and current lab members, Dr. Fahim Mohammad, Dr. Dazhuo Li, and Ernur Saka for providing me with the support, help, and encouragement I need. I really enjoyed working in the lab and enjoyed the interesting conversations we had.

I would like to extend my sincere thanks to the Ministry of Higher Education in Libya and the University of Aljabal Algharbi for the scholarship to undertake the PhD study. Without the financial support they provided, this work would not have been possible.

I am also indebted to Dr. Rouchka, Dr. Nigel Cooper, and KBRIN (Kentucky Biomedical Research Infrastructure Network) for their financial support for the remainder of my Ph.D. study.

Last but not least, I would like to thank my wife and my extended family for their support, patience, and prayers they always provide in all my endeavors.

ABSTRACT

AN ISLAND-BASED APPROACH FOR RNA-SEQ DIFFERENTIAL EXPRESSION ANALYSIS

Abdallah Eteleeb

April 6, 2015

High-throughput mRNA sequencing (also known as RNA-Seq) promises to be the technique of choice for studying transcriptome profiles, offering several advantages over old techniques such as microarrays. This technique provides the ability to develop precise methodologies for a variety of RNA-Seq applications including gene expression quantification, novel transcript and exon discovery, differential expression (DE) and splice variant detection. The detection of significantly changing features (e.g. genes, transcript isoforms, exons) in expression across biological samples is a primary application of RNA-Seq. Uncovering which features are significantly differentially expressed between samples can provide insight into their functions.

One major limitation with the majority of recently developed methods for RNA-Seq differential expression is the dependency on annotated biological features to detect expression differences across samples. This forces the identification of expression levels and the detection of significant changes to known genomic regions. Thus, any significant changes occurring in unannotated regions will not be captured.

To overcome this limitation, we developed a novel segmentation approach, Island-Based (IBSeq), for analyzing differential expression in RNA-Seq and targeted sequencing (exome capture) data without specific knowledge of an isoform. IBSeq

segmentation determines individual islands of expression based on windowed read counts that can be compared across experimental conditions to determine differential island expression. In order to detect differentially expressed features, the significance of DE islands corresponding to each feature are combined using combined p -value methods. We evaluated the performance of our approach by comparing it to a number of existing gene DE methods using several benchmark MAQC RNA-Seq datasets. Using the area under ROC curve (auROC) as a performance metric, results show that IBSeq clearly outperforms all other methods compared.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iv
ABSTRACT	vii
LIST OF TABLES	xiv
LIST OF FIGURES	xvi

CHAPTER

1	INTRODUCTION	1
1.1	Motivation	2
1.2	Dissertation Contributions	4
1.3	Dissertation Outline	5
2	BIOLOGICAL BACKGROUND	8
2.1	Organisms and Cells	8
2.2	Nucleic Acids	9
2.3	Proteins	10
2.4	Central Dogma of Molecular Biology	11
2.5	Genes and Genomes	16
2.6	Gene Expression and Alternative Splicing	17
2.6.1	Gene Expression	17
2.6.2	Alternative Splicing	18
3	NEXT-GENERATION SEQUENCING	24
3.1	First-Generation Sequencing Methods	24

3.2	Next-Generation Sequencing Technologies	27
3.2.1	Roche/454 GS FLX Sequencer	28
3.2.2	Illumina/Solexa Genome Analyzer	30
3.2.3	Applied Biosystems SOLiD Sequencer	31
3.2.4	Polonator G.007	32
3.2.5	Helicos Heliscope	33
3.2.6	Pacific BioSciences RS	33
3.2.7	Ion Torrent	34
3.3	Applications of Next-Generation Sequencing Technologies . . .	34
3.4	Next-Generation Sequencing Data	34
3.4.1	Raw Data (Short Reads) Format	35
3.4.1.1	FASTA Format	36
3.4.1.2	FASTQ Format	36
3.4.2	Sequence Alignment Format	39
3.4.2.1	Sequence Alignment/Map (SAM) Format	39
3.5	Genome Assembly and Alignment	40
3.5.1	Alignment	42
3.5.1.1	Hash-Based Approach	43
3.5.1.2	Burrows-Wheeler Transform (BWT) Approach . .	44
3.5.1.3	Bowtie	47
3.5.2	Assembly	48
3.5.2.1	Graphs and Graph Theory	49
3.5.2.2	Greedy Graph Assembly	50
3.5.2.3	Overlap/Layout/Consensus (OLC)	50
3.5.2.4	De Bruijn Graph (DBG)	51

4	RNA-SEQ METHODOLOGIES	53
4.1	Transcriptome Analysis	53
4.2	Transcriptome Analysis Techniques	54
4.2.1	Candidate Gene Approaches	54
4.2.1.1	Northern Blot	54
4.2.1.2	Reverse Transcription Quantitative PCR (RT-qPCR)	54
4.2.2	Sequencing-Based Approaches	56
4.2.2.1	Expressed Sequencing Tags (ESTs)	56
4.2.2.2	Serial Analysis of Gene Expression (SAGE)	57
4.2.2.3	Massively Parallel Signature Sequencing (MPSS) .	58
4.2.3	Microarray Technology	58
4.3	High-Throughput mRNA Sequencing (RNA-Seq)	61
4.3.1	RNA-Seq Workflow	63
4.3.2	RNA-Seq Applications	64
4.3.2.1	Transcript Assembly	64
4.3.2.2	Transcript Quantification	68
4.3.2.3	Differential Expression (DE) Analysis	72
4.3.2.4	Normalization	73
4.3.2.5	Statistical Modeling of Gene Expression	77
4.3.2.6	Testing for Differential Expression	78
4.3.2.7	Differential Expression Analysis Methods	79
5	IBSEQ: AN ISLAND-BASED APPROACH FOR RNA-SEQ DIFFERENTIAL EXPRESSION ANALYSIS	84
5.1	Introduction	84
5.2	IBSeq Overview	85
5.2.1	IBSeq Framework Steps	87

5.2.1.1	Compute Per Base Abundance	87
5.2.1.2	Genome Partition and Region Classification	88
5.2.1.3	Island Construction	88
5.2.2	Island Differential Expression Testing	89
5.2.3	Combined Significance of DE Islands	91
5.2.4	IBSeq Algorithm	91
5.3	Experimental Results	92
5.3.1	Datasets	95
5.3.1.1	MAQC Datasets	95
5.3.1.2	qRT-PCR Datasets	95
5.3.2	Evaluation of IBSeq Approach for Detecting DEGs . . .	96
5.3.3	Construct Islands and Test for DE Islands	97
5.3.4	Combined Significance of DE Islands	98
5.3.5	Evaluation and Comparison	99
5.4	Conclusion	106
6	COMBINING THE SIGNIFICANCE OF GENOMIC REGIONS - A COMPARATIVE STUDY	107
6.1	Introduction	107
6.2	Methods	109
6.2.1	Combined P-value Methods	109
6.2.1.1	Fisher's Method	109
6.2.1.2	Z-transform Method	110
6.2.1.3	Weighted z -test	110
6.2.1.4	Minimum p -value Method	111
6.2.1.5	Logit Method	111
6.2.1.6	Weighted-sum Method	112

6.2.2	Datasets	112
6.2.2.1	MAQC Datasets	113
6.2.2.2	Marioni’s Liver and Kidney Dataset	113
6.2.3	Differential Expression	114
6.3	Results	115
6.3.1	Results from Liver and Kidney	115
6.3.2	Results from MAQC Datasets	117
6.4	Conclusion	122
7	DISCUSSION AND CONCLUSIONS	123
7.1	Parameters-Determination Analysis	124
7.1.1	Window Size Analysis	124
7.1.2	Classification Threshold <i>P</i> -value Analysis	125
7.1.3	Gap Size Analysis	127
7.2	Further Island Segmentation	128
7.3	Computational and Space Complexity	134
7.4	Future Directions	139
7.4.1	Potential IBSeq Extensions	140
7.4.2	Combining the Significance of Islands	140
7.4.3	Comparison to Transcriptome Assemblers	141
7.4.4	IBSeq Optimization	141
	REFERENCES	143
	APPENDIX A: LIST OF ABBREVIATIONS	164
	CURRICULUM VITAE	166

LIST OF TABLES

TABLE	Page
3.1 Detailed information for current NGS platforms	29
3.2 Key applications of NGS technologies	30
3.3 Applications of NGS technologies	35
3.4 Quality scores and base calling accuracy	39
3.5 Alignment section fields in SAM format	41
3.6 The FLAG field in SAM format	41
3.7 Examples of hash-based aligners	44
4.1 Advantages of RNA-Seq over microarray technology	62
4.2 List of common differential expression Analysis methods	74
5.1 Examples of current differential expression analysis methods	85
5.2 Combined p -value methods implemented in IBSeq	92
5.3 Detailed information of constructed islands	98
5.4 Number of true DE and true non-DE genes found by each method using p -value ≤ 0.05	101
5.5 Number of shared true DE genes detected by each method using p - value ≤ 0.05	104
6.1 Summary of RNA-Seq datasets used in this study	112
6.2 Differentially expressed genes detected by each method using p -value < 0.001	116
6.3 Differentially spliced genes for each method	117
6.4 AUC for each method on the four MAQC datasets	121

7.1	Description of datasets used in the complexity analysis	137
7.2	The amount of time and space recorded for each step in IBSeq using four RNA-Seq datasets	138

LIST OF FIGURES

FIGURE	Page
1.1 RNA-Seq workflow for differential expression analysis	3
1.2 Illustration of regions missed by current annotations	4
2.1 Typical animal, bacteria, and plant cells	9
2.2 The chemical structure of nucleotides	10
2.3 DNA double helix	10
2.4 The chemical structure of amino acids and how they are linked	11
2.5 The Central Dogma of Molecular Biology	12
2.6 The journey of genetic information from DNA to protein	13
2.7 The DNA replication	13
2.8 Example of DNA replication	14
2.9 Protein synthesis process	14
2.10 The genetic code	15
2.11 The structure of chromosomes	17
2.12 Alternative splicing	19
2.13 Sequences involved in the RNA splicing	20
2.14 Alternative splicing mechanism	20
2.15 Alternative splicing events	21
2.16 Alternative exon events	22
3.1 The workflow of Sanger sequencing method	25
3.2 Maxam-Gilbert method for DNA sequencing	26
3.3 Description of Shotgun sequencing method	27

3.4	Possible dinucleotides encoded by each color	32
3.5	SRA database growth	36
3.6	Example of FASTA format	37
3.7	Example of FASTQ format	38
3.8	Example of a SAM File	40
3.9	The concept of sequence alignment	42
3.10	The hash-based approach	43
3.11	Genome alignment techniques	45
3.12	Burrows-Wheeler transform process	46
3.13	Burrows-Wheeler transform for genomic sequence data	46
3.14	Overview of genome assembly	48
3.15	A k-mer graph representation of a read sequence with $k=4$	50
3.16	Example of de Bruijn graph assembly with $k=4$	52
4.1	Developmental milestones of transcriptome analysis	54
4.2	The Northern-blot procedure	55
4.3	Overview of EST construction	56
4.4	Overview of SAGE method	57
4.5	Microarray technology	60
4.6	Workflow of RNA-Seq experiment	63
4.7	Overview of reference-based assembly method	65
4.8	Overview of <i>de novo</i> transcript assembly	67
4.9	The combined method for transcriptome assembly	69
5.1	Workflow of the island-based approach	86
5.2	The input and output of IBSeq steps	87
5.3	Illustration of pre-island definition	89
5.4	Illustration of overlapping islands between samples	90

5.5	The flowchart of IBSeq algorithm	93
5.6	The distribution of p -values for the four methods	101
5.7	Number of DE and non-DE genes detected by each method using p - value ≤ 0.05	102
5.8	The ROC curves for the four methods using MAQC datasets	103
5.9	Overlap between true DE genes found by each method	105
5.10	Overlap between true non-DE genes found by each method	105
6.1	Example of combining p -values from multiple genome regions	107
6.2	Overlap between the number of genes detected by each combined p - value method and Marioni's and Cuffdiff results	118
6.3	Number of true DE and true non-DE genes detected by each method for the four MAQC datasets using p -value < 0.05	119
6.4	ROC curves for the six combined p -value methods on the MAQC datasets	121
7.1	ROC curves for different window sizes	125
7.2	True DE and true non-DE detections using different p -values	126
7.3	True DE and true non-DE detections using different gap sizes	127
7.4	Further segmentation algorithm flowchart	129
7.5	Further island segmentation example ($w = 30$ bp)	131
7.6	The performance of further island segmentation algorithm	131
7.7	Improvement of detecting the true non-DE for Fisher's method using median percentile approach	132
7.8	The amount of time and space utilized by each step in IBSeq using four RNA-Seq datasets	138

CHAPTER 1

INTRODUCTION

Over the past decade, next-generation sequencing (NGS) technologies have developed rapidly, revolutionizing genome research and changing the landscape of genetic studies. They have afforded researchers the ability to sequence known and unknown mRNA transcripts that can be either coding or non-coding using RNA-Seq and captureSeq methodologies. Using the captureSeq approach, Mercer *et al.* [104] were able to expand by 12% the number of exonic structures that did not belong to known models. This indicates the power of next-generation sequencing approaches in providing novel information about the complexity of transcripts. Others have used RNA-Seq to expand the knowledge of transcribed regions [57, 149], including long noncoding RNAs (lncRNAs) and microRNAs (miRNAs) [164, 165]. Perhaps the most well-known of these studies was performed by the ENCODE Consortium [37], which focused on understanding encoded elements within the human genome. The GENCODE group relied heavily on RNA-Seq data to improve the accuracy of protein-coding regions, pseudogenes, and noncoding regions in the human genome [60, 65, 61].

The advent of RNA-Seq has enabled researchers and scientists to study the transcriptome at an unprecedented rate and has lately become the standard technology for transcriptome analysis. It is based on the direct sequencing of complementary DNA (cDNA) [109]. An RNA-Seq experiment starts with the extraction of total RNA or a portion such as polyadenylated RNA [159]. The extracted RNA is then

converted to a library of double stranded cDNA and sheared into small fragments. In the next step, adapters are attached to one or both sides of each cDNA fragment. Using next-generation sequencing platforms, each cDNA fragment is sequenced and a short sequence (read) from one end of the fragment (single-end tag) or from both ends (paired-end tag) is obtained. The obtained reads are mapped to the reference genome or transcriptome to measure the abundance of each transcript.

1.1 Motivation

With the massive and complex datasets generated by next-generation sequencing techniques, there has been a significant effort during the last few years to develop computational methods to draw meaningful findings from this data. As a result of this effort, several methods have been developed to model RNA-Seq data and detect for differential expression across biological samples. The majority of these methods are based on parametric assumptions where discrete probability distributions such as binomial, Poisson and negative binomial are used. For differential expression analysis, most RNA-Seq approaches follow a similar workflow (Figure 1.1) where mapped reads are summarized according to known biological features such as exons, transcripts, or genes which restricts the mapping of read sequences to existing annotations. Thus, reads that map to regions outside annotated features will not be captured even in well annotated genomes (e.g. human and mouse) [116] and consequently changes in those regions will be missed. Additionally, previously undetected cassette-based isoforms will be ignored and summarized accordingly to known isoform annotations. While using known annotations allows for insightful analysis of how gene expression changes in differing conditions, it also is limiting in understanding how the gene structure itself might also change.

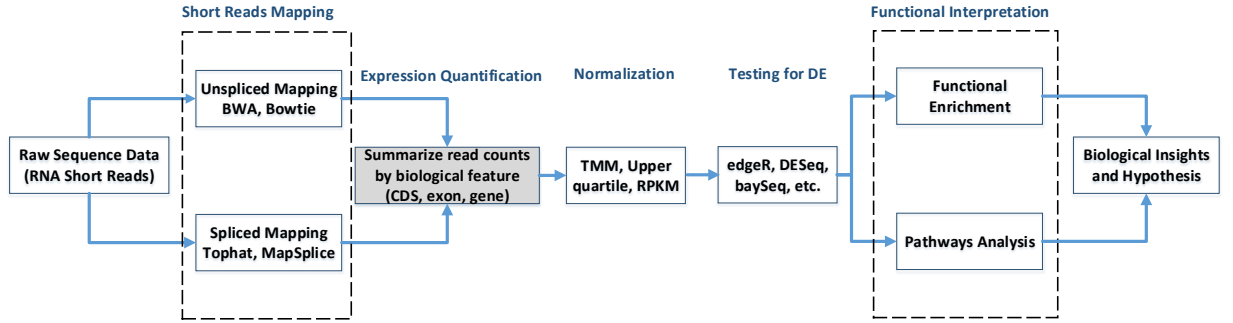


Figure 1.1: RNA-Seq workflow for differential expression analysis. Figure adapted from [97].

To illustrate this problem, Pickrell *et al.* [121] found about 15% of mapped reads were located outside annotated exons in their Nigerian HapMap samples. Alicia *et al.* [116] showed an example of transcripts that fall outside annotated exons for the RNA binding protein 39 gene in LNCaP prostate cancer cells. Our own work, highlighted in Figure 1.2, shows an expression level for microarray data that shows differential expression outside of a known rat gene. This differential transcription would be ignored by current analysis methods, even though it has been experimentally determined that this region is part of the upstream gene. Looking at the annotated mouse homology, it can be inferred that the 3' UTR extends into this region, even if there is no support from the current rat annotation. Further analysis of this differentially expressed transcript shows an association with axonal localization [59].

Furthermore, when detecting for gene DE, most current gene DE methods summarize read counts on the gene level. However, given the fact that most genes consist of multiple exons and the distribution of read counts in exons for a single gene can be different [158], this may provide inaccurate results. Thus, if genes are broken down into smaller regions, such as exons or even smaller fragments, and DE analysis is performed on those regions, the significance of the overall region can be determined using combined p -values which may improve the accuracy of detecting

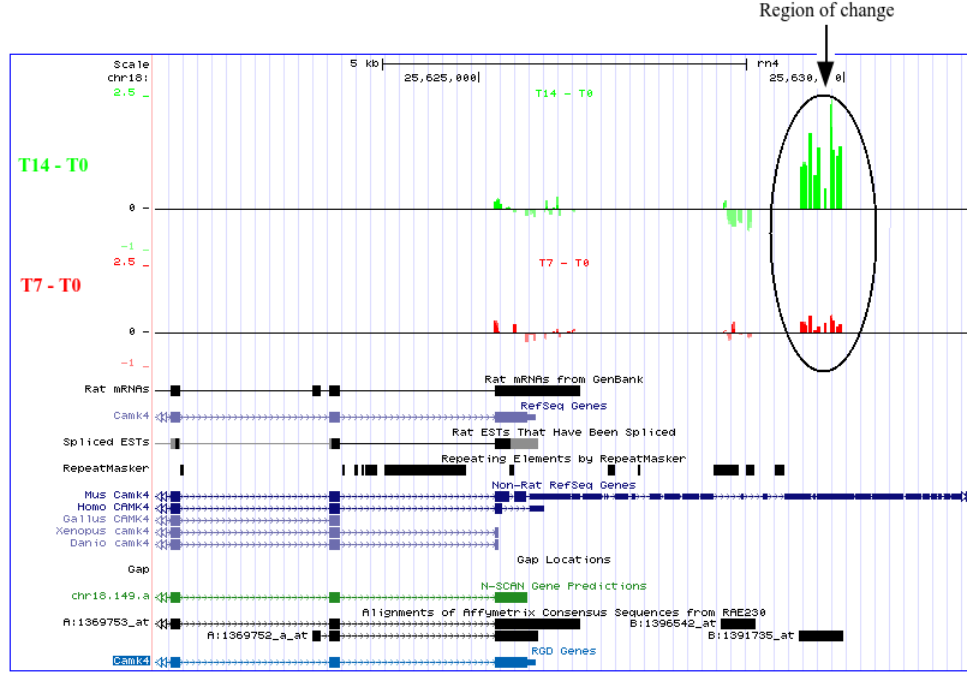


Figure 1.2: Illustration of regions missed by current annotations. The 3' UTR region has a significant change between samples T14 vs T0 and samples T7 vs T0. The annotated mouse CaMK4 gene extends into this region. However, the corresponding rat CaMK4 gene annotation terminates prior to the differentially expressed region, which was subsequently verified to be part of the rat CaMK4 gene [59].

DE genes. Therefore, each region in the overall gene region will participate in the computation of the overall gene significance based on its degree of importance which ensures that regions in the gene are not treated equally.

1.2 Dissertation Contributions

In order to alleviate the issues resulting from dependence on annotations, a novel Island-Based (IBSeq) approach is developed for RNA-Seq differential expression analysis. In this approach (detailed in Chapter 5), the genome is split into small fixed non-overlapping regions (windows). Those regions are then classified based on their read count densities into high and low density regions. In the next step,

adjacent regions with similar densities are merged together constructing larger regions called *islands*. First, a per base count is computed for each sample using BEDTools [125]. To construct regions, per-base read counts are summarized over a small fixed window (10-50bp) to minimize small variance in coverage due to noise. Once regions are constructed for the entire genome, regions are classified as high or low density regions based on an average threshold calculated for each sample. For each sample, high density islands are constructed by merging contiguous high density regions and similarly for the low density islands. The constructed islands are then overlapped and tested for differential expression (DE) across samples using *Welch's t-test* (an adaptation of Student's t-test) or *Wilcoxon* test. Low density islands resulting from the overlap are removed due to the lack of enough alignment to perform the DE test.

To detect which genetic features (e.g. genes) are differentially expressed between samples, the significance of DE islands that overlap with each feature is combined using combined *p*-values methods (e.g. Fisher's method). DE islands that do not correspond to any feature are considered novel DE regions. Those regions are annotated along with their closest features. To evaluate the performance of the IBSeq approach, a comparison analysis is conducted to compare the approach with a number of existing differential expression analysis methods using benchmark Microarray Quality Control (MAQC) RNA-Seq datasets. Using ROC curves and auROC as performance metrics, results show that IBSeq outperforms all other methods as illustrated by an increased auROC.

1.3 Dissertation Outline

The reminder of this dissertation is outlined as follows. Chapter 2 reviews the basic concepts of molecular biology. It starts with a description of basic molecular biology concepts such as cells, DNA, RNA, proteins, genes, and genomes. It also

explains the Central Dogma of Molecular Biology and the process of protein synthesis. This chapter also discusses the mechanism of alternative splicing and shows the different types of alternative splicing events.

Chapter 3 gives a detailed overview of high-throughout next-generation sequencing technologies. It starts with reviewing first-generation sequencing techniques including the Sanger and Maxam-Gilbert methods. Section 3.2 explains in detail the next-generation sequencing techniques and gives a brief description of the widely known sequencing platforms (e.g. Roche 454, Illumina, and SOLiD). It also provides a brief description of NGS applications. Next-generation sequencing data format is discussed in Section 3.4. The chapter ends with a brief review of genome assembly and alignment and explains some of their algorithms.

Chapter 4 provides a detailed description of RNA-Seq Methodologies. It begins with a brief review of the developmental milestones of transcriptome analysis and discuss the different methods (e.g. Expressed Sequencing Tags, Serial Analysis of Gene Expression, and Microarrays) applied in this realm. Section 4.3 deals in detail with RNA-Seq methodologies. It discusses the RNA-Seq workflow and provides detailed information about its applications such as transcript assembly, transcript quantification, and differential expression. It also explains the current state-of-the-art approaches for RNA-Seq analysis particularly in the area of differential expression which is the main focus of this dissertation.

Chapter 5 describes IBSeq, an island-based approach for RNA-seq differential expression analysis. This chapter discusses all aspects of the IBSeq approach. It begins with a brief description of the current available methods for RNA-seq differential expression analysis and shows the limitations associated with them. Section 5.2 describes the IBSeq approach for performing differential expression using RNA-Seq data and explains the design and implementation of the different steps in this

approach. Section 5.3 reports the results of evaluating the performance of IBSeq by comparing it with a number of current differential expression analysis methods using publicly available benchmark RNA-Seq datasets. The chapter ends with a conclusion presented in Section 5.4.

Chapter 6 presents a comparative study of different combined p -value methods for gene differential expression using RNA-Seq data. Since IBSeq approach determines whether a feature (e.g. gene, exon) is significantly differentially expressed or not between samples by combining the p -values of the regions corresponding to that feature, this study is conducted in order to determine which combining p -values method provides the best performance among the widely used methods. In this chapter, six different combining p -value methods are compared using publicly available RNA-Seq datasets. Section 6.2.1 describes the different combining methods considered in this study and Section 6.3 reports the results of this comparison. Section 6.4 concludes this chapter.

Chapter 7 is dedicated for discussions and conclusions and presents the potential future directions.

CHAPTER 2

BIOLOGICAL BACKGROUND

2.1 Organisms and Cells

Cells are the basic unit of all living organisms on earth. They are the components that make any thing alive and control the structures and functions of any unit in all organisms. The size of cells may vary, but typical cell size may range between 1 (*bacteria*) and 100 (*plant*) micrometers [14]. Human bodies are made up of trillions of cells with at least 200 distinct cell types. Organisms can be classified based on their cell type into two main categories:

1. **Single-cell:** organisms made up of only one cell, such as *bacteria* and *yeast*, are called single-celled or uni-cellular organisms.
2. **Multi-cellular:** organisms that consist of more than one cell are known as multicellular organisms.

Organisms are called *prokaryotes* if they lack a cell nucleus (the place where DNA is contained and protected) or any membrane-encased organelles, otherwise they are known as *eukaryotes*. Unicellular organisms may be prokaryotic or eukaryotic. However, most prokaryotes are single cell organisms. Both prokaryotes and eukaryotes cells have a protection barrier composed of a *phospholipids* and *proteins* called a *plasma membrane* used to enclose the *cytoplasm* and protect the cell from the outside environment. It regulates the movement of materials into and out of the cell.

One of the most important organelles in the cell structure are the *ribosomes* which provide the machinery for protein synthesis. Figure 2.1 shows an example of typical animal, bacteria, and plant cells structures.

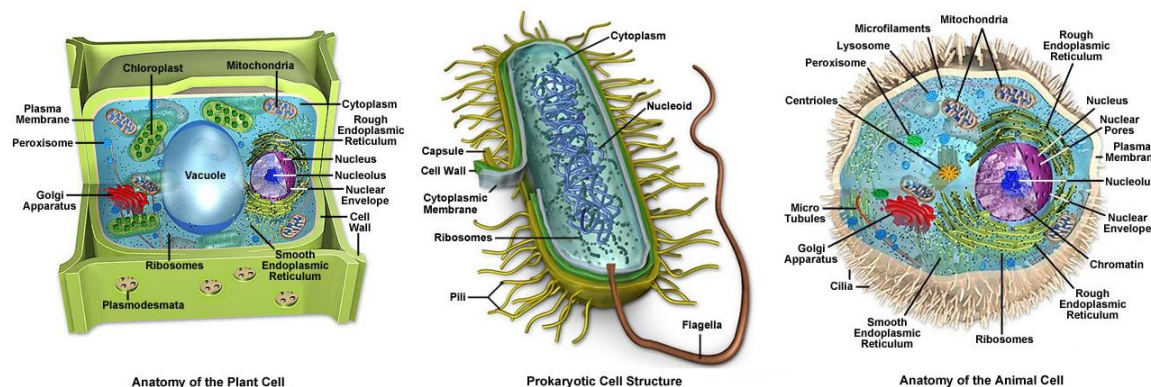


Figure 2.1: Typical animal, bacteria, and plant cells [156]. Used with permission.

In multicellular organisms, cells are organized in tissues (group of cells) which perform a specific function, and several tissues are organized to form organs.

2.2 Nucleic Acids

In all living organisms, genetic information is stored in two types of nucleic acid molecules called *deoxyribonucleic acid* (DNA) and *ribonucleic acid* (RNA). These molecules are used to carry genetic information in the cell and transmit it from one generation to the next [134]. DNA and RNA molecules are polymers consisting of four units: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T) [69]. For RNA, Thymine (T) is replaced by Uracil (U). These units are called *nucleotides* (also known as *bases*) and consist of three main parts: sugar, phosphate group, and one of the two bases (a *purine* or a *pyrimidine*) as shown in Figure 2.2. Since DNA is organized in a double-helix, the nucleotide bases form a complementary pair where Adenine is complementary to Thymine (or Uracil in the case of RNA) and Guanine is complementary to Cytosine. Figure 2.3 shows an example of double-stranded DNA.

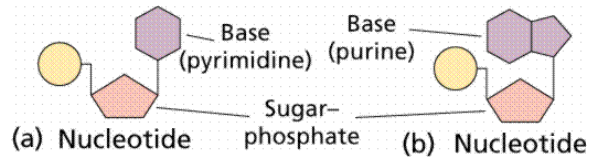


Figure 2.2: The chemical structure of nucleotides [124].

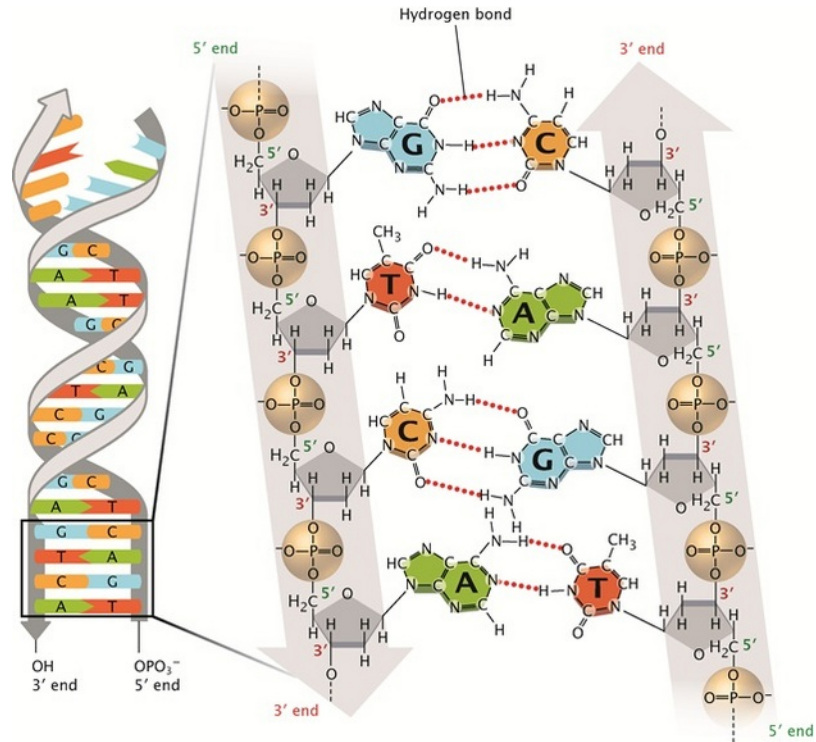


Figure 2.3: DNA double helix [123].

2.3 Proteins

The fundamental components of all living things are proteins [69]. Besides water, proteins are the most common substance in the cell where they take up to 20% of a eukaryotic cell's weight [14]. They do most of the work in the cell and make the life of all organisms possible. Proteins consist of one or more long chains called *polypeptides*. Each polypeptide is made up of 20 different small subunits called *amino acids*, linked together by *peptide bounds* into a single-linear chain. All 20 amino acids have the same basic structure consisting of an amino group (NH_2), carboxyl group

(COOH), and a side-chain (R) attached to a central alpha carbon (C). Figure 2.4 shows the basic structure of an amino acid and how amino acids are linked together by *peptide bounds*.

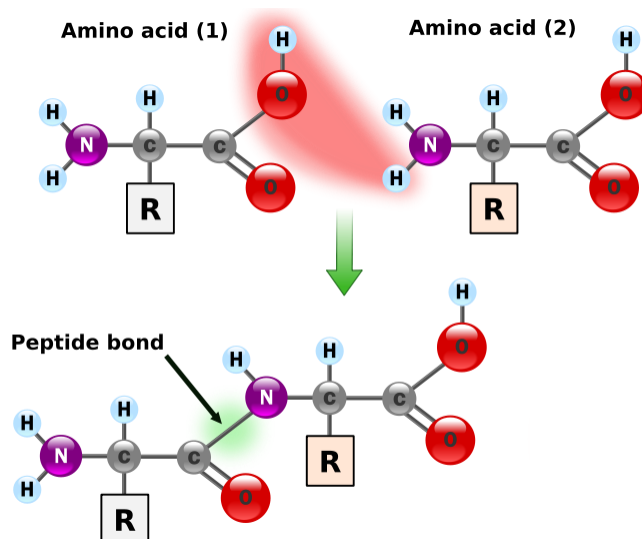


Figure 2.4: The chemical structure of amino acids and how they are linked [2].

The linear sequence of amino acids that make up a protein is the first level of protein structure also known as the *primary structure*. This linear amino acid sequence is derived from the corresponding nucleotide sequence of the messenger RNA in a process called *translation* (discussed in the next section) during protein synthesis.

2.4 Central Dogma of Molecular Biology

In 1958, Francis Crick used the term “Central Dogma” to describe the phenomena that biological information flow occurs in only one direction, from DNA to RNA to proteins. The Central Dogma of Molecular Biology states that once sequential information gets into proteins, it can not get out again. This shows the one way flow which indicates that once genetic information has passed to proteins, it cannot flow back to nucleic acids. In 1970, Crick restated the phenomenon of the Central Dogma

of Molecular Biology and described the information flow in nine possible transforms (Figure 2.5) classified into three groups:

1. **General transfers:** $\text{DNA} \rightarrow \text{DNA}$, $\text{DNA} \rightarrow \text{RNA}$, and $\text{RNA} \rightarrow \text{Protein}$. These transfers have strong evidence that they occur in all cells [29].
2. **Special transfers:** $\text{RNA} \rightarrow \text{RNA}$, $\text{RNA} \rightarrow \text{DNA}$, and $\text{DNA} \rightarrow \text{Protein}$. These transfers may occur under some specific conditions such as in a laboratory or in the case of some viruses [29].
3. **Unknown transfers:** $\text{Protein} \rightarrow \text{Protein}$, $\text{Protein} \rightarrow \text{RNA}$, and $\text{Protein} \rightarrow \text{DNA}$. These transfers are very unlikely to occur [29].

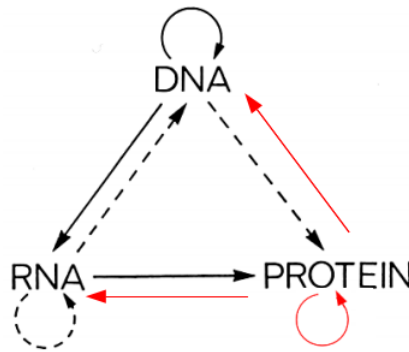


Figure 2.5: The Central Dogma of Molecular Biology [29]. Solid arrows represent the *general transfers* and dotted arrows show the *special transfers*. *Unknown transfers* are shown in red.

The *general transfers* describe the flow of information from one form to another in three primary biological processes, DNA replication, transcription, and translation as shown in Figure 2.6.

1. *Replication* is a process by which a cell makes an exact copy of its DNA molecule prior to a cell division. Thus, every time the cell divides, the double strands of the DNA is separated into single-stranded regions by an enzyme called *DNA*

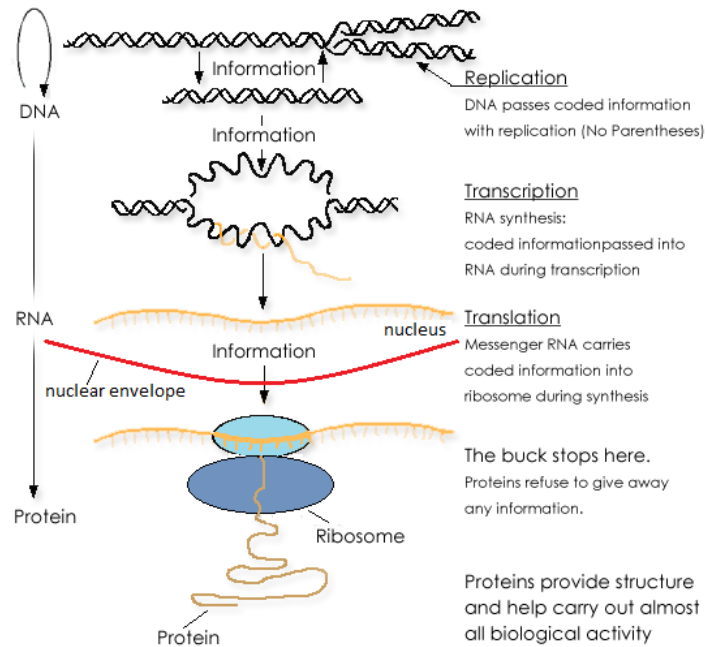


Figure 2.6: The journey of genetic information from DNA to protein. Image adapted from [152].

Helicase. Each strand will serve as a template to form a new strand of complementary DNA resulting in two identical copies of the DNA molecule, each will consist of an old strand and a new complementary strand as shown in Figure 2.7. The formation of the new strands is performed by enzymes called DNA

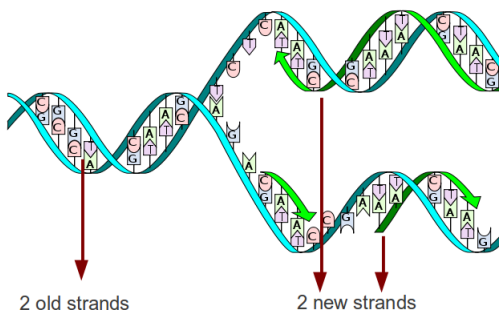


Figure 2.7: The DNA replication [33]

polymerase where they bind to the old single strands and begin synthesizing the new complementary strands. Figure 2.8 shows an example of DNA replication.

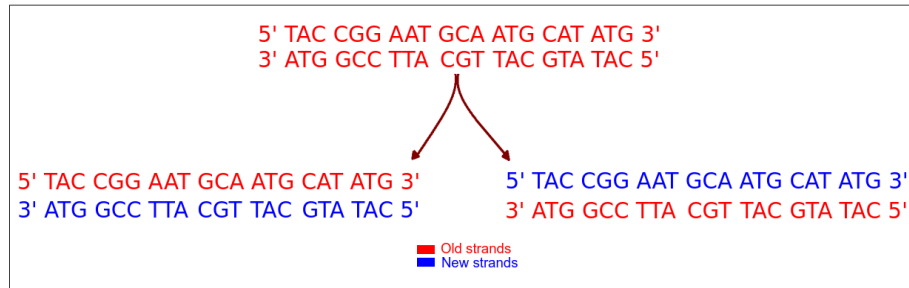


Figure 2.8: Example of DNA replication

2. *Transcription* is the process by which genetic information is transferred from a portion of DNA to an RNA molecule called messenger RNA (mRNA). This process is performed by an enzyme called *RNA Polymerase* that binds to a region on the DNA called a *promoter* (a DNA region allowing RNA polymerase and transcription factors to bind to initiate the transcription of a particular gene) and starts making a copy of a complementary RNA sequence known as the *primary transcript* or *precursor mRNA* (*pre-mRNA*). The pre-mRNA is made up of *introns* and *exons* as shown in Figure 2.9. Exons are then joined

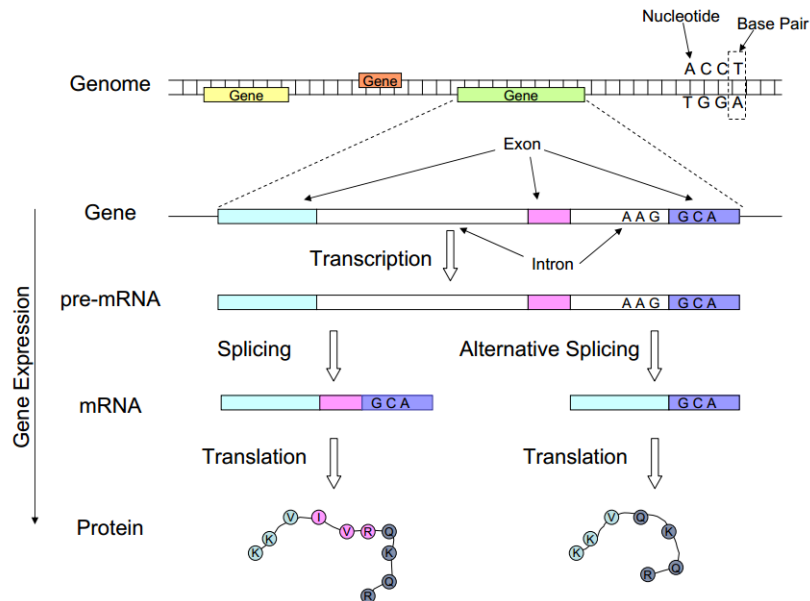
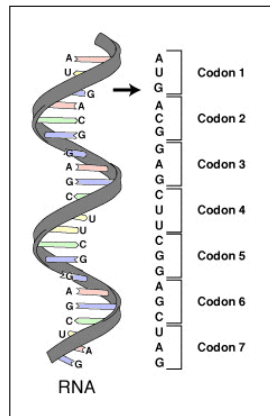


Figure 2.9: Protein synthesis process [43]

together and introns are cut out to form a molecule called *mature mRNA* (see Figure 2.9) in a process known as *RNA splicing*. Some protein transcription factors bind in or next to the promoter initiating transcription by facilitating the unwinding of the double stranded DNA to allow RNA polymerase to read only one single strand DNA and create mRNA molecule. The transcription process is terminated when RNA polymerase reaches the termination sequence. RNA polymerase then releases the mRNA and detaches from the DNA.

3. *Translation* is the process by which messenger RNA (mRNA) is translated into a linear chain of *amino acids* that forms proteins (Figures 2.6 and 2.9). This process is performed by ribosomes (very large complexes of RNA and proteins) with the help of several types of transfer RNA molecules (tRNA), all within the cytoplasm. In mRNA, each three non-overlapping bases called a *codon* map to a particular amino acid. Since there are four bases, 64 different combinations or codons ($4^3 = 64$) can be formed constructing a genetic codon table known as the genetic code. Figure 2.10 shows an example of codons and the genetic code.



		Second base				
		U	C	A	G	
First base	U	UUU } Phenylalanine UUC } UUA } Leucine UUG }	UCU } Serine UCC } UCA } UCG }	UAU } Tyrosine UAC } UAA } Stop codon UAG } Stop codon	UGU } Cysteine UGC } UGA } Stop codon UGG } Tryptophan	U C A G
	C	CUU } Leucine CUC } CUA } CUG }	CCU } Proline CCC } CCA } CCG }	CAU } Histidine CAC } CAA } Glutamine CAG }	CGU } Arginine CGC } CGA } CGG }	U C A G
	A	AUU } Isoleucine AUC } AUA } AUG } Methionine (Start codon)	ACU } Threonine ACC } ACA } ACG }	AAU } Asparagine AAC } AAA } Lysine AAG }	AGU } Serine AGC } AGA } Arginine AGG }	U C A G
	G	GUU } Valine GUC } GUA } GUG }	GCU } Alanine GCC } GCA } GCG }	GAU } Aspartic Acid GAC } GAA } Glutamic Acid GAG }	GGU } Glycine GGC } GGA } GGG }	U C A G

Figure 2.10: The genetic code. (**Left**) series of codons in a part of messenger RNA (mRNA) molecule [45]; (**Right**) The genetic codon table representing the genetic code.

Most of the 20 amino acids are encoded by more than one codon which in general differs only in the last nucleotide. Three of these codons (UAA, UAG, and UGA) are used to end the synthesis of a protein sequence and do not encode for any amino acid (*stop codons*). In contrast, one codon (AUG) is used to start the translation process (*start codon*) and codes for the amino acid methionine (Met), the first amino acid in the polypeptide chain. Thus, the purpose of the *translation* process is to map a sequence of *codons* to a sequence of *amino acids*. The process starts by transporting mature mRNA out of the nucleus to the cytoplasm. The ribosome then binds to the mRNA at the *start codon*. As mRNA passes through the ribosome, the ribosome starts matching *anticodon* sequences carried by the tRNA to the mRNA codon sequence forming a polypeptide chain. This process continues until the ribosome reaches a *stop codon* which ends the synthesis of the polypeptide chain and releases it.

2.5 Genes and Genomes

In all organisms, genetic information is stored in one or more replicable double-stranded DNA molecules called *chromosomes* (Figure 2.11). Each chromosome is made up of two copies of DNA molecule linked together. They contain genes, regulatory elements and other nucleotide sequences. The DNA molecules are wrapped around proteins called *histones* resulting in a structure known as *chromatin* (Figure 2.11). Whereas *prokaryotic* cells have a single chromosome, *eukaryotic* cells have one or more chromosomes. In a single human cell, there are 23 pairs of linear chromosomes, for a total of 46 chromosomes. Twenty-two of these pairs, called *autosomes*, are similar in both males and females. The 23rd pair is the sex-determination system and is referred to as the X and Y chromosomes. Females have two X chromosomes in their cells, while males have both X and Y chromosomes.

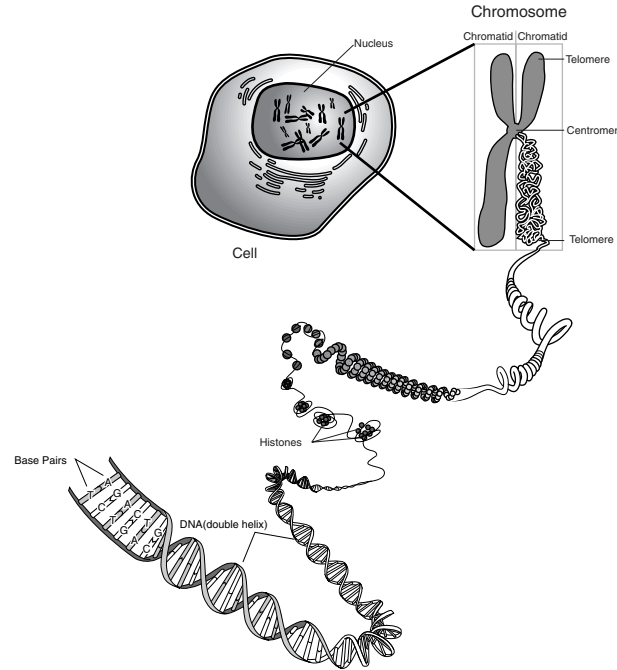


Figure 2.11: The structure of chromosomes [24].

Genome refers to the complete set of genetic information in an organism [69]. A *gene* is DNA segment located on a chromosome that has all required information to make a gene-product(s), which can be protein or RNA. They are the basic units of inheritance. Each chromosome contains thousands of genes. It was estimated by the Human Genome Project that humans have between 20,000 and 25,000 genes [68].

2.6 Gene Expression and Alternative Splicing

2.6.1 Gene Expression

Gene expression is a process by which gene regions on a chromosome are transcribed to RNA and, in most cases, translated to proteins as discussed in section 2.4 and shown in Figure 2.9. Every gene is composed of a set of two segments called *exons* and *introns*. While exon segments end up coding for proteins in the *coding sequence* (expressed sequence), intron segments do not code for proteins and there-

fore represent the *non-coding sequence*. As discussed in section 2.4, the transcription process first generates the primary transcripts which contain both exons and introns. Next, RNA binding proteins called *splicing factors* initiate the *splicing* process in which *introns* are spliced out at *spliceosomes* (a snRNA and protein complex that is used to remove introns from a transcribed pre-mRNA) and *exons* are linked together and transported to the *ribosome* for translation to proteins.

While thousands of genes are present in cells, not all genes are expressed at once. At any particular time, only a small fraction of these genes are expressed. In general, genes are said to be *on* (expressed) if their molecular product can be synthesized or *off* (not expressed) if they cannot be. Thus, it is the role of cells to determine which genes to turn on and which genes to turn off during any time. This process is called *gene regulation*. For instance, a brain cell turns on genes that encode brain proteins, but a muscle cell will leave those genes off. The measurement of gene expression is determined by looking at how much a particular gene is expressed within a cell or tissue. One approximate measure of gene expression is the amount of mRNA produced by various genes in the cell. RNA gene expression analysis is useful for cell function and differentiation studies that estimate which and when genes are expressed and how much their expression changes across biological conditions. There are several techniques for RNA gene expression analysis including Serial Analysis of Gene Expression (SAGE), microarrays, and lately high-throughput mRNA sequencing (RNA-Seq).

2.6.2 Alternative Splicing

Alternative splicing is the process by which pre-mRNA *exons* of a gene are rejoined together in multiple ways producing different mRNA variants known as *isoforms* or *splice variants* during the *RNA Splicing* process. In the human genome for

instance, current estimates suggest that more than 90% of the genes have multiple protein isoforms [76]. The process of alternative splicing is shown in Figure 2.12.

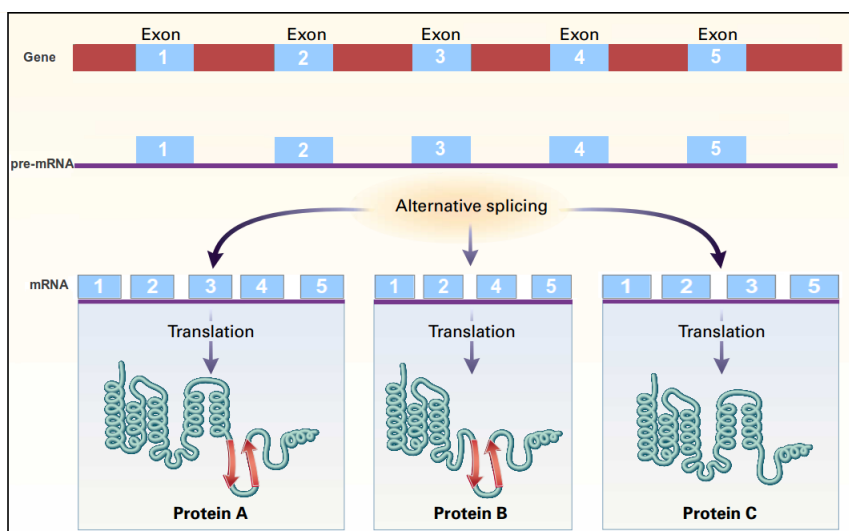


Figure 2.12: Alternative splicing. One gene can produce multiple proteins [54].

When the primary transcript enters the spliceosome, the spliceosome recognizes the sequence of exon-intron boundaries at regions known as the *splice donor site* GU (in the 5' direction) and *splice acceptor site* AG (in the 3' direction). Since those two sequences are not sufficient for the spliceosome to recognize the existence of an intron, another sequence called the *branch site* located 20-50 bases upstream of the acceptor site is used. The spliceosome also recognizes a region upstream 5' from the AG sequence and located about 5-40 bases before the 3' end of the intron being spliced known as *polypyrimidine tract*. This region is rich in pyrimidine nucleotides (C and U). Figure 2.13 shows the regions involved in the splicing process.

The mechanism of splicing process shown in Figure 2.14 uses five small nuclear ribonucleic-protein complexes (snRNAs) known as U1, U2, U4, U5, U6. It begins when U1 binds to the splice site end at the 5' direction making the branch site bind to the G nucleotide at the donor site to form a phosphodiester linkage [19]. Then, U2 binds to the branch site sequence (denoted by A in Figure 2.14). In the next step, U4,

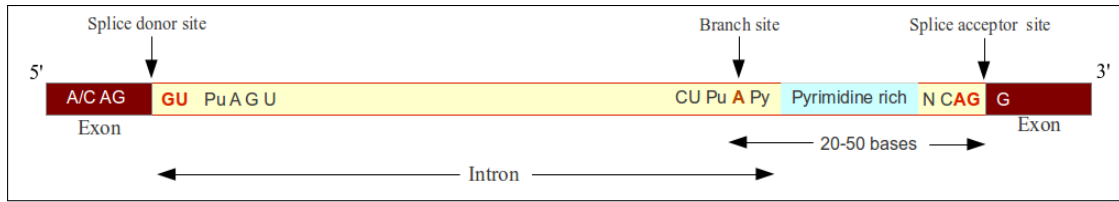


Figure 2.13: Sequences involved in the RNA splicing. (Pu=A or G; Py=C or U) [128].

U5, and U6 complex binds to the 5' splice site replacing the position of U1. U1 and U4 are then displaced and U6 binds to U2 at the 5' splice site and near the branch site. In the final step, U5 binds to the exon sequences and the intron is removed.

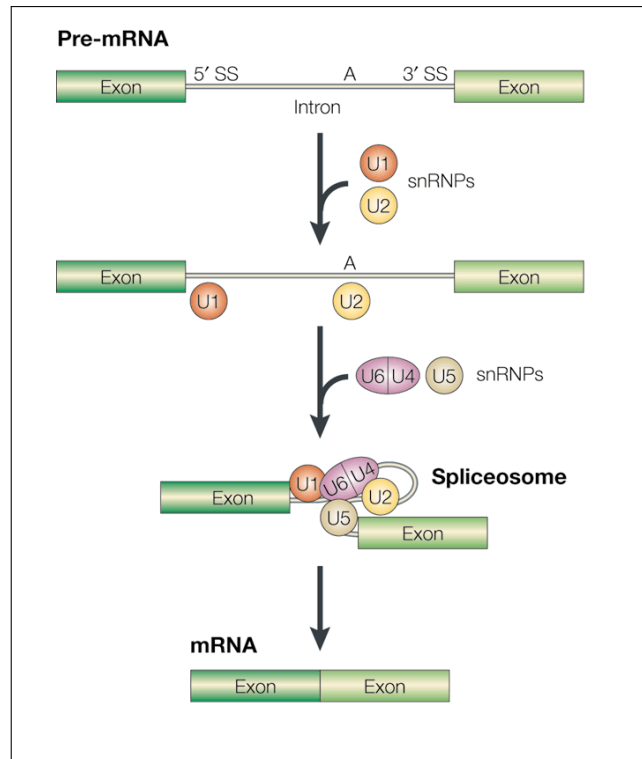


Figure 2.14: Alternative splicing mechanism [36].

There are several types of alternative splicing events, some of which may occur at the 5' or 3' untranslated regions (UTRs), others may occur at the coding regions. Figure 2.15 shows an overview of possible alternative splicing events.

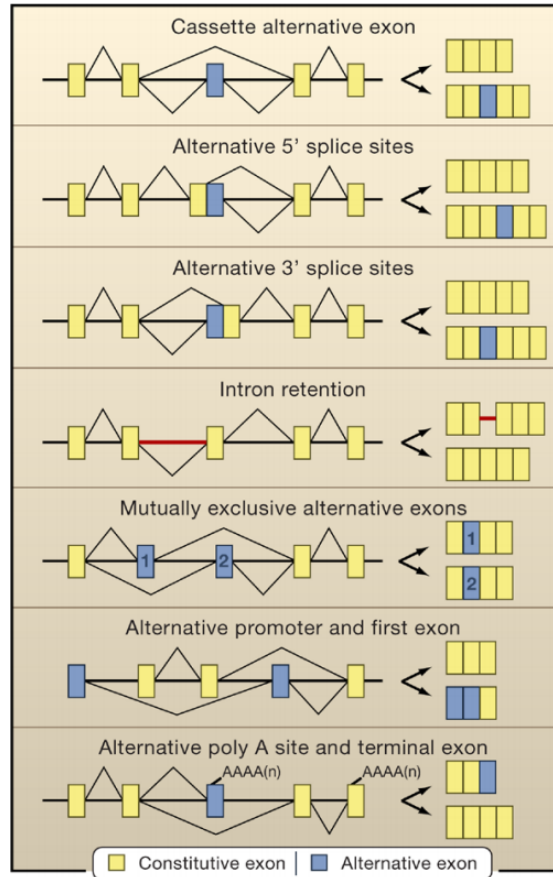


Figure 2.15: Alternative splicing events [11].

- **Exon Cassette (Exon skipping):** in this type, one or more exons are included or excluded from the final processed mRNA product. It is the most common gene splicing in mammals. However, this type is extremely rare in *prokaryotes*.
- **Alternative 5' splice sites:** in this type, two or more splice sites are recognized at the 3' end of an exon. The *donor site* (alternative 5' splice junction) is used, changing the 3' boundary of the upstream exon. This type may account for about 18.4% of alternative splicing events in *eukaryotes* [76].
- **Alternative 3' splice sites:** like the alternative 5' splice sites, alternative 3' splice sites occurs when two or more splice sites are recognized at the 5' end of an exon. The *acceptor site* (alternative 3' splice junction) is used, changing the

5' boundary of the downstream exon. This type may account for about 7.9% of alternative splicing events in *eukaryotes* [76].

- **Intron retention:** this type simply occurs when an intron remains in the final mRNA transcript.
- **Mutually exclusive exons:** this type occurs when multiple cassette exons are used in a mutually exclusive manner. Namely, one of two exons remains in the mature mRNA but not both.
- **Alternative promoter:** this type usually occurs when two promoters are available. A different promoter is used to generate different splice variants. The exons of 5' terminal of the processed mRNA can be switched to generate alternative isoforms. The specific transcription factors of the cell determine which promoter to use.
- **Alternative polyadenylation sites (Poly A):** similarly to alternative promoter, this type occurs when 3'UTR exons of the processed mRNA are alternatively spliced producing alternative polyadenylation sites [19].

In addition to the above discussed events, there are other exon-related events that may occur as well. These events are discussed below and shown in Figure 2.16.

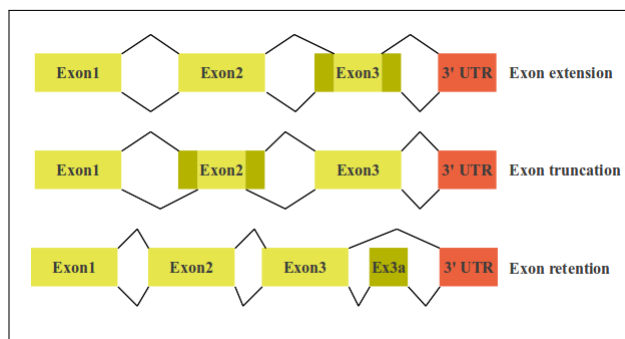


Figure 2.16: Alternative exon events [19].

- **Exon extension:** this event may occur when one or more exons are extended by adding an additional sequence to a transcript producing a slightly modified mRNA sequence.
- **Exon truncation:** this event may occur when a partial (not fully) of a cassette exon is added to a mRNA sequence. This is usually done by removing either 3' or 5' (sometimes both) producing different translated protein [19].
- **Exon retention:** this type of alternative splicing events may occur when additional exons located in an intronic region are included in the final mRNA sequence resulting in an altered translated protein.

CHAPTER 3

NEXT-GENERATION SEQUENCING

3.1 First-Generation Sequencing Methods

Until 2004, the dominant technique for DNA sequencing was the one introduced by Frederick Sanger in 1977 that bears his name. The *Sanger* method, also known as *dideoxy sequencing* or *chain termination*, uses modified nucleotides called *dideoxynucleotides* (ddNTPs) with the normal nucleotides (NTPs). Essentially, the structure of *dideoxynucleotides* is the same as normal *nucleotides* except they contain a hydrogen group on the 3 carbon instead of a hydroxyl group (OH) which act as chain terminators. In this method, DNA samples are divided into four separate sequencing reactions, each containing the single-stranded DNA to be sequenced, the four normal deoxynucleotides (dATP, dGTP, dCTP and dTTP), DNA polymerase, DNA primer, and one of the four radioactively or fluorescently labelled *dideoxynucleotides* (ddATP, ddGTP, ddCTP and ddTTP). Since the *dideoxynucleotides* lack the 3'-OH group required to form a *phosphodiester bond* between two nucleotides, once dideoxynucleotides are incorporated, the process is halted, stopping any further formation resulting in a collection of DNA fragments with different lengths. Each fragment is terminated by the same dideoxynucleotide in each of the four reactions. This collection of DNA fragments is then heat denatured and run on a gel electrophoresis to separate fragments by size. Each reaction is run on one of four lanes (A, T, G, and C), each with a different ddNTP. Thus, the lane containing the ddATP

for instance will have only those fragments that terminate at an adenosine (A) and the same for ddGTP which will have only fragments that all stop at guanine (G) and so on. When all lane contents have been read across the gel, DNA bands are visualized by exposing the gel to a UV light or X-ray film and the DNA sequence is then read from the film. Figure 3.1 illustrates the workflow of Sanger method.

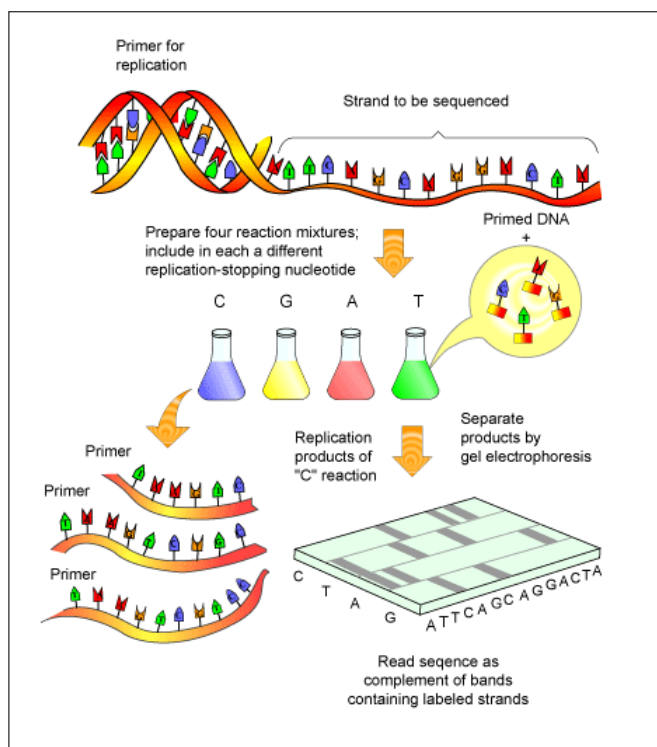


Figure 3.1: The workflow of Sanger sequencing method [73].

Another first-generation sequencing method was introduced by Maxam and Gilbert in 1977 known as a *chemical cleavage* method. This method is based on chemical modification of DNA and subsequent cleavage at specific bases. Like the Sanger method, the DNA template is split into four reactions, G , $A + G$, C , and $C + T$. In each reaction, the DNA fragment is radioactively labeled at the 5' end and chemically cleaved at one of the four nucleotides. Thus, *purines* ($A + G$) might be depurinated by formic acid whereas *pyrimidines* ($C + T$) are methylated using hydrazine. By using hot *piperidine*, the DNA fragments are then cut into a series

of labeled fragments. In order to separate fragments by size, the fragments in the four reactions are arranged side by side in a gel electrophoresis. In the next step, fragments are visualized by exposing the gel to X-ray film for autoradiography which generates a series of dark bands each corresponding to a radio labelled DNA fragment [34]. Figure 3.2 describes the Maxam-Gilbert workflow.

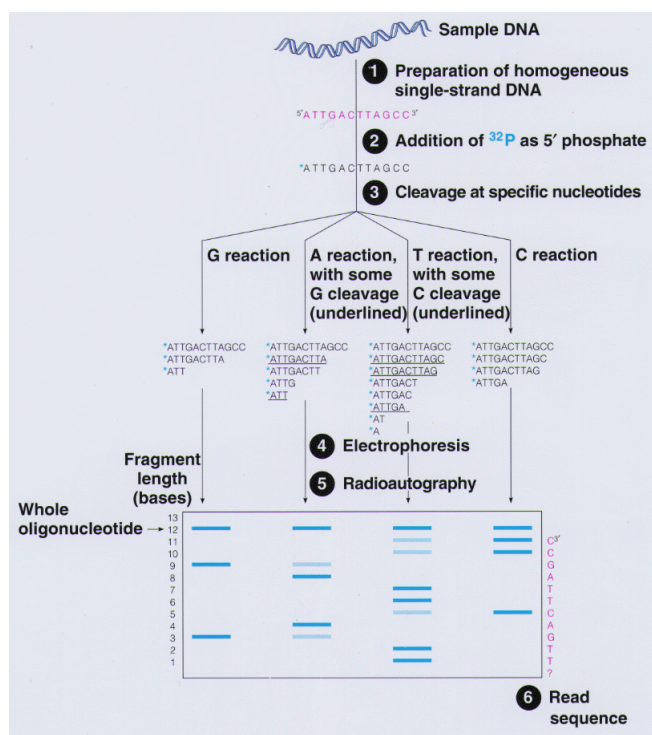


Figure 3.2: Maxam-Gilbert method for DNA sequencing [103]

Although this method was accurate and more popular, it did not take hold as the preferable sequencing technique due to the complexity and the extensive use of toxic chemicals.

Since the automated Sanger method can only accurately sequence up to 1000bp, researchers had to think of other methods that can sequence longer sequences. As a result, a new method called *shotgun sequencing* (also known as *shotgun cloning*) was developed. In this method, the DNA segment of interest is cut into smaller fragments using restriction enzymes or mechanical shearing. Each fragment is sequenced indi-

vidually and the sequences of these fragments are then reassembled based on sequence overlaps into continuous sequence resulting in a complete DNA sequence (Figure 3.3). The shotgun sequencing method was used to sequence the entire human genome by the Human Genome Project (HGP).

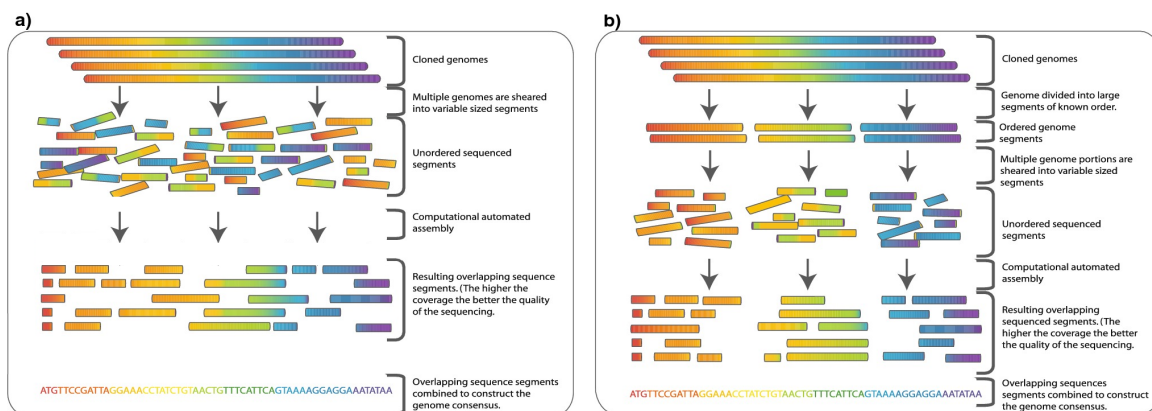


Figure 3.3: (a) Description of Shotgun sequencing method. (b) To reduce the complexity of normal shotgun sequencing resulting from the large sequences, a hierarchical approach is used [26]. The genome is broken down into a set of large equal segments with known order (clone-based methods) which are then sequenced using the normal shotgun sequencing.

3.2 Next-Generation Sequencing Technologies

Due to the limitations associated with first-generation sequencing technologies such as the inability to sequence large genomes in a reasonable time with an optimal cost, there was a great demand to develop new techniques. As a result, new, fast, inexpensive, and more accurate techniques known as *Next-Generation Sequencing (NGS)* were introduced in 2004. Unlike the old sequencing techniques, which are based on chain termination methodologies, the new techniques are based on parallel sequencing. Thus, they are also known as *Massively Parallel Sequencing Techniques*. NGS technologies share similar protocols to perform the sequencing process classified

into three main steps, (1) *template preparation*, (2) *sequencing and imaging*, and (3) *data analysis* [105]. Each technique has its own protocols to perform each step and each generates a different type of data in the form of short sequences called *reads*.

With their ability to generate tens of millions of short sequences in a relatively short time with a low cost, NGS technologies has expanded the frontiers of genomic and transcriptomic research opening new avenues for genetic investigations. However, they require long run times spanning from 8 hours to 10 days based on the platform and the read type (single-end or paired-end) [99]. Several NGS platforms are commercially available including Roche/454 GS FLX, Illumina/Solexa, Applied Biosystems SOLiD (ABI) analyzer, Polonator G.007, Helicos/HeliScope, Pacific BioSciences/RS, and IonTorrent. Each platform generates a different read length ranging from 35-1000bp within a different run time and each has a different throughput. Table 3.1 shows a detailed description of each sequencing platform while Table 3.2 describes the key applications used by each platform.

The first three platforms, Roche 454, Illumina, and SOLiD described in Table 3.1 are the most widely used sequencers dominating the sequencing market. Helicos/Heliscopes, and PacBio/RS are refereed to as *next-next generation* sequencing platforms (or third-generation sequencing). In the next sections, a brief description of each sequencing platform is given.

3.2.1 Roche/454 GS FLX Sequencer

The Roche 454 was introduced to the market in 2004 as the first next-generation sequencing machine and is currently developed by 454 Life Sciences Corporation. The sequencing technology of Roche 454 is based on the *sequencing-by-synthesis* technique known as *pyrosequencing* [168]. This machine uses an emulsion PCR amplification technique to make copies of the DNA templates [30] in which every DNA fragment is

Table 3.1: Detailed information for current NGS platforms.

Sequencing Platform	Amplification Method	Read length (bp)	Throughput (run)	Technology
Roche/GS FLX+	emPCR	Up to 1,000	700 Mb	Pyrosequencing
Illumina/HiSeq 2500	Bridge-PCR	2x100	600 Gb	Sequencing by synthesis
ABI/SOLiD 5500xl	emPCR	50-100bp	>100 Gb	Sequencing by ligation
Polonator/G.007	emPCR	26	8-10 Gbp	Sequencing by ligation
Helicos/Heliscope	No	25-55	21-35 Gb	Single Molecule sequencing
PacBio RS	No	1,000-10,000	13 Gb	Single Molecule Real Time
IonTorrent/Proton	No	100-200	10 Gb	Semiconductor sequencing

bound to a single bead. Each bead is isolated in oil micelles which contain emulsion PCR reactants producing about one million copies of each DNA fragment [168, 30]. During the pyrosequencing process, four different nucleotides are flowed on a solid surface containing a number of wells designed to hold the beads (each well can be used to hold one bead), producing light from a reaction utilizing pyrophosphate generated when nucleotide incorporation occurs [30]. This process continues for a number of cycles and the light for each incorporation is recorded for each bead [30]. Thus, the intensity of the light recorded for a particular well indicates the number of incorporated nucleotides [169]. Initially, the Roche 454 Sequencer had a read length of 100-150 bp, but the more recent, Roche GS FLX Titanium XL+ can produce an average read length up to 1000 bp.

Table 3.2: Key applications of NGS technologies.

Application	Roche 454	Illumina HiSeq	SOLiD 5500xl	Polonator G.007	PacBio RS	Ion Torrent
Whole Genome Sequencing	yes	yes	yes	yes	no	small genomes
Targeted Re-sequencing	yes	yes	yes	yes	yes	yes
<i>De novo</i> Sequencing	yes	yes	yes	yes	yes	yes
Whole Transcriptome Sequencing	yes	yes	yes	yes	no	yes
miRNA Noncoding RNA	yes	yes	yes	yes	no	yes
Epigenetics Gene Regulation	yes	yes	yes	yes	no	yes
Metagenomics	yes	yes	yes	no	no	no
SNP Genotyping & CNV	yes	yes	yes	yes	yes	no

3.2.2 Illumina/Solexa Genome Analyzer

The Illumina/Solexa Genome Analyzer is the most widely used sequencer which was introduced to the market as the second NGS machine. Initially, the platform was introduced by Solexa in 2006 which was later renamed as the Illumina Genome Analyzer (GA) [30]. The sequencing technology of Illumina is based on a *sequencing-by-synthesis* technique. Unlike Roche 454 which uses emulsion-PCR for DNA template amplification, Illumina GA uses a technique called *solid-phase amplification*[105]. In this technique, all four nucleotides are added simultaneously along with the DNA polymerase into oligo-primed cluster fragments in flow cell channels [168] (8-channel sealed glass microfabricated device) which allows bridge amplification

of those fragments producing multiple DNA copies (or clusters) [99] for sequencing. As a result, a reverse complimentary copy of the template DNA is generated. The generated clusters are then imaged and the incorporation of the next cycle of nucleotides are begun after chemically removing the 3' blocked groups and the flurophores of the next incorporation [30]. In the last step, the generated images are analyzed resulting in a separate sequence for each cluster. Given the fact that a high percentage of published papers use short read sequences produced by Illumina technology, Illumina platforms are considered to be the most widely used sequencers. At present, the Illumina HiSeq 2500 can produce 2 x 100 bp (pair-end reads) and has a throughput of 600 Gbp/run (see Table 3.1 for more details).

3.2.3 Applied Biosystems SOLiD Sequencer

The ABI SOLiD (**S**equencing by **O**ligo **L**igation and **D**etection) sequencer was developed by Life Technologies and purchased by Applied Biosystems which introduced it to the market in October 2006. The sequencing technology of ABI SOLiD is based on a *sequencing-by-ligation* approach using emulsion-PCR with small magnetic beads to amplify DNA fragments for sequencing [99, 168]. This technique is similar to the one for Roche 454 except that SOLiD beads are much smaller than Roche beads ($1\mu m$ versus $28\mu m$) [30]. The SOLiD sequencer uses DNA ligase and two-base-encoded probes to amplify fragments [105]. In this system, two slides are used per run, each of which can be divided into four or eight data points. Thus, two adjacent bases represents a single data point and each base is interrogated twice [30]. Four dyes are utilized by the two-base encoded probes to encode for 16 possible two base combinations as shown in Figure 3.4.

The SOLiD instrument generates a different type of data known as *colorspace* data based on the concept of the 2-base encoding technique explained above. Thus,

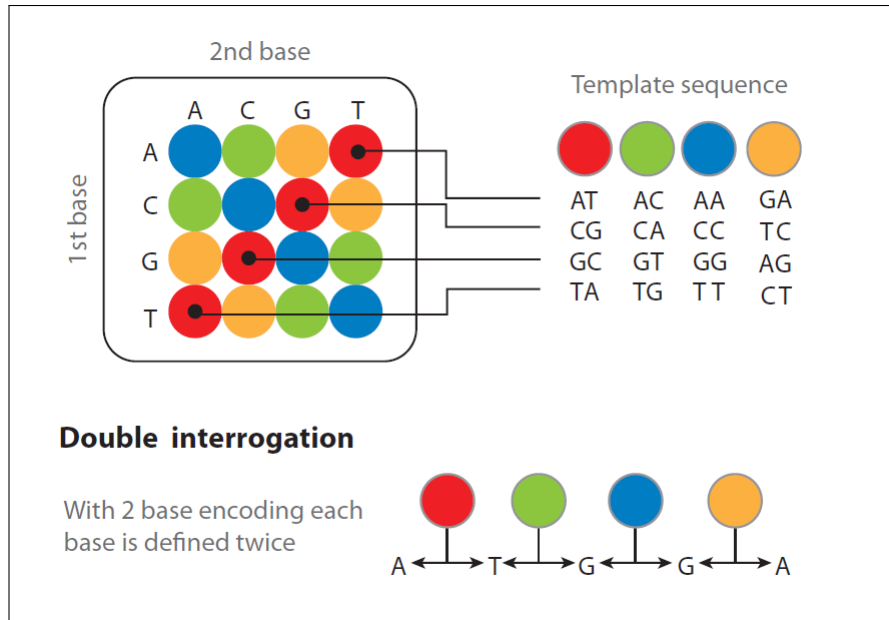


Figure 3.4: Possible dinucleotides encoded by each color [99]

instead of using the normal nucleotide bases A, C, G, and T, SOLiD makes use of the four colors shown in Figure 3.4. Since there are four bases, the four colors are represented as 0, 1, 2, and 3 (blue=0, green=1, yellow=2, red=3) and the final sequence file contains only these numbers. In 2011, Applied Biosystems has introduced the updated platform SOLiD 5500xl with its ability to generate up to 100 Gbp per run with a read length of 50-100 bp.

3.2.4 Polonator G.007

Polonator G.007 is a new platform introduced to the market by Dover Systems in collaboration with the Church Laboratory of Harvard Medical School. The platform is based on the *polony* sequencing technology and uses a *sequencing-by-ligation* technique using randomly arrayed, bead-based, emulsion-PCR for DNA amplification [168]. The machine tends to be cheaper than other NGS machines and inexpensive to operate. The read length of Polonator is 26bp and its throughput is 8-10Gbp per run.

3.2.5 Helicos Heliscope

The Helicos Heliscope was introduced in 2008 as the first single molecular sequencing technology which means that fluorescent nucleotides are added singly. It is classified as the *third-generation sequencing* or *next-next generation sequencing* platform developed by Helicos BioSciences Corporation. This platform has two characteristics that do not exist with the next-generation sequencing platforms discussed earlier. First, since this platform has the ability to sequence a single DNA molecule, there is no need to perform any PCR amplification before sequencing which classifies the platform as *single-molecule real time (SMRT)* [168]. This makes the Helicos Heliscope free from any errors and biases that may occur at the amplification stage which simplifies, eases, and speeds up the process of DNA preparation. Second, the signal generated from hybridizing a particular nucleotide is recorded during the real reaction time making this platform more capable of monitoring it [96]. The average read length of this sequencer is 25-55bp. Although this platform has the advantages of being free from DNA preparation errors, sensitivity can be a big issue [30].

3.2.6 Pacific BioSciences RS

The PacBio RS platform was developed by Pacific BioSciences Corporation and introduced to the market in 2010 as one of the *third-generation sequencing* technologies. Like Helicos Heliscope, this platform uses *single molecule real time sequencing* technology (SMRT) and does not require any PCR amplification before sequencing. This platform performs and analyzes biochemical reactions at the individual molecule level where nucleotides are added singly. This sequencer has several advantages including high speed performance, fast sample preparation (from 4-6 hours instead of days) [96], and producing an average read length of 10,000 bases which is longer than any next-generation sequencer.

3.2.7 Ion Torrent

Ion Torrent introduced its first sequencer (PGM) at the end of 2010. This sequencer was developed by Life Technologies and uses *semiconductor* sequencing technology. This technique works on the concept that when a nucleotide is incorporated into the DNA by a polymerase, a hydrogen ion (H^+) is released. The Ion Torrent sequencer can recognize the incorporation of a nucleotide by detecting and measuring the change in pH [96]. In 2012, Ion Torrent introduced its second generation sequencing platform, The Proton. This sequencer is considered as the first sequencing machine that does not require a fluorescent probe or any scanning materials which make this sequencer fast, cheap, and small in size. The average read length of the Ion Torrent Proton is 100-200 bp and the throughput is up to 10 Gb per run.

3.3 Applications of Next-Generation Sequencing Technologies

The introduction of NGS technologies has made it possible for variety of genomic research areas to utilize the low cost and large amount of data generated by them. To date, these technologies have been comprehensively applied in a variety of realms such as whole-genome sequencing, targeted resequencing, Small RNA sequencing, Epigenetics, and Metagenomics. Table 3.3 describes the common applications of next-generation sequencing technologies in genomic research.

3.4 Next-Generation Sequencing Data

Raw sequence data from next-generation sequencing platforms discussed earlier are stored in an NIH's archive known as the Sequence Read Archive (SRA). It is the primary archive of high-throughput sequencing data where short read sequences are stored and made available to the research community. This has the advantage of reproducing analyses and allowing for new discoveries. The SRA database has

Table 3.3: Applications of NGS technologies [141, 169].

Category	Examples of Applications
<i>De novo</i> genome sequencing	Initial generation of large eukaryotic genomes.
Whole-genome sequencing	Comprehensive polymorphisms and mutation discovery.
Targeted genomic resequencing	Discovery of mutations or polymorphisms.
Transcriptome sequencing	Gene expression and quantification, alternative splicing, transcript annotation, discovery of transcribed SNPs or somatic mutations.
Small RNA sequencing	MircoRNA profiling.
Epigenetics	Transcription factor with its direct targets, histone modification profiling, DNA methylation.
Chromatin immunoprecipitation sequencing (ChIP-Seq)	Genome wide mapping of protein-DNA interactions.
Metagenomics	Environmental genomics.
Personal genomes	Possible usage in personalized medicine.

grown sharply since its first release. Figure 3.5 represents SRA database growth. In addition to the raw NGS data, SRA now stores alignment information in the form of read placements on a reference sequence. In this section, a detailed description of the raw sequence data and alignment data is presented.

3.4.1 Raw Data (Short Reads) Format

Almost all next-generation sequencing platforms report short read sequences in either **Fasta** [94, 119] or **Fastq** [25] format. However, **Fastq** has become quickly the standard format for storing short read sequences. In this section, a brief description of the two common data formats is given.

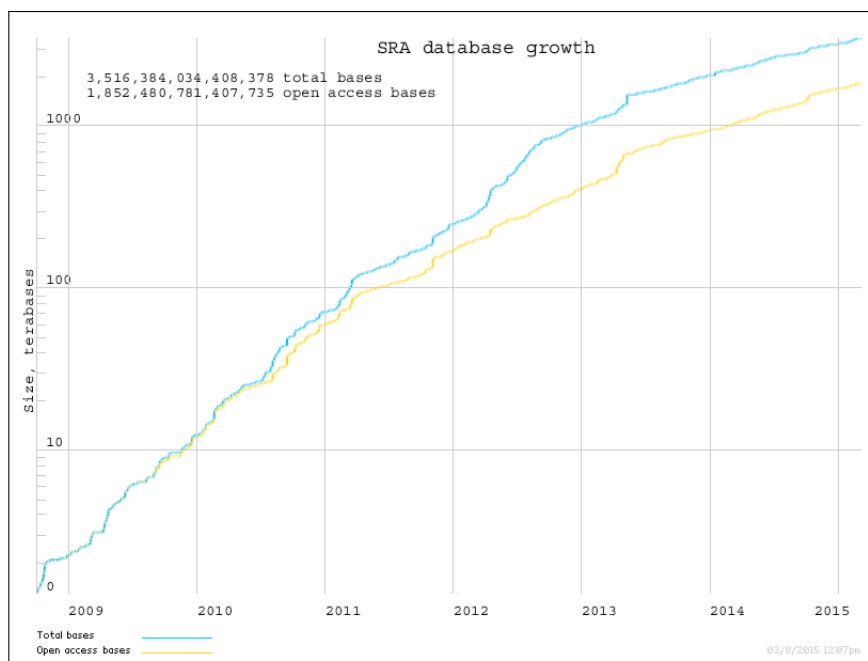


Figure 3.5: SRA database growth. Figure taken from the NCBI website [145]

3.4.1.1 FASTA Format

FASTA format is a text-based format consisting of a single-line description followed by multiple lines of sequence data. It is the simplest and earliest standard format supported by early sequence search algorithms such as FASTA [94, 119] and BLAST [1]. Each sequence in this format starts with an indicator “>” used to distinguish the description line from the sequence data line. The word after the indicator “>” is used as a sequence identifier and separated from the description by a space. Figure 3.6 shows an example of this format.

Since FASTA format does not support quality values, quality values (when they are required) are often reported in separate files as with the Roche 454 sequencer.

3.4.1.2 FASTQ Format

FASTQ format is another text-based format used to store short read sequences. It has become the standard format for storing data from next generation sequencing

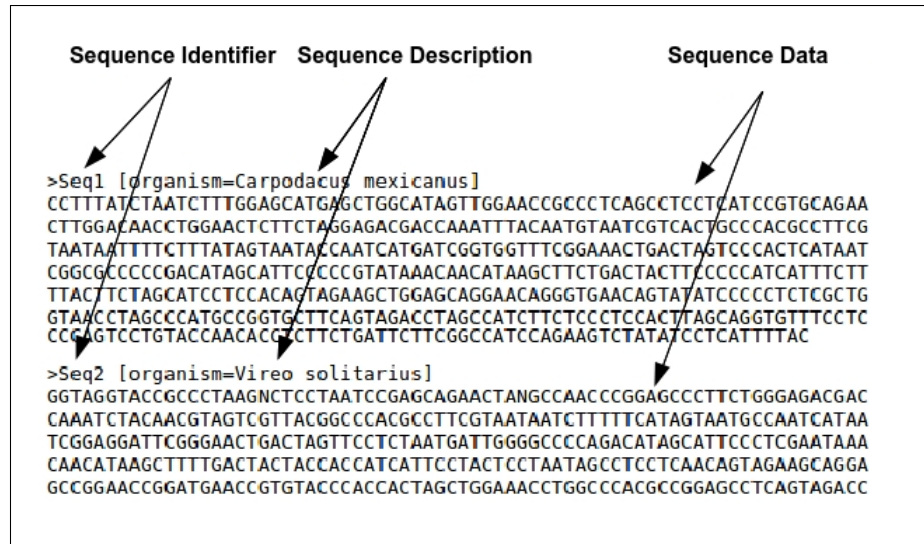


Figure 3.6: Example of FASTA format.

platforms allowing the storage of both sequence and quality scores of each read. Each sequence in FASTQ format consists of four lines. The first line begins with the symbol `@` indicating the beginning of a read sequence followed by a sequence identifier and (optionally) a description. This line is used to identify the sequence and distinguishing it from other sequences. The second line contains the actual sequence letters (bases). The third line begins with the symbol `+` and optionally followed by the same sequence identifier. It is not required to have anything after the symbol `+` but if present, it must be the sequence identifier. The fourth line contains the quality scores encoded for the sequence. This line should have the same length as the second line indicating a quality score value for each base in the sequence. Figure 3.7 shows an example of FASTQ format.

Quality score values (also known as a Phred or Q score) in FASTQ format are integer values representing the probability P that the corresponding base call is incorrect. They are generated based on a quality table (See Table 3.4) that uses a number of quality predictor values. The scores are encoded by adding 33 to the Phred score for Sanger and 64 for Illumina and then converted to an ASCII format.

Table 3.4: Quality scores and base calling accuracy.

Phred Quality Score	Probability of Incorrect Call	Call Accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10000	99.99%
50	1 in 100000	99.999%

identifiers, sequences letters, base call quality values and other meta information all together in one file. The nucleotide sequences and quality scores in SFF files can be extracted into two files (using Roche **sffinfo**), a sequence file in FASTA format (ends with .fna) and a quality file (ends with .qual). SOLiD produces data in color space and the primary output is generated in two separate files, a reads file in *csfasta* format (ends with .csfasta) and quality scores in a *quality* file (ends with .qual).

3.4.2 Sequence Alignment Format

Two sequence alignment data formats are typically used for NGS applications, (1) *Sequence Alignment/Map format (SAM)* [89] and (2) *Binary Alignment/Map (BAM)*, the compressed binary version of SAM format. SAM format has become the standard format for storing alignment data which is often converted into BAM format to allow more efficient storage.

3.4.2.1 Sequence Alignment/Map (SAM) Format

SAM format is a tab-delimited text format consists of two sections, an optional *header* section and an *alignment* section. All header section lines begin with the symbol "@", used to distinguish header lines from alignment lines, followed by two-letter record type code. Each line in the alignment section consists of 11 mandatory fields representing one alignment hit. Figure 3.8 shows an example of a SAM file.

Table 3.5: Alignment section fields in SAM format. Field 12 represents the optional fields.

Col.	Field	Description
1	QNAME	Query NAME (Query pair NAME if paired)
2	FLAG	Bitwise FLAG
3	RNAME	Reference sequence name
4	POS	1-based leftmost mapping POSition/coordinate of clipped sequence
5	MAPQ	MAPping quality (Phred-scaled)
6	CIGAR	Extended CIGAR string
7	MRNM	Mate reference sequence NaMe (“=” if the same as <RNAME>)
8	MPOS	1-based leftmost Mate POSition
9	ISIZE	Inferred insert SIZE
10	SEQ	Query SEQUENCE on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33 gives the Phred base quality)
12	OPT	Variable OPTional fields in the format TAG:VTYPE:VALUE

Table 3.6: The FLAG field in SAM format.

Flag	Decimal Value	Description
0X0001	1	The read is paired in sequencing
0X0002	2	The read is mapped in a proper pair
0X0004	4	The query sequence itself is unmapped
0X0008	8	The mate is unmapped
0X0010	16	Strand of the query (1 for reverse strand)
0X0020	32	Strand of the mate
0X0040	64	The read is the first read in a pair
0X0080	128	The read is the second read in a pair
0X0100	256	The alignment is nor primary
0X0200	512	The read fails platform/vendor quality checks
0X0400	1024	The read is either a PCR or an optical duplicate

any analysis, short read sequences need to be mapped back to a reference genome in order to determine the locations from which they originate. This process, referred to as a *reference-based assembly*, is performed assuming a reference genome is available.

It is the preferable approach since it is fast, easy, and capable of determining the locations of original sequence.

In cases where an organism does not have a reference genome, read sequences can be *de novo* assembled in order to construct the original reference sequence. Since read sequences are very short and large in number, this approach is more complicated and difficult than the previous one.

3.5.1 Alignment

Alignment is the process of mapping a DNA sequence to its reference sequence of origin to determine the most probable source location in the genome reference. It usually reports the most likely sequence of origin either as an identical sequence (100% match) or similar sequence (allowing a number of mismatch bases in the sequence alignment) [72]. Figure 3.9 shows the principle of sequence alignment (for illustration purposes, reads are represented by only 4bp length).

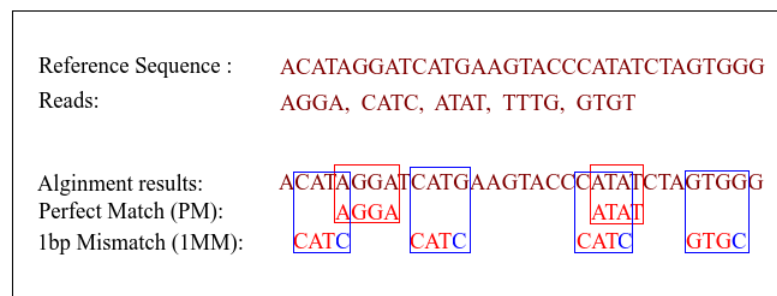


Figure 3.9: The concept of sequence alignment [72].

Several alignment algorithms have been developed in the last few years including the traditional alignment algorithms BLAST [1] and BLAT [75]. The main purpose of the BLAST and BLAT algorithms is to align DNA/protein sequences to a library or database of sequences to find shared sequence homology. The BLAST and BLAT tools tend to be efficient at aligning a small number of longer sequences. However, for large amounts of short reads, BLAST and BLAT can be very slow and

computationally expensive which make them impractical for mapping millions of short reads to a reference genome. Consequently, there was a critical need to develop new algorithms and tools to map NGS data in a fast and efficient manner. As a result, a number of short read alignment approaches have been developed, most of which are based on two approaches (explained below): (1) *Hash-Based* approach (also known as *Spaced-Seed Indexing*) and (2) *Burrows Wheeler Transform (BWT)* approach.

3.5.1.1 Hash-Based Approach

Most of the initially designed alignment algorithms are based on a *hash-table* approach [41]. The idea behind this method is to use a mapping function to map identified values called *keys* to their associated *values* through a special index. Alignment algorithms have implemented this type of data structure to first index the sequences and then associate them with read identifiers as shown in Figure 3.10.

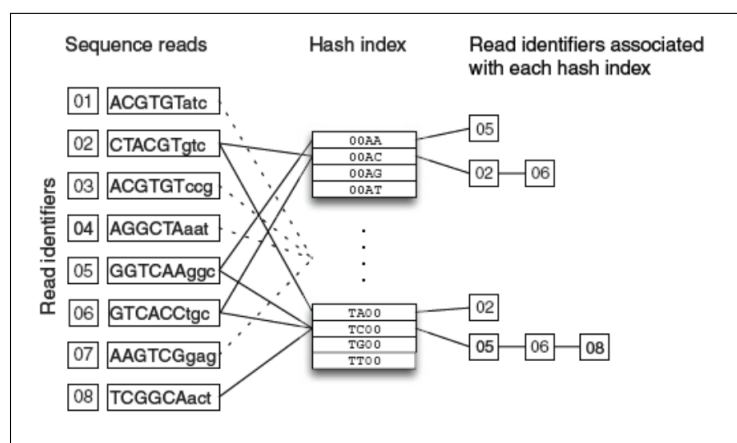


Figure 3.10: The hash-based approach [41].

In this approach, the hash table is constructed using either the sequences reads or the reference genome. If the table is constructed using the sequence reads, then the reference genome is used to search the table and vice versa for the second case. Examples of tools using this approach are shown in Table 3.7.

Table 3.7: Examples of hash-based aligners [41].

Tool	Hash-Table Construction Technique
MAQ [85]	Based on read sequences
ELAND (Illumina)	Based on read sequences
SOAP [84]	Based on reference genome
MOSAIC [63]	Based on reference genome
SHRiMP [135]	Based on read sequences
ZOOM [93]	Based on read sequences
BFAST [64]	Based on reference genome

The hash table in this approach is usually implemented as *spaced seeds* in which a read sequence is divided into four equal-sized subsequences called “seeds”. The idea behind this is that if a read, as a whole, can be perfectly mapped to the genome sequence, then all its seeds will mapped perfectly as well. On the other hand, if there were a one mismatch of mapping the entire read, this one base difference should fall within one of the seeds [155]. By using this technique to align the six possible pairs of the seeds to the genome reference, one can be sure all read hits with two mismatch are reported [41, 155]. Figure 3.11a shows the spaced seed indexing methodology.

3.5.1.2 Burrows-Wheeler Transform (BWT) Approach

The Burrows-Wheeler transform (also known as Block-Sorting) is a technique introduced by Michael Burrows and David Wheeler in 1994 for data compression. The idea of this approach is based on character rotations and sorting. The transformation of the input string is performed by rotating all string characters after appending a special character to the end of the string. This character should be smaller than all alphabets in the string. In the next step, resulted rotations are sorted in lexicographical order and the last column is taken as the output string, $\text{bwt}(s)$. Figure 3.12

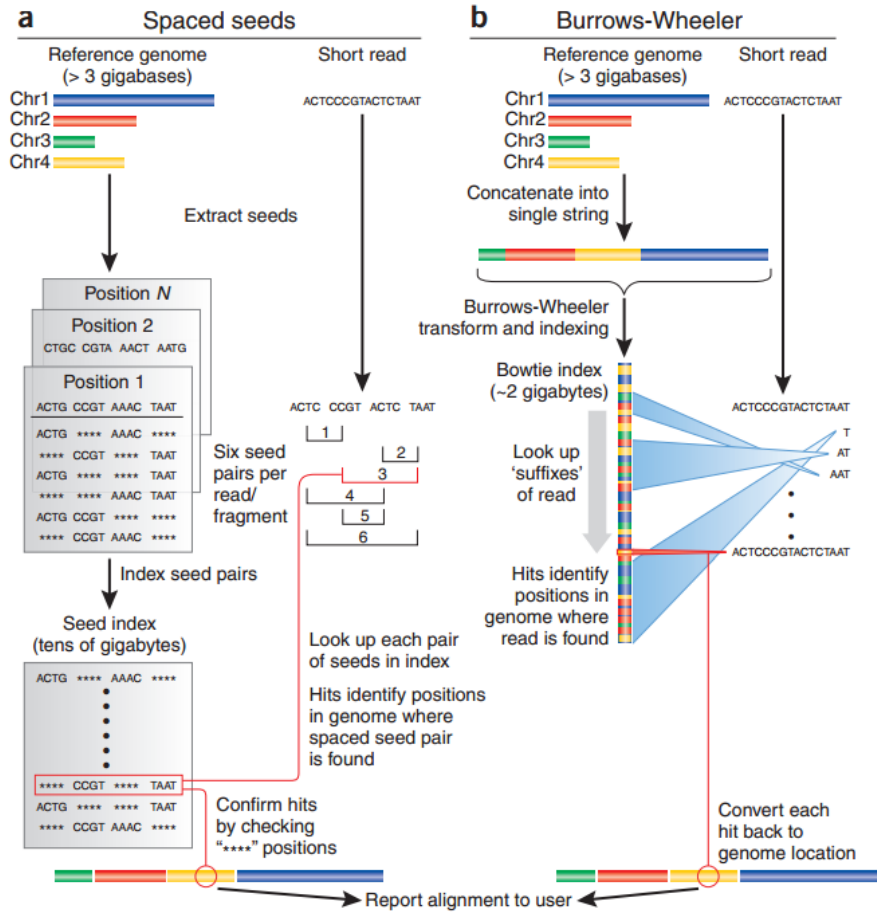


Figure 3.11: Genome alignment techniques [41]. **(a)** Spaced seed indexing method. Reference sequence positions are divided into equal-sized segments called “seeds”. Seeds are then paired and stored in a look-up table. Each read sequence is also divided into 4 segments and the seed pairs are used as keys to search for the matching positions in the reference sequence [155]. **(b)** Bowtie implementation of Burrows-Wheeler transform. Reads are aligned base by base from right to left and all active locations are reported. If no match position where the read might map is found, Bowtie backs up and make substitution.

shows an example of this process. For the human genome, this method can be used the same as in the example above. Figure 3.13 shows an example of creating a BWT of *14-mer* genomic sequence.

Input	All possible rotations	Sort rows in alphabetical order	Taking the last column	Output
T = mississippi	mississippi@ ississippi@m ssissippi@mi sissippi@mis issippi@miss ssippi@missi sippi@missis ippi@mississ ppi@mississi pi@mississip i@mississipp @mississippi	@mississippi i@mississipp ippi@mississ issippi@miss issippi@miss issippi@miss mississippi@ pi@mississip ppi@mississi sippi@missis sissippi@mis ssippi@missi ssissippi@mi	i p s s m @ p i s s i i	BWT(T) = ipssm@pissii

Figure 3.12: Burrows-Wheeler transform process. The Burrows-Wheeler matrix T is constructed as a matrix whose rows represent all possible rotations of T. The property of reversible permutation of BWT(T) allows the original text to be reconstructed. Note that the output string has many repeated characters which make it more easy to compress.

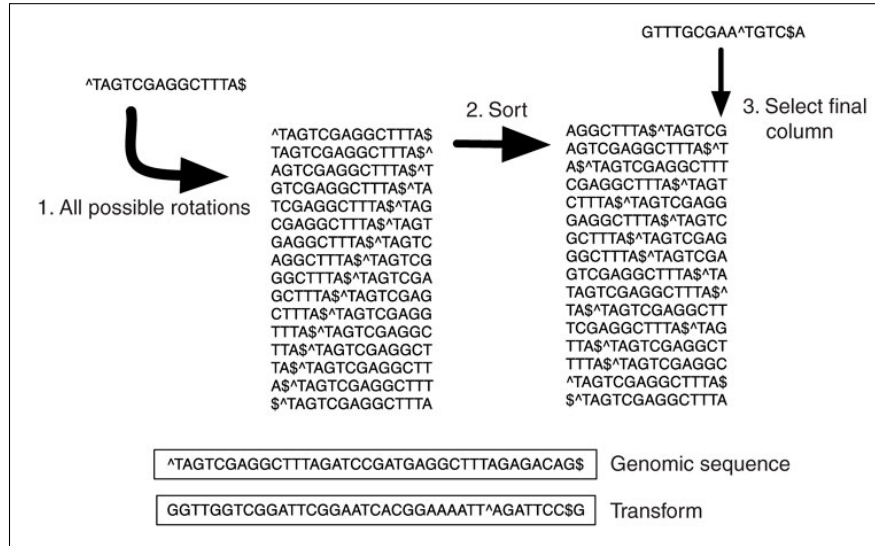


Figure 3.13: Burrows-Wheeler transform for genomic sequence data [41].

Examples of tools using this approach include BOWTIE [79], BWA [86], and SOAP2 [87]. These tools implement BWT technique using Ferragina-Manzini index (FM-index) [39] data structure to align read sequences to the reference sequence. As a fact, algorithms implementing BWT are much faster than hash-based algorithms.

Since Bowtie is considered the most widely used algorithm for aligning NGS data, a brief description of the tool is given below.

3.5.1.3 Bowtie

Bowtie [79], as defined by its developers, is an ultrafast, memory-efficient short read aligner that align short DNA sequences (reads) to large genomes. The implementation of Bowtie is based on a Burrows-Wheeler transform indexing schema shown in Figures 3.12 and 3.13. It uses an FM-index to build genome indices. Using this transform, the entire human genome can be compressed and indexed into about 2.2 gigabytes (Bowtie 1) of memory (3.2 gigabytes for Bowtie2). Bowtie has the ability to align 25 million of reads with length 35bp to the human reference genome in an approximately one CPU hour utilizing about 1.3 gigabytes of memory [79]. In order to find the exact match hits, Bowtie uses BWT with the Ferragina-Manzini (FM) exact-matching algorithm. Since this algorithm does not allow for mismatches and favors high quality reads, Bowtie has extended it by introducing a novel approach to the FM algorithm called *a quality-aware backtracking* [79]. To limit excessive backtracking, Bowtie has introduced another extension called *double indexing* [79].

Bowtie uses BWT to align reads base by base to the transformed reference genome starting from the end of the read (right to left) as shown in Figure 3.11b. When a read is traversed, all matched locations to which the read might align are reported. If no matched location is found in which the read might perfectly map, Bowtie backtracks to the previous base and incorporates a base and restarts the search [155]. Using this technique, Bowtie has proved to be one of the fastest alignment algorithms (faster than the hash-based algorithm of MAQ by 30-fold [41]).

3.5.2 Assembly

As discussed earlier, the reference-based assembly approach is the preferable choice if the reference genome exists. However, in cases where the reference genome is not available (*de novo* sequencing), the short DNA sequences (reads) must be assembled to computationally reconstruct the original DNA sequence. In general, assembly refers to the process of grouping short reads into *contigs* and contigs into *scaffolds* (Figure 3.14), without using any prior knowledge of the genome. Contigs and consensus sequences are built from multiple sequence alignment (overlapping between reads) of short reads with no gaps. Scaffolds (or supercontigs) refers to an ordered and oriented set of contigs separated by gaps. These gaps might be identified by one or more "N" where the consecutive number of N's determines the gap length [107]. The process of constructing scaffolds from contigs is called *scaffolding*.

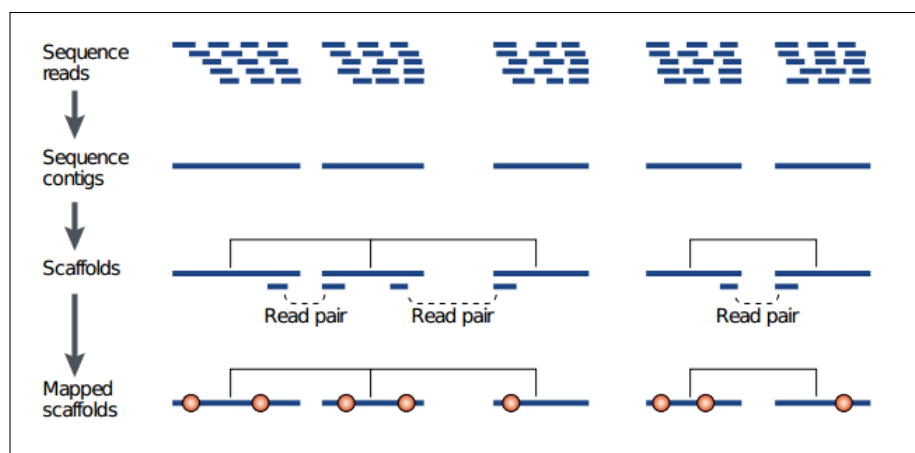


Figure 3.14: Overview of genome assembly [51].

Several graph-based assembly algorithms have been developed for NGS data which are classified into three main categories, (1) *Greedy graph assembly*, (2) *Overlap/layout/consensus (OLC)*, and (3) *De Bruijn graphs (DBG)* algorithms [107]. In order to discuss and understand these approaches, a brief review of graph theory concept should be given.

3.5.2.1 Graphs and Graph Theory

A *graph* is a data structure representation composed of a set of *nodes* (*vertices*) and a set of *edges* (*arcs*) between them. It is defined as $G = (V, E)$ where V is the set of nodes (vertices) and E is the set of edges (arcs). Each edge in the graph is connected by a pair of nodes u and v and $u, v \in V$. A graph is called a *directed* (or *digraph*) graph if the edges have a direction associated with them and *undirected* if the edges have no direction. The directed edge that points to a node at the end of the edge (sink node) is called the *incoming* edge for that node. Similarly, if the node is the source node of the directed edge, the edge is called *outgoing* edge for that node. The number of edges incident to a node v represents the degree of the node and defined as $deg(v)$. A sequence of directed edges e_1, e_2, \dots, e_n such that each node is adjacent to the next form a *path* P defined as $P = ((v_1, v_2), (v_2, v_3), \dots, (v_k, v_{k+1}))$. Thus, the number of edges in a graph path represents its length. A graph is called *connected* if each pair of nodes can be joined by a path and *disconnected* otherwise. If a path starts and ends with the same node, it is called a *graph cycle*. A *cyclic graph* is defined as a graph containing at least one *cycle* whereas *acyclic* graph is a graph that does not contain any cycle. A *directed* and *acyclic* graph is referred to as a *directed acyclic graph* (DAG).

The sequences of short reads can be represented as an overlap graph where nodes represent reads and edges represent the overlaps between reads [107]. Since edges represent the overlaps, each path in the graph will represent a contig. This can be performed using a *k-mer* graph (for a de Bruijn graph) such that nodes represent all equal-sized subsequences (*k-mers*) and edges represents the overlaps between the subsequences by $k-1$ bases (Figure 3.15b). Alternatively, nodes can be used to implicitly represent the overlap between the subsequences by $k-1$ bases and edges represent the subsequences [107] as shown in Figure 3.15c.

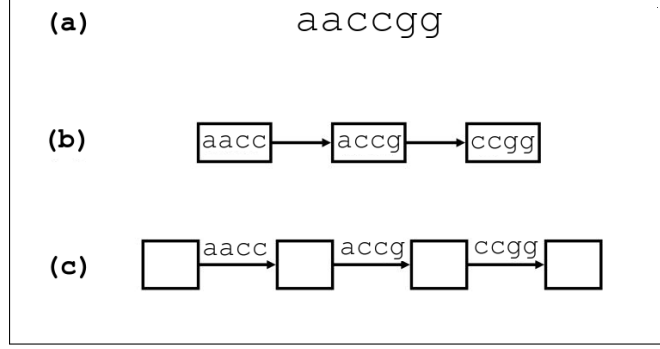


Figure 3.15: A k-mer graph representation of a read sequence with $k=4$ [107]. The path in the figure represents consensus sequence. The overlaps are computed by performing a pairwise alignment between all reads

3.5.2.2 Greedy Graph Assembly

Among the simplest types of assembly algorithms is the greedy graph assembly. In this approach, individual short reads are grouped together into contigs starting with one read (or contig) and continually adding more overlapping reads (or contigs) that have the best overlap until no more reads/contigs can be added [107, 122]. It is a *prefix-to-suffix* overlap which means the prefix of one read overlaps with a sufficient number of bases with suffix of another read. The best overlap in this context refers to the reads with the highest overlap score determined by the number of matched bases in the overlap. Examples of algorithms using greedy assembly strategy include SSAKE [161], SHARCGS [35], and VCAKE [71].

3.5.2.3 Overlap/Layout/Consensus (OLC)

The OLC approach uses the concept of an overlap graph in three steps, (1) *overlap*, (2) *layout*, and (3) *consensus*.

1. **Overlap:** in this step, pairwise overlaps between read sequences are discovered by comparing the reads to each other. This comparison is performed using a *heuristic seed* and *extend* approach to find a set of *k-mers* across all reads and

determine which reads share the k -mers. The set of k -mers will be then used as a *seed* for the alignment between reads [107].

2. **Layout:** in the layout step, the *overlap graph*, discussed in step (1), is constructed so that an approximate layout of the read sequences can be given. This graph is then analyzed to define the paths corresponding to fragments in the genome being assembled. The final aim of this analysis is to define a single path that includes all nodes such that a node is visited only one time. This path will correspond to the reconstruction of the genome using all read sequences [122].
3. **Consensus:** in the final step, consensus sequence is constructed using multiple sequence alignment (MSA).

The OLC approach often used for assemblies designed based on the old Sanger sequences. Examples of OLC algorithms are the Celera assembler [112], Arachne [8], and CAP3 [67].

3.5.2.4 De Bruijn Graph (DBG)

De Bruijn graph assembly (or the *Eulerian approach*) is the most widely adopted approach to assemble NGS short reads generated by Illumina and SOLiD. It is also based on the concept of k -mer graphs discussed earlier. Generally, De Bruijn graph is constructed as follows. To start, all reads are broken down into k -mers (sub-sequences of length k) in which each node in the graph represent a k -mer of length $k-1$ prefixes and suffixes of the original k -mer. Two nodes are connected by a directed edge if $(k-1)$ -suffix of the source node is a $(k-1)$ -prefix of the sink node resulting in an overlap of $k-2$ as shown in Figure 3.16.

The assembly problem in this context is equivalent to finding a path that includes all edges in the graph [122]. This path, by which the program Euler assembler

CHAPTER 4

RNA-SEQ METHODOLOGIES

4.1 Transcriptome Analysis

Transcriptome refers to the set of all transcripts and their abundance in the cell [159]. Generally, the term refers to the set of all RNA (mRNA, tRNA, rRNA, and non-coding RNA) transcripts in a particular cell. By studying transcriptomics, different genome-level functions can be identified, such as estimating when and where each gene is expressed in the cell/tissue at a given time, detecting the amount of mRNAs (RNA expression levels) in a particular cell (expression profiling), and discovering new genes. Thus, one important goal of analyzing the whole transcriptome is to define and catalogue the characteristics of all transcripts expressed in a particular cell (or tissue) for a specific developmental stage [27]. Different techniques have been developed over time for transcriptome analysis, starting with *Northern blot* analysis developed in 1977 leading to whole transcriptome sequencing with NGS in 2006. Figure 4.1 shows the developmental milestones of transcriptome analysis.

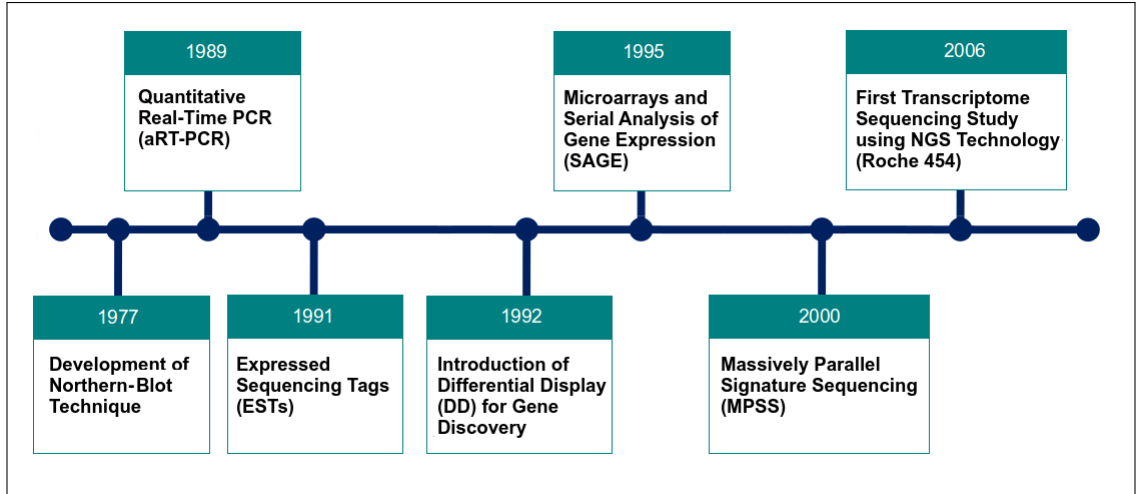


Figure 4.1: The developmental milestones of transcriptome analysis.

4.2 Transcriptome Analysis Techniques

4.2.1 Candidate Gene Approaches

4.2.1.1 Northern Blot

Northern blot was the first technique developed for transcriptome analysis by James Alwine, David Kemp, and George Stark in 1977 to detect specific RNA (or isolated mRNA) sequences for gene expression studies. The Northern blot procedure starts with the extraction of mRNA from tissue samples or cells which is then purified. The extracted mRNA is then size-separated by gel electrophoresis and separated RNA samples are transferred to a nylon membrane. The RNA is detected in the final step using an isotopic or non-isotopic labeled cDNA or RNA probe (Figure 4.2). The throughput of this technique is quite low (detection of a few known transcripts) while large amounts of input RNA is required [108].

4.2.1.2 Reverse Transcription Quantitative PCR (RT-qPCR)

Reverse transcription quantitative PCR (RT-qPCR) (also known as quantitative real-time PCR or qRT-PCR) is one of the most popular techniques for accurately

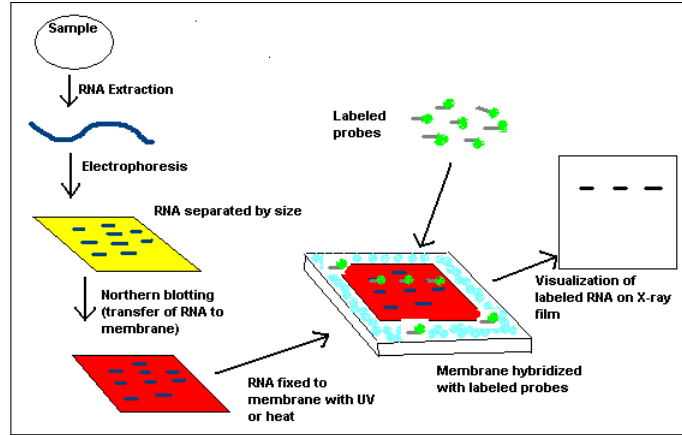


Figure 4.2: The Northern-blot procedure. [115].

measuring the mRNA level for a gene or locus proposed in 1989. It is used today as means of validating RNA expression for a limited number of transcripts. This technique requires the RNA to be converted into a more stable form called complementary DNA (cDNA) using an enzyme called reverse transcriptase. Then PCR amplification and probe hybridization is performed for the DNA molecule of interest. To quantify the amount of mRNA template in the sample, the probe needs to be fluorescently labeled and the emission of this fluorescent label is recorded at each PCR amplification step, allowing for a very accurate measurement of the original RNA. An important step before performing any sample-to-sample comparison is to normalize the output data generated from the RT-qPCR experiment. This technique has the advantage of increasing the throughput and reducing the required amount of input mRNA. However, RT-qPCR is not able to perform transcriptome-wide analysis, with a throughput ranging on the order of hundreds of transcripts at a time [108].

4.2.2 Sequencing-Based Approaches

4.2.2.1 Expressed Sequencing Tags (ESTs)

Expressed sequence tags (ESTs) are short DNA sequences (200 to 800 bases) generated by sequencing the complementary DNA (cDNA) and used to determine if a gene is expressed in a cell at a particular time. This process is performed by sequencing 200-500 nucleotides from one or both ends of each cDNA creating 5'ESTs and 3'ESTs (Figure 4.3). ESTs are then used to search genome databases (e.g. GenBank, EMBL, and DDBJ) to find a matching sequence. Since ESTs are sequenced from the transcribed regions, ESTs have been mainly used for discovering novel genes and coding regions.

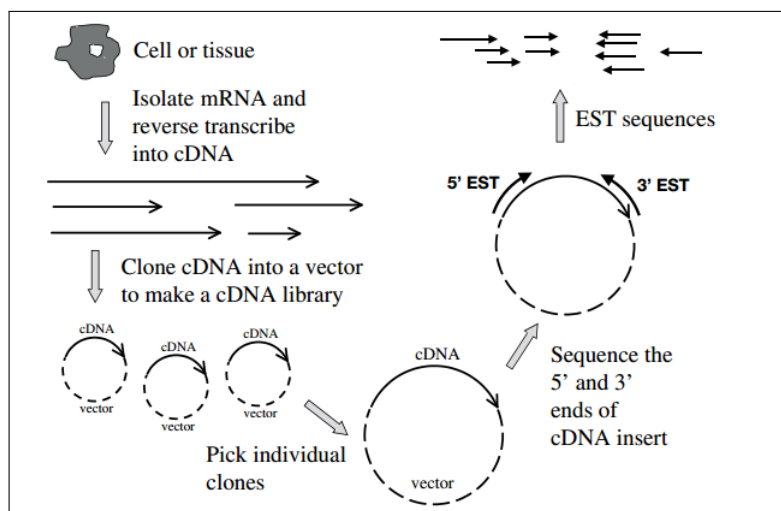


Figure 4.3: Overview of EST construction [9].

Since ESTs are a quick and inexpensive to construct, for a long time, this method was the core method for gene transcript discovery. However, ESTs can be error-prone, and do not typically cover the entirety of each transcript.

4.2.2.2 Serial Analysis of Gene Expression (SAGE)

The SAGE method, developed by Dr. Victor Velculescu in 1995, was the first tagged sequencing technique used for gene expression profiling. It was introduced as an alternative method to microarrays for the detection of differentially expressed genes by comparative analyses. This method was originally used for the investigation of differentially expressed genes in colon cancer. Thus, a large number of SAGE studies were focusing on cancer research. The general procedure of a SAGE experiment starts with the isolation of mRNA from the input sample. From each mRNA fragment, a small sequence (9-10 base pairs) called a *SAGE tag* is sequenced. These tags can be serially analyzed and linked together to form a long chain. To identify the abundance of each transcript, the number of times each a SAGE tag appears (called *SAGE tag number*) is recorded. Figure 4.4 shows an overview of SAGE method.

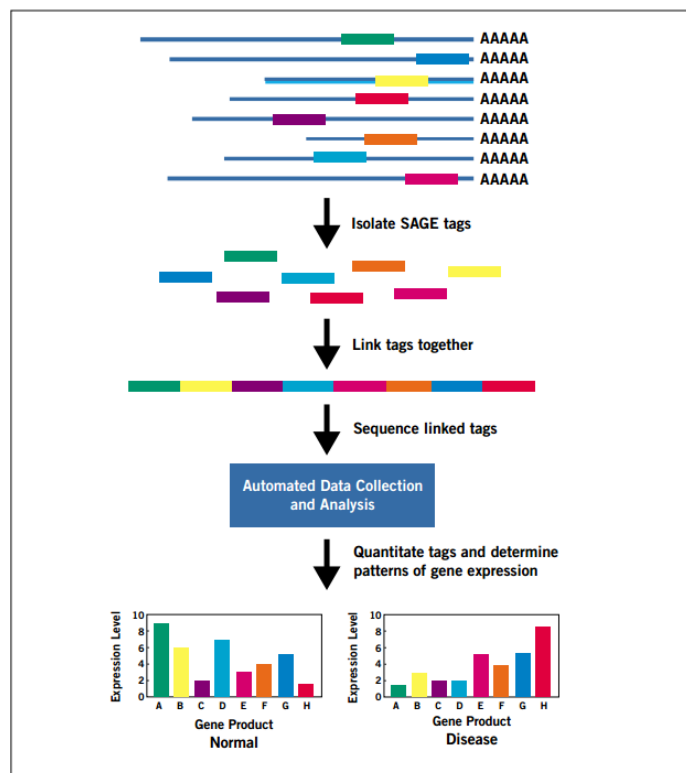


Figure 4.4: Overview of SAGE method [138].

Although SAGE can be superior to microarrays since it that does not require any prior knowledge of isolated genes and it produces digital counts of transcript abundance, the method had not been used as widely as microarrays [108].

4.2.2.3 Massively Parallel Signature Sequencing (MPSS)

MPSS was introduced by Sydney Brenner in 2000 for conducting in-depth expression profiling. It can be used to analyze the expression level of all genes in a sample by counting the number of individual mRNA molecules produced by each gene [127]. Similar to SAGE and unlike microarrays, MPSS does not require any prior knowledge of identified genes before performing an experiment. It generates digital data by counting all mRNA molecules in a sample. This process is performed through the generation of a 17-20 nucleotides signature sequence from each mRNA at a specific site upstream from its poly(A) tail [127]. This sequence is called a “signature” which is used to identify the mRNA molecule. Thus, measuring the expression level of any gene means counting the number of “signatures” for a gene’s mRNA.

4.2.3 Microarray Technology

Despite its limitations, microarray technology is the most widely used technique for transcriptome analysis [109]. It has dominated gene expression studies for the last 15 years. Microarray technology is a *hybridization-based* technique that allows simultaneously the analysis of hundreds of samples and measures the expression levels of tens of thousands of known genes. The microarray itself is made of a collection of microscopic DNA spots attached to a solid surface (usually glass or silicon). Each DNA spot on the array contains picomoles of a specific single-stranded DNA sequence or oligonucleotide called a *probe*. A single microarray chip can have hundreds of thousands of spots, each can contain millions of genomic DNA or short stretch of

oligo-nucleotide strands that correspond to a particular gene. Microarrays are used in several studies including *detecting and measuring gene expression* (the most popular use where the expression of a set of genes in one condition (e.g. diseased) is compared to the same set of genes in another condition (e.g. healthy)), *microarray mutation analysis* (mainly for SNP detection), and *comparative genomic hybridization*, which is used to assess genome content in different cells or closely related organisms.

There are two types of microarrays: (1) *Spotted arrays* (spotting the DNA onto the surface) and (2) *In-situ synthesised oligonucleotide arrays* where oligos are built up base-by-base on the surface. The DNA microarray can be classified in their structures into three types: (I) *Single channel arrays* (Affymetrix gene chips), (II) *Multiple channel* (dual color (cDNA) microarrays), and (III) *Specialty approaches* (Bead arrays such as Lynx, Illumina).

The general procedure of a microarray experiment, described in Figure 4.5, starts by extracting RNA molecules from the cell/tissue of interest. The extracted RNA molecules are then reverse-transcribed into cDNA using an enzyme called reverse transcriptase. The produced cDNAs are labeled with different fluorescent dyes, typically *Cy3* and *Cy5* (e.g. red for condition A and green for condition B) using a two-channel approach (i.e. Illumina) or biotin for single channel microarrays (i.e. Affymetrix). Once the cDNAs of the sample have been labeled, they are allowed to hybridize onto the the same glass slide. Performing this step will cause the cDNA sequence to hybridize to specific spots on the glass slide containing its complementary sequence. To remove any extra hybridization solution (unbound probe), a washing step is performed to make sure only the labeled target of interest is the actual one. Following hybridization, the spots are excited by a laser and scanned at appropriate wavelengths to detect the different dyes [50]. The detected fluorescence is stored as an image in a file (usually in tagged image format (.tiff)) for further analysis. The

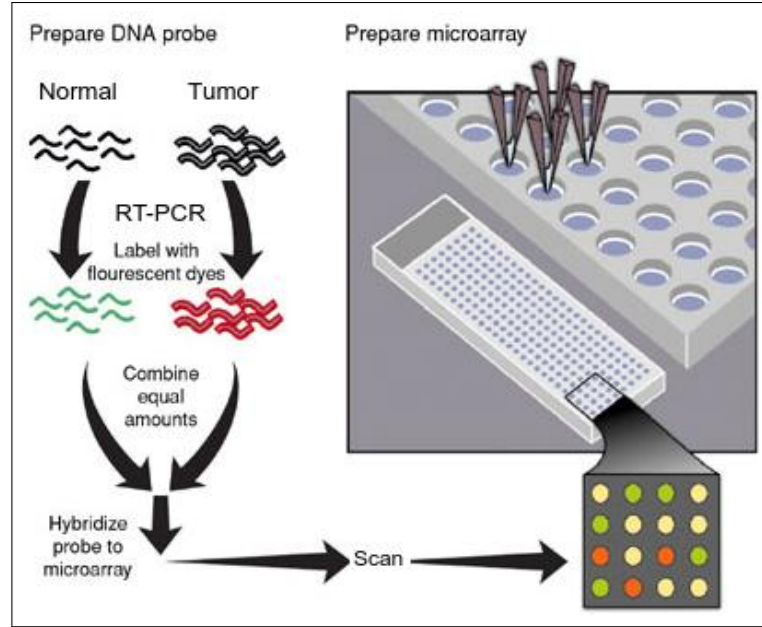


Figure 4.5: Microarray technology [81].

colors in the image represent the status of the gene in both conditions. Thus, when a gene is expressed abundantly in the condition of interest (e.g. diseased) but not in the control condition (e.g. healthy), the spot would appear as a red. In contrast, if a gene was expressed at a higher level in the control, the spot would appear as a green. In cases where a gene was expressed in both conditions, the spot would appear as a yellow and as a black if it was not. The produced image in the last step is processed and background and feature pixels are transformed into intensity values to quantify the spots. Intensity values are combined into unique quantitative measures reflecting the expression level of the gene deposited in a specific spot. In order to perform a comparative analysis, signal intensities need to be normalized. Once microarray data is normalized, various differential expression analysis methods can be applied to detect differentially expressed genes across conditions.

Despite its power of measuring the expression of thousands of genes, microarray technology suffers from a number of limitations including:

1. The technology requires the use of prior knowledge of the organism which make it unable to detect novel transcripts.
2. The dynamic range of intensity levels is limited by the resolution of the scanner used (typically 16-bits).
3. Since the data representing the expression level is derived from the hybridization intensity, this data is noisy. [108].
4. There exists a likelihood of a cross-hybridization between mRNA sequences and non-specific targets.

4.3 High-Throughput mRNA Sequencing (RNA-Seq)

The introduction of high-throughput sequencing technologies has revolutionized genome research in many areas including their applications to transcriptome profiling studies. *RNA-Seq* (or *Whole Transcriptome Shotgun Sequencing*) refers to the use of deep sequencing technologies for transcriptome analysis. The advent of RNA-Seq has enabled researchers and scientists to study the transcriptome at an unprecedented rate and has lately become the standard technology for transcriptomics. It is based on the direct sequencing of complementary DNA (cDNA) using next-generation sequencing technologies [109]. RNA-Seq is proving to be the technique of choice for studying transcriptome profiles offering several advantages over *hybridization-based* approaches such as microarrays (Table 4.1), by providing the ability to detect known and novel transcripts and to precisely measure transcript expression levels independent from any prior knowledge of the genome sequence. Unlike microarrays, which generate expression signal intensities, RNA-Seq generates quantitative expression read counts. Thus, increasing read counts provides higher dynamic ranges at higher resolution, which improves both sensitivity and quantitative accuracy. In addition, RNA-Seq

makes it possible to access some transcriptome structures such as allele-specific expression, novel promoters, isoforms [116], alternative spliced variants, and sequence variation (e.g. SNPs). RNA-Seq short reads (35-150 bases) provide information about how two exons are connected, whereas long reads are useful for determining how multiple exons are connected [159]. Unlike microarrays, RNA-Seq has a low background noise with high resolution. While microarrays offer resolution at the probe length, RNA-Seq allows for a single base resolution. Such granularity allows for better detection of splice variants. Furthermore, the ability to distinguish different isoforms and different allelic expression is limited in microarrays but is high in RNA-Seq [159]. Also, the dynamic range for quantifying expression differences is limited to a few hundred folds in microarrays, but can be nearly 10,000 fold with RNA-Seq data. One key limitation for microarrays is the dependency on a reference genome. Although RNA-Seq can take advantage of such an annotation, it also offers the ability for *de novo* transcriptomics.

Table 4.1: Advantages of RNA-Seq over microarray technology.

Application	RNA-Seq	Microarray
Data Type	Quantitative read counts	Relative intensities
Technology	High-throughput sequencing	Hybridization
Resolution	Single base	From several to 100bp
Genome references	Required in some cases	Required
Dynamic range	~10000-fold	Few hundred-fold
Background noise	Low	High
RNA amount required	Low	High
Alternative splicing/novel isoforms	Able to detect	limited
Discover new genes	Yes	No

4.3.1 RNA-Seq Workflow

The workflow of an RNA-Seq experiment is straightforward and generally starts with the extraction of total RNA or a polyadenylated RNA. The extracted RNAs or Poly(A) is then converted to a library of double stranded cDNA and sheared into small fragments. In the next step, adapters are attached to one or both sides of each cDNA fragment. Using next-generation sequencing platforms, each cDNA fragment is sequenced and a short sequence (read) from one end of the fragment (single-end tag) or from the two ends (paired-end tag) is obtained. Figure 4.6 shows the typical RNA-Seq workflow. The obtained reads are then mapped to the reference genome

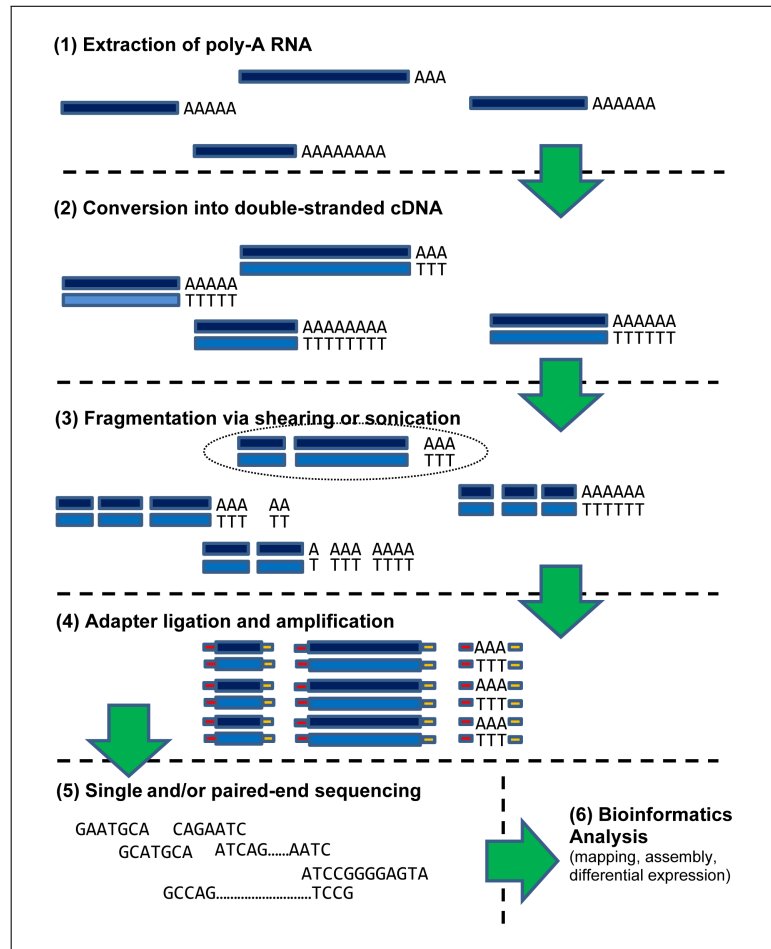


Figure 4.6: Workflow of RNA-Seq experiment.

to measure the abundance of transcripts. If the reference genome was not available, read sequences can be *de novo* assembled to construct the full set of transcripts and estimate their abundance.

4.3.2 RNA-Seq Applications

RNA-Seq methodology has been applied to a variety of applications including (I) transcriptome reconstruction, (II) transcript quantification, and (III) detection of significant changes in the transcript expression levels across biological conditions. In this section, a brief overview of each application is given.

4.3.2.1 Transcript Assembly

Transcriptome assembly, the foundation of transcriptome studies [101], is the process of identifying the complete set of transcripts in the transcriptome. To perform this task, RNA-Seq read sequences generated by NGS platforms need to be assembled prior any further analysis. Three main methods were identified for transcriptome assembly, (I) *reference-based assembly (or ab initio)*, (II) *de novo assembly*, and (III) *combined assembly*.

Reference-Based Assembly

Reconstructing the transcriptome in this method is built upon the available reference genome, where read sequences in the first step are aligned directly to the reference genome (Figure 4.7a) to determine their original locations using one of the alignment tools mentioned in section 3.5. In the second step, a graph representing all possible transcripts is built from the overlap between all reads (Figure 4.7b). Transcripts are constructed in the final step by traversing the graph and defining paths that, as a result, should correspond to transcripts. (Figure 4.7c) [101].

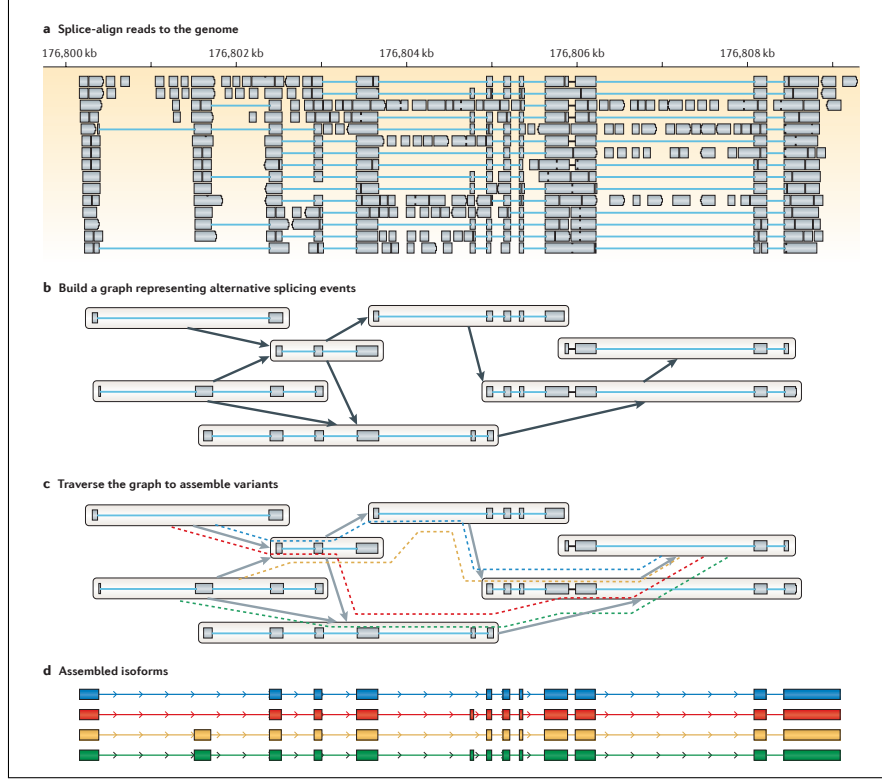


Figure 4.7: Overview of reference-based assembly method [101].

A variety of assembly algorithms have been developed using reference-based assembly including **G-Mo.R-Se** (Gene Modeling using RNA-Seq) [31], **Cufflinks** [154], and **Scripture** [55]. Whereas G-Mo.R-Se builds a *de novo* gene model based on an exon identification approach, Cufflinks and Scripture use the concept of a graph to assemble transcripts by using spliced reads (reads spanning exon-exon junctions). In order to construct transcripts, both Cufflinks and Scripture apply similar approaches to construct the graph, but differ in the traversing strategy. While Cufflinks constructs an overlap graph based on the spliced alignment locations of the reads, Scripture makes use of individual bases and all possible connections between them (graph topology) to construct the graph.

There are a number of advantages associated with *reference-based assembly* including its efficiency and sensitivity. Efficiency because assembly can be run on a

small RNAs using parallel computing and sensitivity because of the ability to assemble low abundance transcripts [101]. However, this method suffers from a few drawbacks. For example, the reference genome quality may have a big impact on the assembly process. Namely, if the reference genome contains a large number of mis-assemblies or deletions, the assembly process will result in a mis-assembled transcriptome. Another drawback is the errors resulting from the alignment process which will be carried over to the assembly process as well. Last, since this method requires an organism reference sequence, the method cannot be applied to organisms without a reference genome. As an alternative, one may use a closely related species but the limitation with this is that the assembly process will not be perfect and some transcripts from divergent regions will be missed.

***De novo* Transcript Assembly**

De novo assembly is a genome-independent method that does not require any predefined reference genome. Thus, instead of mapping reads to the reference genome, reads are directly used, based on their overlap, to construct transcripts. Assembly algorithms such as **Trans-ABYSS** [130], **Rnnotator** [102], **Multiple-k** [150], **Trinity** [49], and **Oases** [137] assemble transcripts based on the construction of a *De Bruijn* graph (discussed in section 3.5.2) but they differ slightly in the strategy for traversing the graph. Once the *De Bruijn* graph is built, paths are traversed and false branch points are trimmed resulting in paths that represent transcripts (Figure 4.8). The advantage of this method is that it does not require a reference genome which means it can be applied to organisms without a reference genome. Since it does use a reference sequence, this method is free from errors that may result from the alignment process. In addition, the *de novo* assembly approach can be applied even with the availability of a reference genome to reconstruct transcripts transcribed from missing

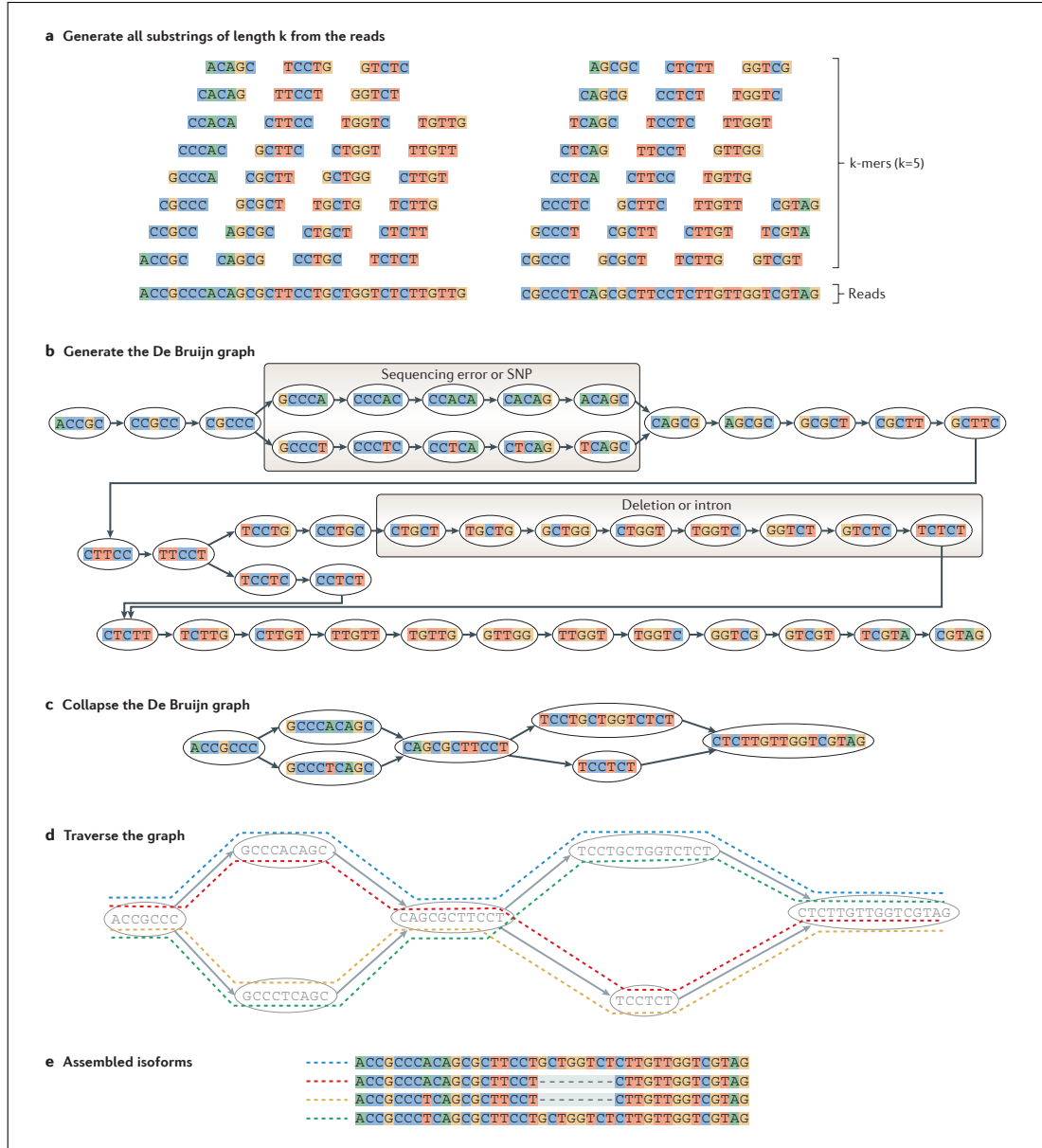


Figure 4.8: Overview of *de novo* transcript assembly [101]. **(a)** K -mers of length $k=5$ are generated from each read. **(b)** A *de Bruijn* graph is constructed where nodes represent k -mers and edges represent overlaps between them. **(c)** The *de Bruijn* is collapsed by merging adjacent nodes into a single node. **(d)** The graph is traversed and paths are defined where each path corresponds to a separate isoform. **(e)** Isoforms are then assembled.

regions in the genome assembly. However, as with any computational method, the *de novo* assembly has a few drawbacks. First, it is difficult to distinguish between sequence variations and sequencing errors which make it nontrivial to determine the trade-off between sensitivity and complexity [44]. Second, sequencing errors increase the complexity of the graph by producing branch points. Last, determining the k seed length can be an issue which may affect the assembly process. For instance, a smaller value of k will result in a large number of overlaps and therefore more complex graph and vice versa. Thus, choosing k in most cases will depend on the coverage. For example, a small value of k is preferable when the coverage is low since it increases the number of overlapping nodes to the graph.

The Combined Method

This method as it says from its name combines the two previous methods, *reference-based assembly* and *de novo transcript assembly* taking the advantages of both methods. Two main strategies are possible for this method, *align-then-assemble* or *assemble-then-align* as shown in Figure 4.9.

In the *align-then-assemble* approach, RNA-Seq reads are first aligned to the reference genome accounting for possible splicing events. Then transcripts are reconstructed from the spliced alignments. In the *assemble-then-align* approach in contrast, first transcripts are assembled directly from the RNA-Seq reads and splice-aligned to the genome to define exon and intron structure and variation between alternative spliced transcripts [56].

4.3.2.2 Transcript Quantification

Another application of RNA-Seq is the estimation of transcripts expression levels (relative mRNA quantities) at the gene and isoform levels. It is performed in

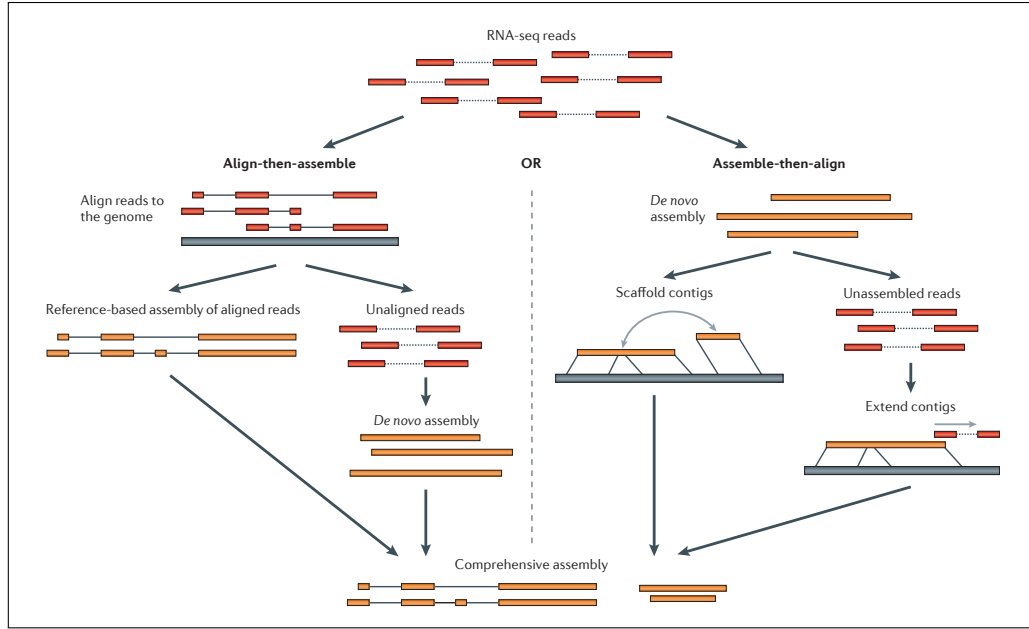


Figure 4.9: The combined method for transcriptome assembly. **(Left)** The *align-then-assemble* approach (e.g. Cufflinks and Scripture). **(Right)** The *assemble-then-align* approach (e.g. Trans-ABYSS, Trinity, Oases, and others) [101].

two steps, (1) aligning RNA-Seq reads to the reference genome and (2) measuring the abundances of genes and isoforms based on the read alignments generated in (1). Different methods have been developed for transcript quantification which can be categorized into two strategies, *count-based* methods and *isoform-expression* methods.

In count-based methods, all transcripts are assumed to have a single isoform and reads are mapped uniquely to the transcripts (which is not always the case). In its simplest form, *count-based* methods estimate isoform expression levels by counting the number of uniquely mapped reads to a single isoform. Although this strategy may work for some cases, it cannot be applied for genes with multiple isoforms. Thus, the count-based method is appropriate for single isoform genes such as *bacteria* in which alternative splicing does not occur [118]. However, due to alternative splicing events in *eukaryotic* species where most genes have multiple isoforms, reads may map to multiple isoforms resulting in an uncertainty of assigning reads to transcripts and

estimate their expression levels. Because of its estimation bias and incorrect estimation for alternative spliced genes [90], this method has a little use in the transcript quantification realm and as result more complex methods have been developed to handle those limitations.

The isoform-expression (or multi-read) methods have been developed to address the issue of reads that map ambiguously to multiple isoforms and genes. Several algorithms have implemented techniques such as generative models of RNA-Seq reads, Poisson models, quadratic programming, and expectation-maximization (EM) algorithms to estimate expression at both gene and isoform levels. Examples of these are Cufflinks[154], ERANGE [109], RSEM [90], MISO [74], IsoEM [114], and rQuant [12]. Most of these algorithms use a *likelihood function* to estimate isoform relative abundances. Maximum likelihood estimate (MLE) is the term used to describe the process of maximizing the likelihood function to infer isoform and gene expression levels. All transcript quantification algorithms need to normalize read counts prior to the quantification process in order to have accurate estimation results.

In their ERANGE package, Mortazavi et al. have proposed a multiread rescue method by initially estimating gene abundances from normalized counts of unique reads and use them to assign multireads to the most probable locations and re-estimate the abundances based on the counts generated after the assignment. The ERANGE reports transcript abundances in RPKM (Reads Per Kilobase of transcript per Million mapped reads) (See Section 4.3.2.4 for more details about RPKM). Cufflinks uses fractions of mapped reads to gene exons to estimate the relative expression after normalizing for gene length. It uses fragment counts instead of read counts to measure the abundance of transcripts by using FPKM (**F**ragment **P**er **K**ilobase of exon per **M**illion fragments mapped). To estimate the relative abundance of transcripts, Cufflinks uses a generative statistical model of RNA-Seq to derive the likeli-

hood for the abundances of a set of transcripts using a set of fragments. This model is used in cases where fragments map to multiple transcripts by allowing probabilistic deconvolution of RNA-Seq fragments densities [154].

Mixture-of-isoforms (MISO), a probabilistic framework, uses Bayesian inference to compute the probability that a read originated from a particular isoform. To measure the abundance of a set of isoforms, MISO treats isoforms as a variable and estimates a distribution over the values of this variable. The estimation process is performed based on sampling where a set of techniques called *Markov Chain Monte Carlo (MCMC)* is used. A more detailed description of the MISO framework can be found in [74]. RSEM and IsoEM use similar models based on the well-known *expectation-maximization* technique. RSEM does not require a reference genome, but it uses a set of reference transcript sequences instead. These transcripts will be preprocessed and used as a reference to which RNA-Seq reads will be aligned in order to estimate the expression levels of transcripts and their credibility intervals (CI). RSEM estimates maximum likelihood (ML) expression levels using expectation-maximization (EM) algorithm. In addition to the computation of ML, RSEM computes 95% credibility intervals and posterior mean estimate (PME) to measure the expression levels of each gene and isoform [90] (refer to [90] for more details about RSEM).

IsoEM is a novel expectation-maximization (EM) algorithm for isoform frequency estimation proposed by Nicolae et al. [114]. It takes the advantage of the information provided by the distribution of insert sizes generated during the process of library preparation. The **E-step** in IsoEM computes the expected number of reads $n(j)$ coming from isoform j with the assumption that isoform frequencies $f(j)$ is correct based on weights $w_{r,j}$ (refer to [114] for how these weights are computed). In the **M-step**, for each isoform j , a new value of the isoform frequency $f(j)$ is set to $c(j)/(c(1) + c(2) + \dots + c(N))$ where $c(j)$ denotes the normalized fragment coverage and

N is the set of isoforms. For a detailed description of IsoEM model, refer to [114].

rQuant has implemented a different technique based on solving quadratic programming and estimating different biases produced during the library preparation, sequencing, and read mapping. The basic idea of rQuant is to minimize the deviation of the observed read coverage from the expected coverage at each base by solving the following optimization problem:

$$(w_1, \dots, w_T) = \underset{w_1, \dots, w_T \geq 0}{\operatorname{argmin}} \sum_{p \in P} \left(C_p - \sum_{t=1}^T w_t D_{t,p} \right)^2$$

where T is the set of transcripts, w_1, \dots, w_T are the transcript abundance estimates, P is the set of genomic positions, C_p is the observed read coverage, and $D_{t,p}$ is the read density estimation for a transcript t at position p . If this model were used without considering bias estimation then $D_{t,p} = 1$ in case transcript t is exonic at position p and $D_{t,p} = 0$ otherwise. rQuant optimized the model by building a predictive model for the density and finding parameters θ in which resulting read densities fit properly to the observed read coverage. The optimized model is then defined as:

$$\theta = \underset{\theta}{\operatorname{argmin}} \sum_{l=1}^L \sum_{p \in P_l} \left(C_p - \sum_{t=1}^T w_t D_{t,p}(\theta) \right)^2 + R(\theta)$$

where L represents the number of loci, P_i denotes the set of positions for each locus, $D_{t,p}(\theta)$ is the θ is the read density parametrized for each transcript t at position p , and $R(\theta)$ is a regulatory term used to avoid model overfitting [12].

4.3.2.3 Differential Expression (DE) Analysis

One of the primary applications in RNA-Seq is the study of gene expression profiling across experimental conditions. The number of reads that map to a gene is a direct measure of its expression at the transcription level. Thus, the study of determining which genes have changed significantly in terms of their RNA expression across biological samples is referred to as differential expression analysis. This step

is essential in most RNA-Seq studies. Identifying which genes are differentially expressed (DE) between samples help researchers understand the functions of genes in response to a given condition. In this section, we review the most recently developed and widely used methods for differential expression analysis. We look at the different statistical models each method uses to test for differential expression. Since a large number of methods and tools have been developed in the last few years for DE analysis, not all DE methods are discussed here, but instead, we put more emphasis on the most widely used methods including DESeq [160], edgeR [132], DESeq [3], baySeq [58], and Cuffdiff [154]. A comprehensive list of the DE methods can be found in Table 4.2.

The detection of which genes have significant DE across samples requires the use of statistical hypothesis tests to model RNA-Seq count data. For any DE analysis, three components should be considered: (1) normalization of read counts, (2) statistical modeling of gene expression, and (3) testing for differential expression.

4.3.2.4 Normalization

In order to derive an accurate comparison within and between samples, normalization is performed on read counts to adjust for sequencing depth variations and other systematic technical variations which results in a comparable data across conditions. Thus, to discover significant changes in expression, studies have shown that normalization is an essential step in the analysis of differential expression. Several normalization techniques have been proposed in the literature. Marioni et al. [100] use the total read count (TC) to normalize read counts. This normalization method divides transcript read count by the total number of reads as follows: $\frac{X_{ij}}{N_j}$, where X_{ij} is the number of reads for gene i in sample j and N_j is the number of reads in sample j (library size). Such an approach is equivalent to the total intensity normalization

Table 4.2: List of common differential expression Analysis methods.

Method	Technique
DEGseq [160]	MA-plots based method, assuming normal distribution for $M A$.
edgeR [132]	Exact test based on NB distribution.
DESeq [3]	Exact test based on NB distribution.
baySeq [58]	Empirical Bayesian method (compute posterior probabilities of models, based on Poisson or NB distribution).
Cuffdiff [154]	NB distribution to model the variance in fragment counts.
LRT [100]	Likelihood ratio test based on Poisson model.
PoissonSeq [91]	R package based on Poisson log-linear model.
GPseq [146]	Likelihood ratio test for two-parameter generalized Poisson model.
NOISeq [151]	Empirical approach to model the noise distribution of DE by contrasting fold-change differences (M) and absolute expression differences (D) for all the features in samples within the same condition.
EBSeq [82]	Empirical Bayesian approach that models a number of features observed in RNA-seq data.
SAMseq [92]	Nonparametric approach for identifying DE in RNA-Seq data.
npSeq [92]	Nonparametric approach for identifying DE in RNA-Seq data. Similar to SAMseq with only difference that npSeq uses symmetric cutoffs, while SAM uses asymmetric cutoffs.
NBPSeq [32]	Negative Binomial (NB) models for two-group comparisons and regression inferences from RNA-sequencing data.
ShrinkSeq [157]	Bayes-empirical Bayes method that analyzes RNA-Seq data by estimating multiple shrinkage priors. It supports a variety of count models such as NB mode.
TSPM [7]	A Two-Stage Poisson Model for testing RNA-Seq data.
Limma [144]	An R package that uses linear models for the analysis of gene expression data arising from microarray or RNA-Seq technologies.
Alexa-Seq [52]	A method to analyze RNA-Seq data to catalog transcripts and assess differential and alternative expression of known and predicted mRNA isoforms in cells and tissues.
ASC [163]	Empirical Bayes method to detect differential expression.
BBSeq [170]	A method designed for the DE analysis of the RNA-Seq count data. The method incorporates two approaches: (1) a simple beta-binomial generalized linear model, (2) mean-overdispersion model used to capture the gene specific dispersion.
DiffSplice [66]	An ab initio method for the detection of DE alternative splicing isoforms under different conditions using RNA-seq reads.
QuasiSeq [98]	An R package used to apply the QL (quasi-likelihood), QLShrink and QLSpline methods to quasi-Poisson or quasi-negative binomial models for identifying DEGs in RNA-seq data.
BitSeq [47]	A Bayesian approach for estimation of transcript expression level from RNA-seq experiments and estimating differential expression (DE) between conditions.
MATS [140]	A Bayesian framework for flexible detection of differential alternative splicing from RNA-Seq data.
Myrna [80]	A cloud computing tool for calculating differential gene expression in large RNA-seq datasets. It includes short read alignment with interval calculations, normalization, aggregation and statistical modeling. It uses both parametric and non-parametric tests.
CEDER [158]	An R package developed to detect DEGs using RNA-Seq by combining significance of exons within a gene.
DEXSeq [4]	An R package that finds differential exon usage based on RNA-Seq exon counts. It uses GLMs of the NB distribution (NB-GLMs) to model exon counts.
SplicingCompass [6]	A method to predict genes that are differentially spliced between two different conditions using RNA-seq data. It uses geometric angles between the high dimensional vectors of exon read counts.
MISO [74]	A probabilistic framework that quantifies the expression level of alternatively spliced genes from RNA-Seq data, and identifies differentially regulated isoforms or exons across samples.

procedure applied for microarrays. Similar to the TC method, Bullard et al. [16] proposed a quantile normalization, borrowed from microarray technology, in which the total counts is replaced by the upper quantile (UQ) of the counts. The main concept of quantile normalization is to match the distribution of read counts in each lane to a reference distribution defined in terms of median counts across sorted lanes. Replacing the UQ by the median, another form of quantile-based normalization called a median normalization is used. To correct for differences in library sizes and gene length, Mortazavi et al. [109] introduced RPKM (Reads Per Kilobase of transcript per Million mapped reads). The RPKM is defined as: $RPKM = 10^9 \times \frac{C_g}{l(t) \times N}$, where C_g is the number of reads mapped to gene g , $l(t)$ is the length of transcript t , and N is the total number of mappable reads in the sample. There are two cases in this context to consider. In the first case when DE analysis is used to compare genes within a sample (each gene is compared relative to other genes in the sample), the length of the gene is important and should be considered for normalization to avoid bias. This is clear since longer transcripts will by their nature have more read counts. In this case, read counts should be normalized by gene length. RPKM has been widely used to normalize read counts using both the library size and the gene length. In the second case when DE analysis is applied to compare the expression of the same genes in different samples, the gene length is not considered in the normalization procedure. This is also clear since genes have the same lengths across samples.

As an alternative to RPKM, Transcripts Per Million (TPM) [109] procedure normalizes RNA-Seq data by dividing the number of reads of a transcript by the total clone count of the sample multiplied by 10^6 . Results using this method are reported as reads/TPM for each sample. One of the limitations of TPM is the inability to handle datasets marked with different RNA composition. Thus, another method

called Trimmed Mean of M-values (TMM) was proposed by Robinson et al. [133] as an attempt to remove RNA compositional bias. By estimating the relative RNA production levels, TMM equates the overall expression levels of genes between samples under the assumption that a large number of the genes are not differentially expressed. To calculate the normalization scaling factor, this method uses a weighted trimmed mean of the log ratios between two samples [133].

DE methods use different normalization procedures, some of which have improved the procedures discussed above. For example, Marioni et al. use the TC method; DESeq provides three choices for normalization, 'none', 'median', and 'loess' (loess regression); Mortazavi et al. use RPKM; and Trapnell et al. implement a slightly modified version of RPKM called FPKM (Fragments Per Kilobase of exon per Million mapped fragments) in their Cuffdiff method. R Bioconductor packages such as edgeR, DESeq, and baySeq use different normalization approaches as well. Whereas DESeq and baySeq use the library size, edgeR implements the TMM method. DESeq uses the median of scaled counts (similar to the quantile normalization) to estimate the normalization [78]. For each sample, the DESeq scaling factor is computed for each gene as the median of the ratio of its read count over its geometric mean across all samples [3, 126]. Using the assumption that most genes are not DE, DESeq uses the median of ratios associated with each sample to obtain the scaling factor. NOISeq, proposed by Tarazona et al. [151], uses several options for normalization including TMM, RPKM, and UQ. Limma (Linear Models for Microarray Data) [144], an R package designed initially for DE analysis of microarray data but lately adapted for RNA-Seq data, implements a quantile normalization approach. EBSeq [82] provides two choices for normalization, either by using the median of scaled counts (used in DESeq) or a quantile normalization approach. PoissonSeq [91] uses a normalization procedure which assumes a Poisson model for the data.

4.3.2.5 Statistical Modeling of Gene Expression

The detection of which genes have changed significantly between biological samples requires the use of statistical hypothesis tests to model count data from RNA-Seq experiments. Currently, most statistical models are based on parametric assumptions for modeling RNA-Seq data. Discrete probability distributions such as *binomial*, *Poisson*, and *negative binomial (NB)* distributions have been used to model RNA-Seq count data [78]. In RNA-Seq studies using a single source of RNA, the distribution of counts across technical replicates for the majority of genes was indeed Poisson [116, 78] in the form of $f(n, y) = \frac{(\lambda^n e^{-\lambda})}{n!}$, where n is the number of read counts and λ is the expected number of reads in each transcript [126]. Early methods such as the Likelihood ratio test proposed by Marioni et al. [100], DESeq [160], PoissionSeq, and Gpseq [146] have been developed to detect differentially expressed genes based on this distribution. However, since the variance in this distribution is equal to the mean, it suffers from the inability to capture biological variability within RNA-Seq data [116, 78]. Given the fact that the variance of many genes is likely to exceed the mean resulting in *over-dispersion*, Poisson-based analyses using biological replicates will be prone to high false positive rates and therefore this distribution will be impractical in this situation.

To address *over-dispersion* and account for biological variability, methods such as edgeR, DESeq, baySeq, and Cuffdiff have been developed based on the negative binomial distribution (NB) to model read counts. These methods address *over-dispersion* by defining the relationship between the variance v and mean μ . For example, edgeR and DESeq define this relationship as $v = \mu + \alpha\mu^2$, where α is the dispersion factor. edgeR provides two options for α , a common dispersion (estimated from all genes) and tagwise dispersion (estimated for individual genes) [132, 78, 42]. DESeq on the other hand estimates the dispersion parameter by using a combination

of two terms for the variance, one to estimate the Poisson (the mean expression μ) and the second is the raw variance of the gene used to model the biological expression variability [3, 126]. Cuffdiff computes two variance models, i.e., one for single-isoform genes and one for multi-isoform genes. For single-isoform genes, Cuffdiff computes the expression variance similar to DESeq using NB distribution. When a gene has multiple isoforms, Cuffdiff models over-dispersion by using the beta negative binomial distribution [154]. BaySeq differs from the above three methods and implements an empirical Bayesian model based on NB distribution. This model estimates the prior probability parameters by bootstrapping from the data and then applies the maximum likelihood method. PoissonSeq models RNA-Seq count data by using a Poisson log-linear model. The mean μ_{ij} in this model is defined as a log-linear model $\log \mu_{ij} = \log d_i + \log \beta_j + \gamma_j \gamma_i$, where d_i is the library size of sample i , β_j is the expression level of gene j , and γ_j is the correlation of gene j with condition γ_i [100, 91, 126]. $\gamma_j = 0$ if there is no association between gene j and γ_i , and $\gamma_j \neq 0$ otherwise.

4.3.2.6 Testing for Differential Expression

Once the parameters are estimated, statistical tests such as t-test, Wilcoxon test, or Fisher's exact test (FET) can be applied on the normalized data to detect significant differentially expressed genes between samples. Both DESeq and edgeR use a variation of the FET adopted for a negative binomial distribution. Cuffdiff compares the log ratio of gene expression in two conditions against the log ratio of one condition and calculates the test statistics as $T = \frac{E[\log(y)]}{\text{Var}[\log(y)]}$, where y is the log ratio of the normalized counts between the two conditions ($Y = \frac{FPKM_a}{FPKM_b}$). baySeq employs an empirical Bayesian approach to determine DE between conditions. For every gene, baySeq estimates two models, one model assumes the expression pattern is the same and the second assumes the expression pattern is different across conditions. Thus,

the posterior likelihood can be estimated using the prior estimates and the likelihood of the distribution of the data to decide if a gene is differentially expressed. PoissonSeq tests for DE by determining the significance of the correlation term Y_j in the linear model using a score statistic [126, 91]. The p-value is then derived using a *chi*-square distribution since the score statistic is shown to follow this distribution. Other DE methods use different statistical tests to test for DEGs. For example, limma uses a moderated t-statistic test to derive the p-value.

4.3.2.7 Differential Expression Analysis Methods

In this section, a number of the most recently developed and widely used methods for differential expression analysis are discussed as a related work of our approach.

Cuffdiff

Cuffdiff is a Cufflinks module that aims to find significant changes in transcript expression, splicing, coding output, and promoter use. It uses the Cufflinks transcript quantification module to calculate transcript/gene expression levels and tests for significant changes. The main input of Cuffdiff is the reference transcripts as a Gene Transfer Format (GTF) file and two or more SAM (Sequence Alignment/Map) or BAM (binary version of SAM) files containing fragment alignments for two or more samples. The output of Cuffdiff is a set of several files containing changes in expression at the level of isoforms, primary transcripts, and genes. To test for DE, Cuffdiff compares the log ratio of gene expression in two conditions against the log ratio of one and calculates the test statistics. This ratio requires the knowledge of the variance of the expression level in each condition which is calculated for a transcript's expression levels as follows:

$$Var[FPKM_t] = \left(\frac{10^9}{\bar{l}(t)M} \right)^2 (Var[X_t])$$

where $Var[X_t]$ is the variance in the number of fragments coming from the transcripts across replicates. Cuffdiff uses the negative binomial distribution (NB) to model the variance in fragment counts across replicates and the square root of the Jensen-Shannon (JS) divergence to quantify the changes in relative abundance. Thus, if we have abundances p_1, p_2, \dots, p_n , then the entropy of the discrete distribution p is defined as:

$$H(p) = - \sum_{i=1}^n p_i \log p_i$$

and the JS divergence between a set of m distributions p^1, p^2, \dots, p^m is defined as:

$$JS(p^1, \dots, p^m) = H\left(\frac{p^1 + \dots + p^m}{m}\right) - \frac{\sum_{j=1}^m H(p^j)}{m}$$

Based on this JS divergence, Cuffdiff assigns p -values to the observed changes.

edgeR

The R Bioconductor package, edgeR was initially developed for SAGE but since the methods are applicable to RNA-Seq, it has been also used for detecting differential expression in RNA-Seq data. The edgeR is based on the negative binomial distribution (NB) if data are over-dispersed. However, in cases where there is no over-dispersion, the Poisson model is used. The edgeR count model is defined as: $Y_{gij} \sim NB(M_j p_{gi}, \phi_g)$, where Y_{gij} represents the observed data for gene g in sample j and experimental group i . The parameter M_j denotes the total number of reads in a sample (library size) whereas the parameter p_{gi} represents the relative abundance of gene g in group i . ϕ_g is the dispersion parameter. In the case of over-dispersion, the NB model is parameterized with the mean $\mu_{gi} = M_j p_{gi}$ and variance $v = \mu_{gi} + \mu_{gi}^2 \phi_g$. However, in the case of no over-dispersion ($\phi_g = 0$), the NB model is reduced to Poisson model. The main input to edgeR is a table of counts constructed as a matrix

whose rows represent biological feature (e.g. genes, transcripts, or exons) and columns represents different samples. The output is a list of differentially expressed genes.

DEGSeq

DEGSeq is another R Bioconductor package developed for RNA-Seq data. The statistical model this package uses is based on a Poisson distribution. Two novel methods have been proposed in this package, an MA-plot-based method with random sampling and an MA-plot-based method with technical replicates where M is the log ratio of the counts between two conditions for gene g and A is the average of the log concentration of the gene in the two groups [42]. Along with those two methods, three existing methods, Fisher's exact test (FET), likelihood ratio test (LRT), and samWrapper have been integrated into DEGSeq to identify differential expressed genes. In the MA random sampling, RNA sequencing can be modeled as a random sampling process where each read is sampled independently and uniformly from every possible nucleotide in the sample. Thus, the number of reads coming from a gene/transcript follows a binomial distribution, which can be approximated by a Poisson distribution. With this assumption, DEGSeq is not applicable to data with over-dispersion which limits its use for RNA-Seq analysis. The input of this package is uniquely mapped reads, a gene annotation of the corresponding genome, and gene expression counts for each sample. The output includes a text file containing the gene expression values for the samples, p -values, and two kinds of q -values (adjusted p -values) and an XHTML summary page.

DESeq

DESeq is an R Bioconductor package that analyzes RNA-Seq count data using the negative binomial distribution and an estimator of the distribution's variance.

DESeq uses a similar statistical model to edgeR with a few extensions allowing for more general data-driven relationships of variance and mean. Under the assumption of a locally linear relationship between variance and mean expression levels, the variance can be estimated using data with similar expression levels [78]. The input of DESeq is a table of count data that reports for each sample the number of reads that have been assigned to a gene. Thus, a table cell in the i -th and j -th column represents the number of reads mapped to gene i in sample j . The output is a list of differentially expressed genes with p -values and q -values. The NB distribution DESeq uses to model count data is defined as: $K_{ij} \sim (\mu_{ij}, \sigma_{ij}^2)$, where K_{ij} denotes the read counts for gene i in sample j . This model has two parameters, the mean μ_{ij} and the variance σ_{ij}^2 . These two parameters are often not known in advance and therefore have to be estimated from the data. The mean μ_{ij} can be defined as: $\mu_{ij} = q_{i,\rho(j)} s_j$, which is the product of the expected read count (per gene and condition) $q_{i,\rho(j)}$ and size factor s_j which represents the coverage of library j . $\rho(j)$ is the experimental condition of sample j . In contrast, the variance is defined as:

$$\sigma^2 = \mu_{ij} + \underbrace{s_j^2 \cdot v_{i,\rho(j)}}_{\text{raw variance}}$$

where $v_{i,\rho(j)}$ is the per gene raw variance parameter. This parameter is assumed to be a smooth function of $q_{i,\rho}$ and defined as: $v_{i,\rho(j)} = v_p(q_{i,\rho(j)})$, which should allow the pooling of data from genes with similar expression strength. To perform testing, DESeq uses Fisher's exact test (FET) on NB data. Thus, for two conditions A and B, the *null hypothesis* for a given gene is that the counts of the two conditions are equal ($q_{iA} = q_{iB}$). The test statistic is performed using FET and the p -values computed using the following formula:

$$p_i = \frac{\sum_{\substack{a+b=k_{is} \\ p(a,b) \leq p(k_{iA}, k_{iB})}} p(a, b)}{\sum_{a+b=k_{is}} p(a, b)}$$

where k_{iA} and k_{iB} are the total read counts in each condition and $k_{is} = k_{iA} + k_{iB}$. Variables a and b denote the even probabilities for any pair of numbers a and b . For more details about the computation methods of the above model, refer to DESeq in the work of Anders and Huber [3].

baySeq

baySeq is an R Bioconductor package that assumes the data follows a negative binomial distribution. baySeq differs from the above two packages in the strategy of estimating significance by employing an empirical Bayesian approach to determine differential expression across conditions. The baySeq approach starts by first bootstrapping to estimate prior parameters from the data and then assessing posterior likelihoods of the models by applying either maximum likelihood or quasi-likelihood methods [42]. In general, the baySeq approach aims to identify the behavior of samples in terms of similarity and difference for each given model. Thus, for each gene there will be two hypotheses either the expression pattern is the same or different between two conditions. Under those two hypotheses, the posterior likelihood can be estimated using the prior estimates and the likelihood of the distribution of the data to decide if a gene is differentially expressed. The statistical models of baySeq are based on both Poisson and NB distributions. The Poisson distribution is defined as $Y_{gij} \sim (M_j p_{gi})$ assuming that the prior p_{gi} follows a gamma distribution $p_{gi} \sim \Gamma(\alpha_{gi}, \beta_{gi})$. The second model which is based on NB distribution is defined as $Y_{gij} \sim NB(M_j p_{gi}, \phi_g)$. The baySeq package accepts the table of read counts (similar to DESeq, DEGSeq, and edgeR) assigned to each gene for each sample as an input and reports a list of differentially expressed genes as an output.

CHAPTER 5

IBSEQ: AN ISLAND-BASED APPROACH FOR RNA-SEQ DIFFERENTIAL EXPRESSION ANALYSIS

5.1 Introduction

As discussed in Chapter 4 Section 4.3.2.3, the main application of RNA-Seq is the study of which genetic features are significantly differentially expressed across biological samples. It has been the most extensively investigated application for RNA-Seq studies. Uncovering which features are significantly differentially expressed between samples can provide insight into their functions. With the large magnitude of data generated by next-generation sequencing technologies, a significant effort has been made during the past few years to develop computational approaches that can accurately and quickly detect the significant change in expression across samples. The majority of the developed methods have been designed based on parametric statistics in which discrete probability distributions such as *binomial*, *Poisson*, and *negative binomial* are used. Table 5.1 shows examples of the current differential expression methods along with their statistical models. Refer to Sections 4.3.2.5-4.3.2.7 for more details about these methods.

One major limitation with the majority of these methods is they rely on genomic annotation. Thus, in order to detect which features are DE between samples, these methods usually require an annotation file (e.g. GTF/GFF, BED, or count table). The major drawback with this limitation is that any reads aligned outside

Table 5.1: Examples of current differential expression analysis methods.

Method	Statistical Model
Cuffdiff	Negative binomial distribution to model the variance in fragment counts
edgeR	Exact test based on negative binomial distribution
DESeq	Exact test based on negative binomial distribution
LRT	Likelihood ratio test based on Poisson model
Gpseq	Likelihood ratio test for two-parameter generalized Poisson model
DEGseq	MA-plots based methods, assuming normal distribution for $M A$
baySeq	Empirical Bayesian to compute posterior probabilities of models, based on Poisson or negative binomial data distribution

the annotated genome will be discarded and any significant change occurring outside annotated regions will not be captured.

In this Chapter, a novel Island-Based approach, IBSeq, is presented as an attempt to alleviate the issues resulting from relying on genomic annotation.

5.2 IBSeq Overview

In an attempt to overcome the limitation mentioned above and detect expression differences in any genomic region regardless of whether a genomic annotation is available, we developed a novel Island-Based approach, IBSeq. The general workflow of this approach is shown in Figure 5.1 while Figure 5.2 describes the input and output of each step in the approach. Generally, the method begins by dividing the genome into small, fixed, non-overlapping regions (windows) which are then classified into high and low density regions based on their underlying read count. Contiguous adjacent regions with similar densities are merged together to construct larger regions called *islands*. Constructed island locations are then mapped between samples in order to refine island boundaries and tested for differential expression. Features (typically genes) overlapping a set of DE islands are tested for DE by using combined

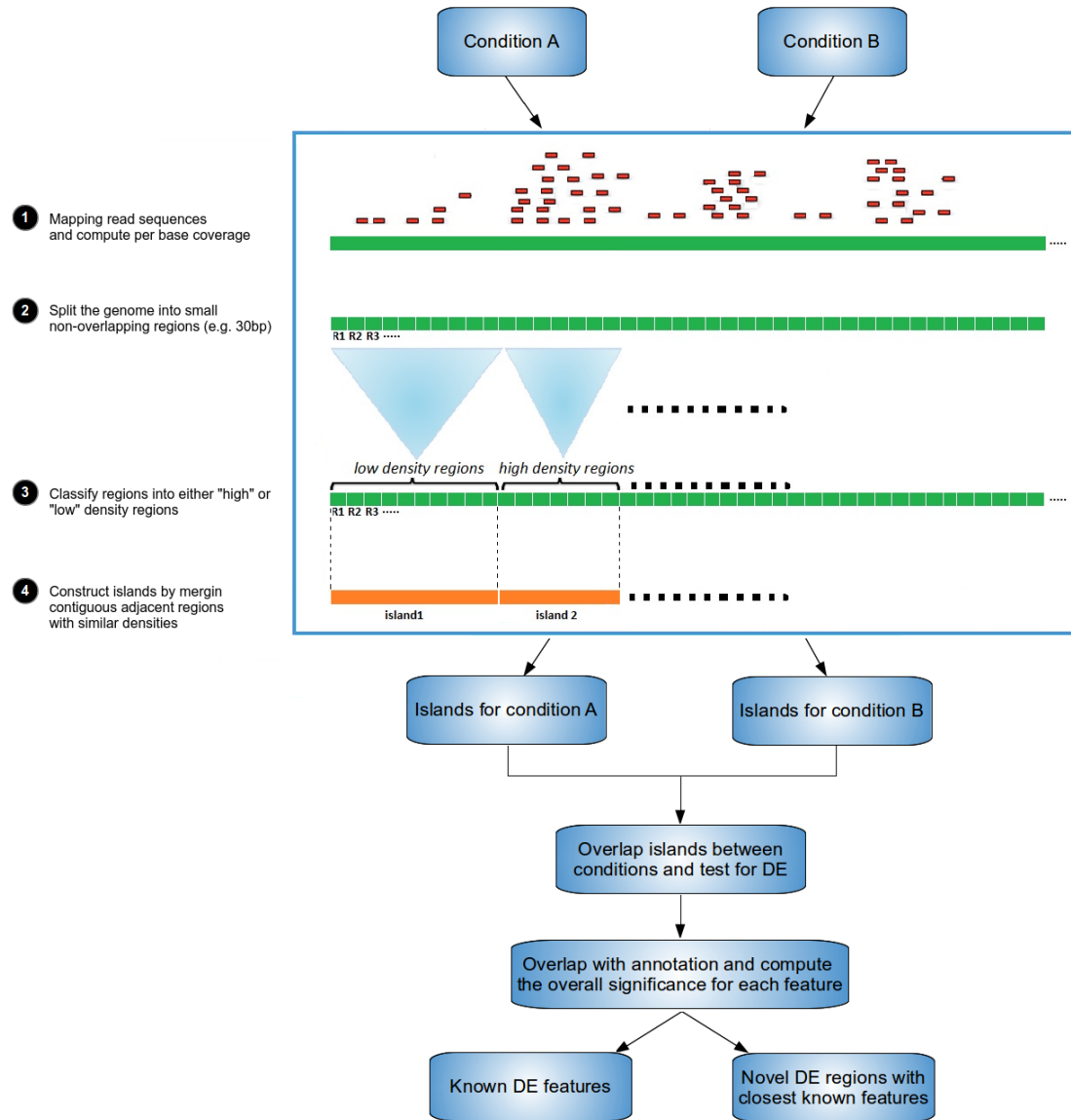


Figure 5.1: Workflow of the island-based approach.

p -value methods. DE islands that do not overlap with any features are considered novel regions which are annotated along with their closest features.

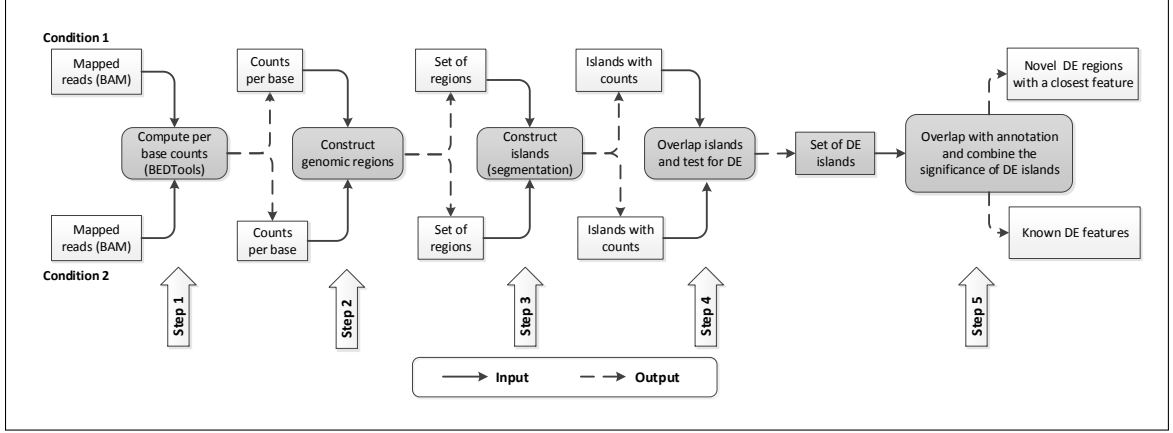


Figure 5.2: The input and output of IBSeq steps.

5.2.1 IBSeq Framework Steps

5.2.1.1 Compute Per Base Abundance

To generate per base abundance (Figure 5.2, step 1), aligned short read sequences (often in the form of SAM/BAM format) are first converted into BED format (a tab-delimited text file that defines a feature track). Each BED file for each sample is split by chromosome and a per base count is computed for each chromosome separately using BEDTools [125]. Thus, if s_j represents sample j , then C_{s_j} represents the complete set of per base counts separately for each chromosome $C_{s_j} = \{C_{s_j,chr1}, C_{s_j,chr2}, \dots, C_{s_j,chr_k}\}$, where $C_{s_j,chr1}$ is the per base count for chromosome 1, $C_{s_j,chr2}$ is the per base count for chromosome 2, and so on for each of the k chromosomes. The purpose of computing per base counts for each chromosome has two advantages: (1) the process is much faster than considering all chromosomes simultaneously and (2) the approach will have more flexibility to work with specific chromosomes in case differential expression analysis needs to be performed for a particular chromosome.

5.2.1.2 Genome Partition and Region Classification

In step 2, region construction (or genome partition) begins by first summarizing per-base read counts, generated in step 1, over a fixed window to minimize small variance in coverage due to noise (Figure 5.2, step 2). The size of the window is allowed to vary using smaller window sizes (10-60bp). Thus, for each sample s_j , a set of regions R_{s_j} is constructed from the set of per base counts C_{s_j} for each chromosome. $R_{s_j} = \{R_{s_j,chr1}, R_{s_j,chr2}, \dots, R_{s_j,chr_k}\}$. Once the genome is split into windowed regions, each region is classified as *high* or *low* density (density is based on the number of reads in this context) using an average threshold t adapted from Zang *et al.* [166]. Thus, regions with read counts above or equal to the threshold t are classified as *high* density regions and regions with read counts below the threshold are classified as *low* density regions. The threshold t is sample specific and is defined based on a user-defined *p-value* and the probability quantile function of the *Poisson* distribution as an approximation for the expected number of reads:

$$\sum_{k=t}^{\infty} P(k, \lambda) \leq \text{p-value}$$

where k is the number of reads in a window and λ represents the average number of reads across all regions in the genome and calculated as $\lambda = wS_j/G$, where w is the region size, S_j is the total number of reads in experiment j , and G is the effective genome length.

5.2.1.3 Island Construction

In the island construction process, preliminary islands (or *pre-islands*) are constructed for each sample by merging contiguous regions with similar densities. High density pre-islands are constructed from adjacent high density regions and similarly low density pre-islands are constructed from adjacent low density regions (Figure 5.2, step 3). Generally, the high density pre-islands are constructed from the set of the

high density regions R_{high,s_j} and low density pre-islands are constructed from the low density regions R_{low,s_j} . Thus, the complete set of pre-islands in sample s_j is defined as:

$$I_{s_j} = \{I_{s_j,chr1}, I_{s_j,chr2}, \dots, I_{s_j,chr_k}\}$$

The low density regions denote the start and end points for individual high density pre-islands. Each high density pre-island is allowed to include a number of *low* density regions based on a pre-defined cost threshold c (or gap size). Figure 5.3 shows an example where one *low* density region is allowed in a given pre-island.

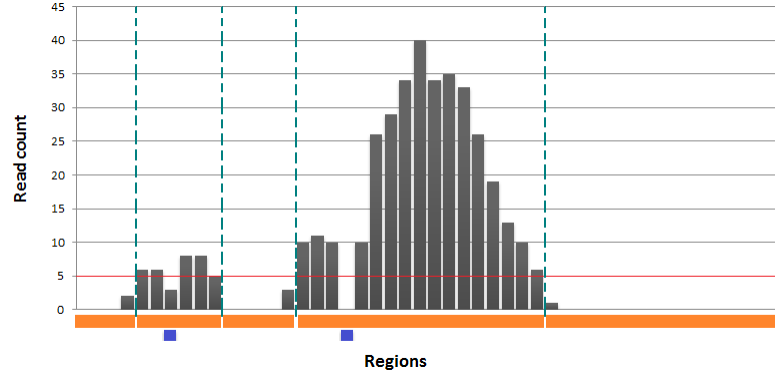


Figure 5.3: Illustration of pre-island definition [166]. Regions are shown as genome coordinates along the x-axis with each bar representing one region. The y-axis denotes the read count for each region. The orange bar denotes the constructed islands using a threshold $t = 5$ (red line) and a gap size 1. The blue boxes show low density regions included in that pre-island.

5.2.2 Island Differential Expression Testing

The primary goal of island DE testing is to test the null hypothesis H_0 that an island has the same expression level between samples versus the alternative hypothesis H_1 that an island has a significant difference between samples. In order to perform this test, constructed pre-islands are overlapped between each two samples

(pairwise comparison) and split into smaller islands where the start and stop locations are different as shown in Figure 5.4. Each island then comprises an overlapping

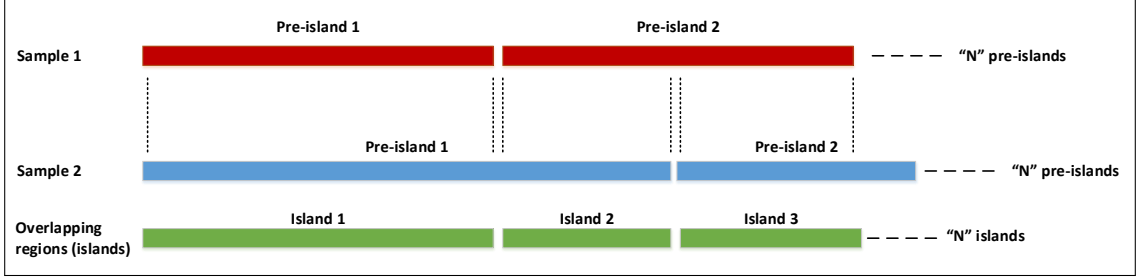


Figure 5.4: Illustration of overlapping islands between samples. The overlapping region (island) has to correspond to a high density pre-island in at least one sample to be considered for the DE test. Overlapping regions constructed from low density pre-islands in both samples are removed and not considered for the DE test.

region between the two samples that can be subsequently tested for differential expression between conditions using statistical tests such as a parametric *t*-test or a non-parametric *Wilcoxon test* (Figure 5.2, step 4). Islands constructed from *low* density pre-islands across samples are removed and only islands constructed from *high* density pre-islands in at least one sample are kept. To conduct an accurate comparison, read counts are first normalized based on the total number of mapped reads in each sample. We call this normalization method *Islands Per Million (IPM)* (an adaptation form of the well-known method *transcripts per million (TPM)*). The IPM method is defined as:

$$IPM = \frac{K_{ij}}{M_j} \times 10^6$$

where K_{ij} is the read counts of island i in sample j and M_j is the total number of mapped reads for sample j . Since islands tested for DE have the same length, it is not needed to include the island length in the normalization computation. To test for DE islands across the two samples, two statistical tests *Welch's t*-test and *Wilcoxon test* are used on the normalized *IPM* values. The Welch's t-test is an adaptation of the

well-known *Student's t-test* in which the test assumes the two samples have unequal variances. The test statistic, T , and degrees of freedom are defined as:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}}, \quad df = \frac{\left(\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}\right)^2}{\frac{s_1^4}{N_1^2(N_1-1)} + \frac{s_2^4}{N_2^2(N_2-1)}}$$

where \bar{X}_i represents the i th island mean, s_i^2 represents the i th island variance, and N_i represents the i th island size. As an alternative to the parametric Welch's t-test, we also perform a nonparametric Wilcoxon Rank-Sum test (also known as Mann-Whitney U test). The Wilcoxon test is based on the ranks of the observations and not the raw data. The test-statistic, T , is calculated as the sum of ranks in the smaller group. To understand this test, suppose that N_{1i}, \dots, N_{ni} represents the read counts of islands i in n samples. If $R_{ij}(N)$ is the rank of all counts N_{ij} , the Wilcoxon test statistic, T then is defined as:

$$T_i = \sum R_{ij}(N) - \frac{n_1(n_1+1)}{2}$$

where $N = n_1 + n_2$ and n_1 is the length of the island in the first sample.

5.2.3 Combined Significance of DE Islands

To detect which genetic features (e.g. genes) are differentially expressed between samples, the significance of DE islands that overlap with each feature is combined using combined p -values methods (e.g. Fisher's method). DE islands that do not correspond to any feature are considered novel DE regions. Those regions are annotated along with their closest features. In IBSeq, six combined p -value methods, shown in Table 5.2 and detailed in Chapter 6, are implemented.

5.2.4 IBSeq Algorithm

The IBSeq approach as described above consists of a number of steps to perform the differential expression analysis. From the computer science perspective, these

Table 5.2: Combined p -value methods implemented in IBSeq.

Method	Description
Fisher's [40]	$\chi_F^2 = -2 \sum_{i=1}^k \ln(p_i)$. If the null hypothesis H_0 is true for all k tests, χ_F^2 will have a <i>chi</i> -squared distribution with $2k$ degrees of freedom.
Z-transform [148, 95]	$Z = \frac{\sum_{i=1}^k Z_i}{\sqrt{k}}$. If the null hypothesis H_0 is true for all k tests, Z will have a standard normal distribution.
Weighted Z-Test [110]	$Z_w = \frac{\sum_{i=1}^k w_i Z_i}{\sqrt{\sum_{i=1}^k w_i^2}}$. Generalized form of the Z-transform method.
Minimum P-value [153]	$P = 1 - (1 - p_{[1]})^n$. if $p_{[1]}$ is the minimum of p_1, p_2, \dots, p_n , then $p_{[1]}$ has a beta distribution with parameters 1 and n in case H_0 is true for all n tests.
Logit [111]	Each p -value is transformed to a logit, $\ln(\frac{P}{1-P})$. $T = -\sum_{i=1}^k \ln(\frac{P_i}{1-P_i})$. Under H_0 , T is logically equivalent to: $\sqrt{\frac{k\pi^2(5k+2)}{3(5k+4)}}$ and has standard t-distribution with $5k+4$ degrees of freedom.
Weighted-Sum	$P = \frac{\sum_{i=1}^k l_i p_i}{\sum_{i=1}^k l_i}$, where p is the p -value and l is the island length.

steps are the algorithm processes needed to perform a task which can be represented as a flowchart (Figure 5.5). The corresponding pseudocode of the IBSeq algorithm flowchart is depicted in Algorithm 1.

5.3 Experimental Results

In order to examine the performance of the IBSeq approach, we conducted a gene differential expression analysis using available RNA-Seq data described below. To evaluate its performance, the IBSeq was compared to a number of current gene DE methods including Cuffdiff, DESeq, and edgeR. Although we have implemented

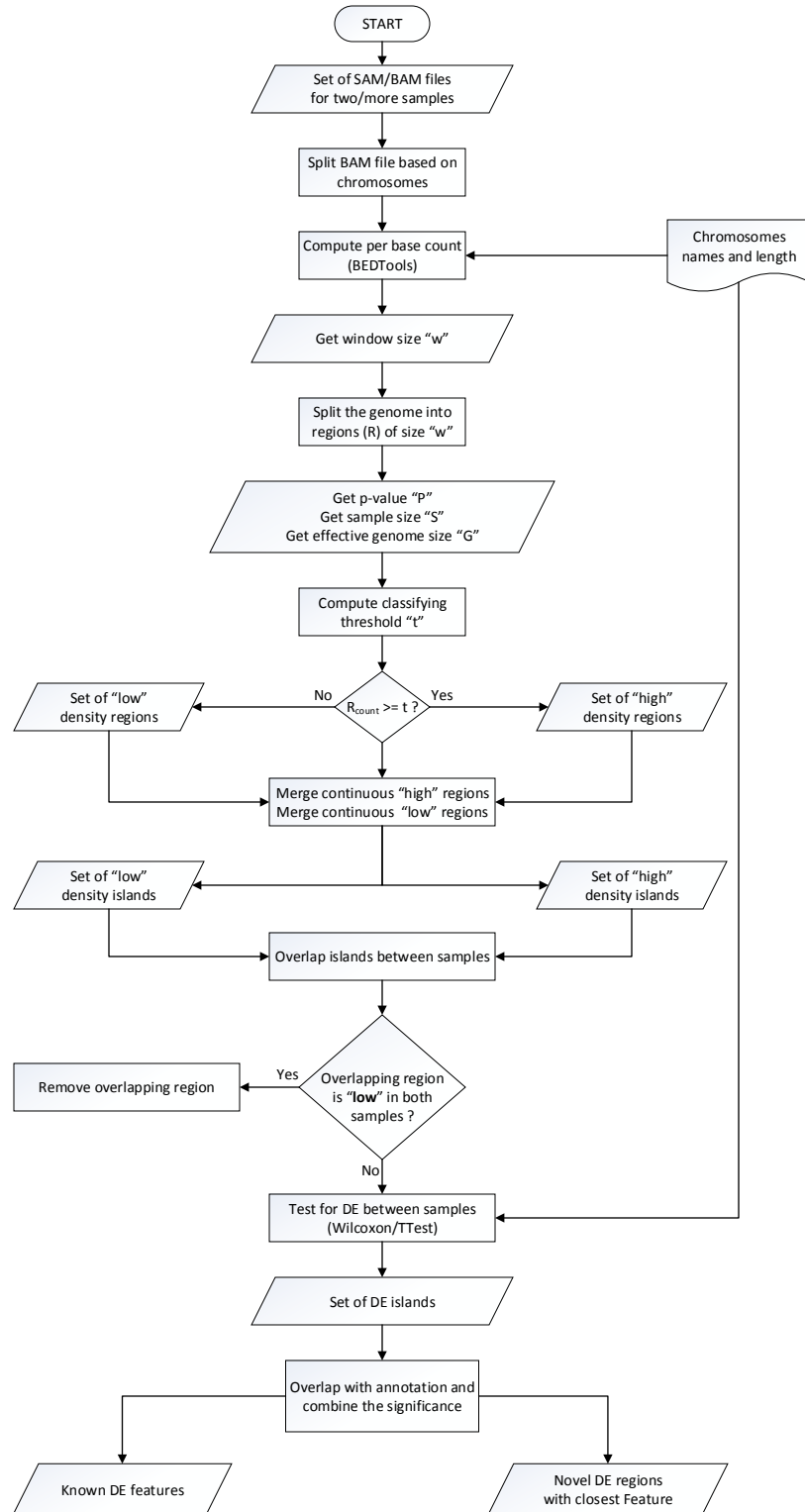


Figure 5.5: The flowchart of IBSeq algorithm.

Algorithm 1 IBSeq Algorithm

Input: set of aligned RNA-Seq reads A for m replicates R in n samples S

$$A = \{\{A_{s_1,1}, \dots, A_{s_1,m}\}, \{A_{s_2,1}, \dots, A_{s_2,m}\}, \dots, \{A_{s_n,1}\}, \dots, \{A_{s_n,m}\}\}$$

Output: set of differentially expressed islands I across samples.

set of known differential expression features F across samples S

set of novel differential expression regions N across samples S

```
1: while  $T \neq \emptyset$  do
2:   for each sample  $s_i \in S$  do
3:     for each replicate  $r_j \in R$  belong to  $S_i$  do
4:       convert alignment SAM file  $a_{s_i r_j}$  into BAM  $b_{s_i r_j}$ 
5:       convert BAM  $b_{s_i r_j}$  into BED  $d_{s_i r_j}$  and split  $d_{s_i r_j}$  by chromosomes
6:       compute per base count set  $C_{s_i r_j}$  for replicate  $r_j \in S_i$  from BED file  $d_{s_i r_j}$ 
           $C_{s_i r_j} = \{C_{s_i r_j, chr1}, C_{s_i r_j, chr2}, \dots, C_{s_i r_j, chr_k}\}$ 
7:       construct a set of regions  $G_{s_i r_j}$  for replicate  $r_j \in S_i$  from  $C_{s_i r_j}$ 
           $G_{s_i r_j} = \{G_{s_i r_j, chr1}, G_{s_i r_j, chr2}, \dots, G_{s_i r_j, chr_k}\}$ 
8:     end for
9:     combine replicate files  $r \in S_i \rightarrow G_{s_i}$ 
10:    construct a set of islands  $I_{s_i}$  for sample  $S_i$  from  $G_{s_i}$ 
           $I_{s_i} = \{I_{s_i, chr1}, I_{s_i, chr2}, \dots, I_{s_i, chr_k}\}$ 
11:  end for
12:  /* Test for DE islands between  $N$  conditions (for each pair  $s_i, s_j \in S$ ) */
13:  for  $i \leftarrow 1$  to  $N$  do
14:    for  $j \leftarrow 1$  to  $N$  do
15:      overlap the set of islands  $I_{s_i}, I_{s_j}$  between  $s_i, s_j$ 
16:      fetch read counts for each overlapped region from  $G_i, G_j$ 
17:      test for DE islands between  $s_i, s_j$  ( $I_{DE} \leftarrow I_{DE(s_i, s_j)}$ )
18:    end for
19:  end for
20:  /* combine the significance of each comparison */
21:  for each comparison  $s_i, s_j$  do
22:    overlap  $I_{DE}$  with annotation  $GTF/BED$  and computer overall significance
      for each feature  $F$ 
23:    compute novel DE regions  $G_{novel(s_i, s_j)}$ 
24:  end for
25: end while
```

six combined p -value methods, only Fisher's method was used in this analysis since the aim is to compare the performance of the IBSeq to the current methods. Chapter 6 discusses the IBSeq performance using the six combined p -value methods.

5.3.1 Datasets

5.3.1.1 MAQC Datasets

To test the performance of the IBSeq, two datasets related to the MicroArray Quality Control (MAQC) Project [142] were obtained. The experiments in the two datasets analyze two biological samples: Ambion’s human brain reference (Brain) and Stratagene’s human universal reference RNA (UHR) [16]. In both datasets, the two samples were prepared using one library preparation and sequenced in seven lanes and two flow-cells using an Illumina Genome Analyzer II (GAIIx). The first dataset was sequenced with RNA-Seq reads of length 35bp with only one biological replicate [16]. This dataset was obtained from NCBI’s Sequence Read Archive (SRA) with Accession IDs: SRX016359 and SRX016367 for Brain and UHR respectively. The second dataset was sequenced with 50bp RNA-Seq read length with one biological replicate [113]. This dataset was obtained from SRA with Accession IDs: SRX027129 and SRX027130 for Brain and UHR respectively.

5.3.1.2 qRT-PCR Datasets

As part of the MAQC project, 1044 genes were selected to be assayed by qRT-PCR. The expression of those genes were quantitatively measured for Brain and UHR samples using TaqMan Gene Expression Assay [16, 158]. This data is used as a “gold-standard” to evaluate the performance of IBSeq for detecting DEGs obtained from Gene Expression Omnibus (GEO) with series ID GSE5350. Four replicates were obtained for Brain (GSM129638-GSM129641) and four replicates for UHR (GSM129642-GSM129645). We removed genes whose identifiers are not present in RefSeq resulting in a total of 1033 genes. We follow Bullard *et al.* [16] and Wan *et al.* [158] for processing this data and compute the expression level of each gene for

each replicate. Thus, for gene i at replicate j , the expression is defined as:

$$Y_{i,j} = \frac{\log_2(\Delta C_{i,j})}{\log_2(e)}$$

where $\Delta C_{i,j} = C_{i,POLR2A} - C_{i,j}$ denotes the original qRT-PCR expression (C is the normalized threshold cycle number and POLR2A is the reference gene). This was done to transform the original expressions, which are in log base-2, to the natural logarithmic scale. The log-fold change is then defined as the difference of average across the four replicates $\bar{Y}_{UHR,j} - \bar{Y}_{Brain,j}$. To define the DE genes (positive set) and non-DE genes (negative set), genes with absolute log-fold change > 2 are considered DE genes and genes with absolute log-fold change < 0.2 are considered as non-DE genes. Out of 1033 genes, 309 genes fall in the positive set (true positives TP) and 174 genes in the negative set (true negatives TN). Genes with absolute log-fold change > 0.2 and < 2 are discarded and not used in this study.

5.3.2 Evaluation of IBSeq Approach for Detecting DEGs

To test the performance of the IBSeq, the Receiver Operating Characteristic (ROC) is used to evaluate the relationship between sensitivity (true positive rate) and specificity (false positive rate). We evaluate the results of the IBSeq approach for detecting DEGs by comparing it to three widely used methods: *Cuffdiff*, *DESeq*, and *edgeR*. For each method, the p -value is used to determine which genes are DE and which ones are not. Thus, for a given p -value threshold, we consider genes with p -values smaller than or equal to the threshold as DE genes and genes with p -values greater than the threshold are non-DE genes. Using the qRT-PCR data as a “gold-standard”, the predicted results are compared to the set of 483 genes generated in Section 5.3.1.2 and true positive rate (TPR) and false positive rate (FPR) are calculated. These two measures are computed as follows:

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{TN}{TN + FP}$$

where TP denotes the true positive, FN is the false negative, and TN is the true negative sets. Using this information, we generate ROC curves for all methods based on different p -value cutpoints using both datasets I and II. We use the area under the ROC curve (AUC) calculated using the trapezoidal rule to measure the accuracy of each method and evaluate the performance for detecting DEGs.

5.3.3 Construct Islands and Test for DE Islands

To construct islands, short read sequences of the two samples Brain and UHR in each dataset were first mapped to reference genome (hg19) using **Bowtie** [79] with the default parameters allowing for two mismatches. To construct regions, we applied a window of size 30bp. To classify regions, the average thresholds for the two samples were computed using a p -value of 0.05 resulting in $t = 3$ for both samples. Thus, regions with read count above or equal to 3 reads were classified as *high* density regions and regions with read counts below 3 reads were classified as *low* density regions. Using $t = 3$ and $c = 1$ (c is the gap size), islands were constructed for Brain and UHR samples for each dataset. Table 5.3 shows detailed information about the constructed islands for each sample in the two datasets.

Table 5.3 indicates that the average length of *low* density islands is much larger than the average length of *high* density islands which agrees with the fact that a large portion of the human genome (about 98%) is non-coding and only about 2% is coding regions. Thus, the coding regions (transcribed regions) should fall within high density islands. To test for DE islands, two statistical tests *Welch's t-test* and *Wilcoxon* test were applied to compute the test statistics T and the p -values.

Table 5.3: Detailed information of constructed islands.

Dataset I				
Sample	Number of Islands		Average Island Length	
	High	Low	High	Low
Brain	389,376	389,399	78.4487	3896.63
UHR	391,680	391,703	80.0481	3871.65

Dataset II				
Sample	Number of Islands		Average Island Length	
	High	Low	High	Low
Brain	465,298	465,321	86.0159	3240.47
UHR	446,156	446,179	87.2413	3381.96

5.3.4 Combined Significance of DE Islands

Since all the methods being compared report results at the gene level, an overall p -value for a gene needs to be generated from the island p -values. Therefore, the p -values of the islands overlapping with each gene are combined using Fisher’s method [40]. Fisher’s method computes the overall p -value p by combining the significance of multiple tests using the formula:

$$-2 \sum_{i=1}^k \ln(p_i) = \chi_{2k,p}^2$$

where p_i is the p -value of the i th island and k is the number of islands tested. Thus, if none of the islands are DE, the p -values p_i are independent and uniformly distributed on the unit interval $p_i \sim U(0, 1)$ which indicates the null hypothesis H_0 is true. Hence, $\chi_{2k,p}^2$ denotes the upper p point of the probability of a chi-squared distribution with $2k$ degrees of freedom [158, 28, 62].

5.3.5 Evaluation and Comparison

The performance of IBSeq approach was evaluated using the benchmark RNA-Seq datasets for the Brain and UHR samples. The portion of the qRT-PCR data that we selected in Section 5.3.1.2 with 309 genes in the positive set (true DE) and 174 genes in the negative set (true non-DE) was used to compare the results of the IBSeq to the other DE methods. Since our approach is based on combining the p -values of islands overlapping with the genes, for all methods, the p -value was used as a measure of significance in this study.

When we computed the overall p -values for the two sets of genes, using dataset I, for the true DE set with 309 genes, nine genes were missing (none of the islands overlapped with those genes) and 22 genes were missing from the true non-DE set with 174 genes. Per base counts for each of these missing genes were checked and it was determined they have low counts and consequently their corresponding islands were classified as *low* density and therefore were removed. To verify this conclusion, we compared the counts in Cuffdiff and in the DESeq and edgeR count table. We found a strong agreement between our approach and the other methods in terms of low read counts. For instance, Cuffdiff reported that out of the 9 missing genes, 8 genes were not tested (NOTEST) indicating there were too few counts to perform a significance test and similarly out of 22 missing genes, 20 were not tested for the same reason. Giving this strong evidence these genes are not DE between the two samples, they were treated as non-DE genes and counted as false negatives (FN) for the nine missing genes and as true negatives (TN) for the 22 genes. Similarly with dataset II, 8 genes were missing in the positive set and 25 genes were missing in the negative set due to low counts. Results from Cuffdiff supported our approach in that Cuffdiff described all those genes as NOTEST which indicate the low counts. We performed the same filtering for dataset II and included those genes in the false negative set

and true negative set in the calculation of true positive and false positive rates. In order to compare IBSeq with other existing DE methods, we performed differential expression analysis for the same MAQC datasets (I and II) using Cuffdiff, DESeq, and edgeR and computed the p -values for the set of 483 (309+174) genes. With the exception of Cuffdiff, the differential expression analysis of DESeq and edgeR were performed using the same count table of all genes annotated in RefSeq. This count table was generated using `htseq-count` version 0.5.4p1 [5] with the same RefSeq GTF file downloaded from the UCSC genome browser.

For the set of 483 genes, first we looked at the p -value distribution (Figure 5.6) generated by each method using dataset I and dataset II. Using a p -value cutoff ≤ 0.05 (5%), we could observe that our approach performs well in detecting the true DE genes whereas it performs slightly worse in detecting the true non-DE genes. This is illustrated in Figure 5.6 where the p -value histograms of the IBSeq is highly skewed to 0 indicating that a large number of true DE genes will be detected (giving the fact that approximately 65% of the gene set falls in the positive set). Since this histogram is slightly skewed far from 0, there is a high possibility that the IBSeq approach will not perform well in detecting true non-DE genes. In contrast, the p -value histograms of Cuffdiff, DESeq, and edgeR were not as highly skewed to 0 as the IBSeq approach indicating the likelihood of not performing well in detecting true DE genes. However, the histograms show a moderate shift toward 1 meaning those methods will perform well in detecting true non-DE genes.

Although Cuffdiff, DESeq, and edgeR did not perform well in detecting true DE genes, they were excellent in detecting almost the complete set of the true non-DE genes with 172, 173, 171, respectively out of 174. Table 5.4 shows the number of true positive (TP) and true negative (TN) genes detected by each method using a p -value ≤ 0.05 and Figure 5.7 shows the bar graph of those numbers.

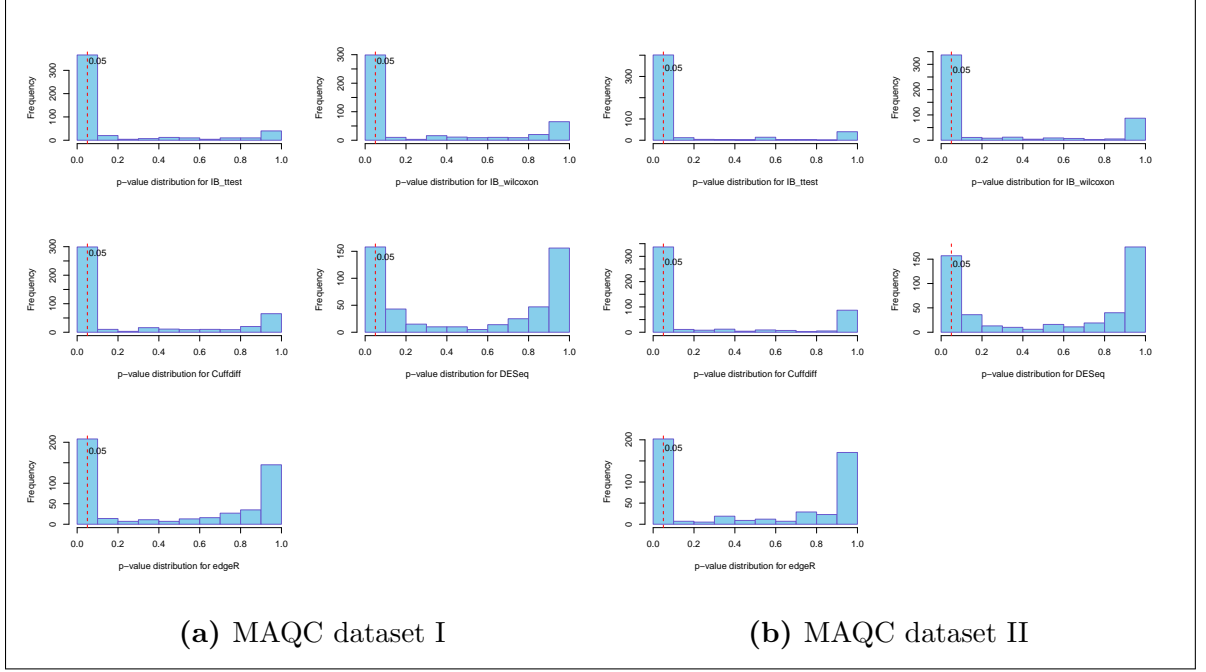


Figure 5.6: The distribution of p -values for the four methods (IB=IBSeq).

Table 5.4: Number of true DE and true non-DE genes found by each method using p -value ≤ 0.05 .

Method	TP(D I)	TP(D II)	TN(D I)	TN(D II)
IBSeq-TTest	282	291	113	80
IBSeq-Wilcoxon	269	280	149	128
Cuffdiff	190	176	172	173
DESeq	136	134	173	173
edgeR	193	185	171	172

Table 5.4 indicates that the IBSeq approach performs well in detecting TP genes whereas Cuffdiff, DESeq, and edgeR were much better in detecting the TN genes. As we see in Table 5.4 and Figure 5.7, the IBSeq approach was not able to detect a high number of true non-DE genes like other methods. DESeq and edgeR performed similarly since both methods use similar statistical tests (a form of Fisher's exact test) and both model read counts by using a negative binomial distribution

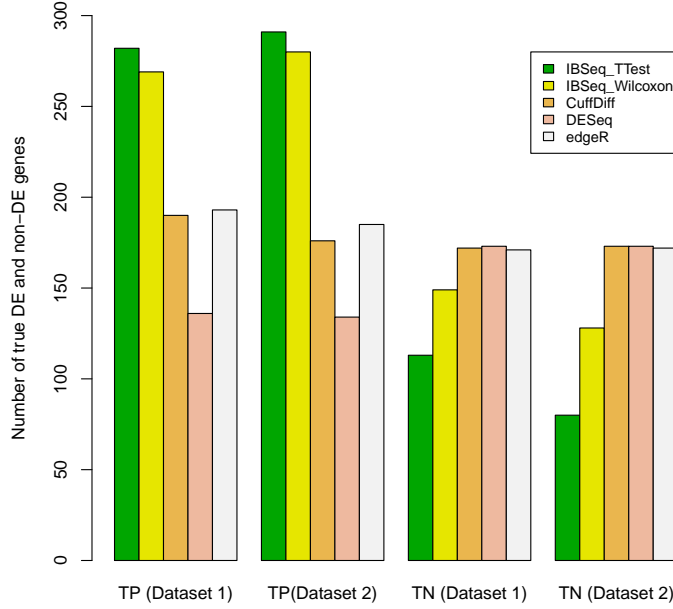


Figure 5.7: Number of DE and non-DE genes detected by each method using p -value ≤ 0.05 .

(NB). According to the DESeq documentation, DESeq is conservative in detecting DE genes. Thus, it is of no surprise we do not see a large number of true DE genes detected by DESeq. To plot the ROC curves for the four methods, we set different thresholds of the p -values and calculated the true positive rate (TPR) and false positive rate (FPR) for each method. Generally, a method that performs better will give a ROC curve with higher TPR than other methods with the same value of FPR. We computed the AUC and use it as a measure to compare the performance of each method. Figure 5.8 shows the ROC curves of the four methods on the two MAQC datasets.

Looking at the AUC of each method in Figure 5.8, it is clear the two versions of our approach (*t-test* and *Wilcoxon*) outperform other methods in both datasets. IBSeq using the Wilcoxon test performed the best among the four methods with AUC

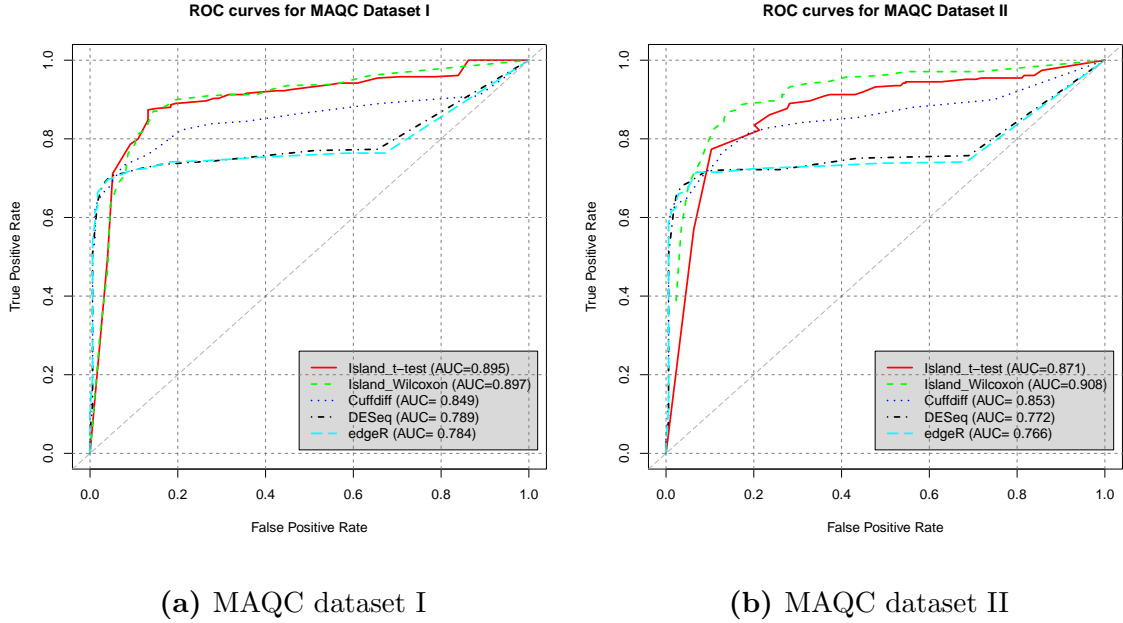


Figure 5.8: The ROC curves for the four methods using MAQC datasets.

= 0.897 for dataset I and AUC= 0.908 for dataset II. Similarly, IBSeq using Welch's t-test performs well in both datasets with AUC = 0.895 for dataset I and AUC = 0.871 for dataset II. Cuffdiff performed better than DESeq and edgeR but not as well as IBSeq.

We further looked at the number of differentially expressed genes shared between each pair of methods (Table 5.5) for both datasets. This gives an indication on the level of agreement between methods in detecting the true DE genes. Table 5.5 indicates a strong agreement in detecting true DE genes between the two versions of IBSeq. Compared to other methods, both versions of IBSeq were able to detect almost all true DE genes detected by other methods for the two datasets. For instance, out of 190 true DE genes detected by CuffDiff, the IBSeq approach was able to detect 183 and 178 respectively for the two versions in the first dataset. In the second dataset, the number is even higher as shown in Table 5.5. The same observation is applied for DESeq and edgeR where almost all true DE genes detected by those methods were

Table 5.5: Number of shared true DE genes detected by each method using p -value ≤ 0.05 . The diagonal represents the numbers of true DE genes detected by each method.

Dataset I					
	IBSeq-TTest	IBSeq-Wilc.	Cuffdiff	DESeq	edgeR
IBSeq-TTest	282	268	183	135	186
IBSeq-Wilc.		269	178	132	180
Cuffdiff			190	123	165
DESeq				136	136
edgeR					193

Dataset II					
	IBSeq-TTest	IBSeq-Wilc.	Cuffdiff	DESeq	edgeR
IBSeq-TTest	291	279	173	134	181
IBSeq-Wilc.		280	170	131	178
Cuffdiff			176	118	154
DESeq				134	134
edgeR					185

also detected by our approach. This indicates the set of DE genes found by the IBSeq contains a large number of DE genes found by other methods. To look at the overlap between all methods and determine the number of true DE genes and true non-DE gene shared between all methods, Figures 5.9 and 5.10 depicts the complete overlap between the number of TP and TN genes detected by each method.

One caveat with the choice of the MAQC datasets is the ratio of DE to non-DE genes is skewed in comparison to typical datasets where it might be expected that only 5-10% of the genes are differentially expressed. These datasets were chosen for comparative purposes since they contain experimental validation for differentially expressed genes. That being said, we have also applied the IBSeq approach to whole transcriptome RNA-Seq data as well (results not shown) for the datasets discussed in Chapter 1 Figure 1.2. Initial results suggest a similar performance to the MAQC

data with the majority of novel islands detected within or in close proximity to known transcribed regions.

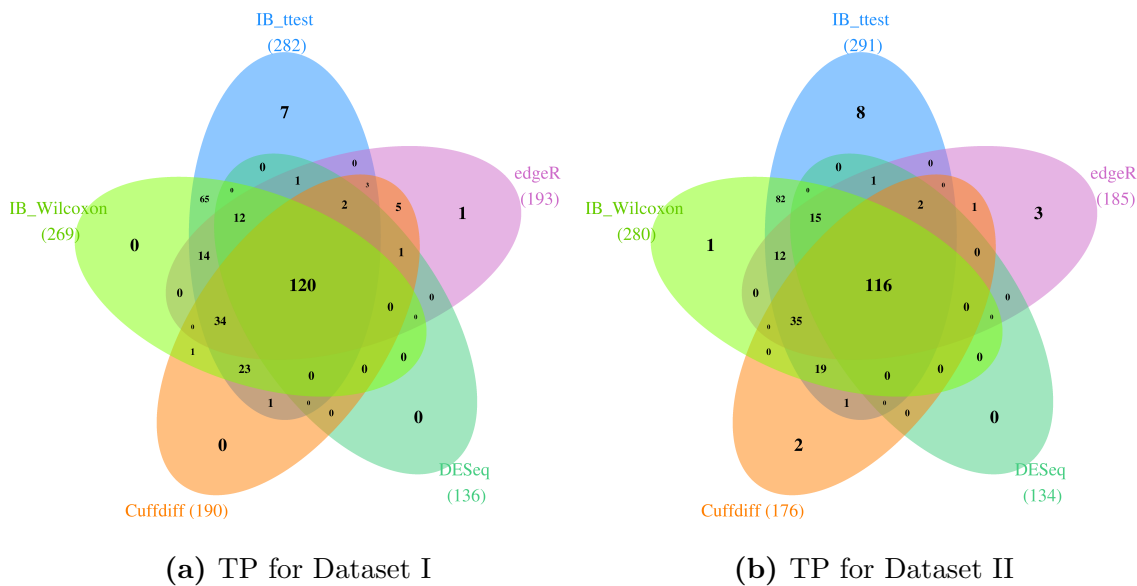


Figure 5.9: Overlap between true DE genes found by each method.

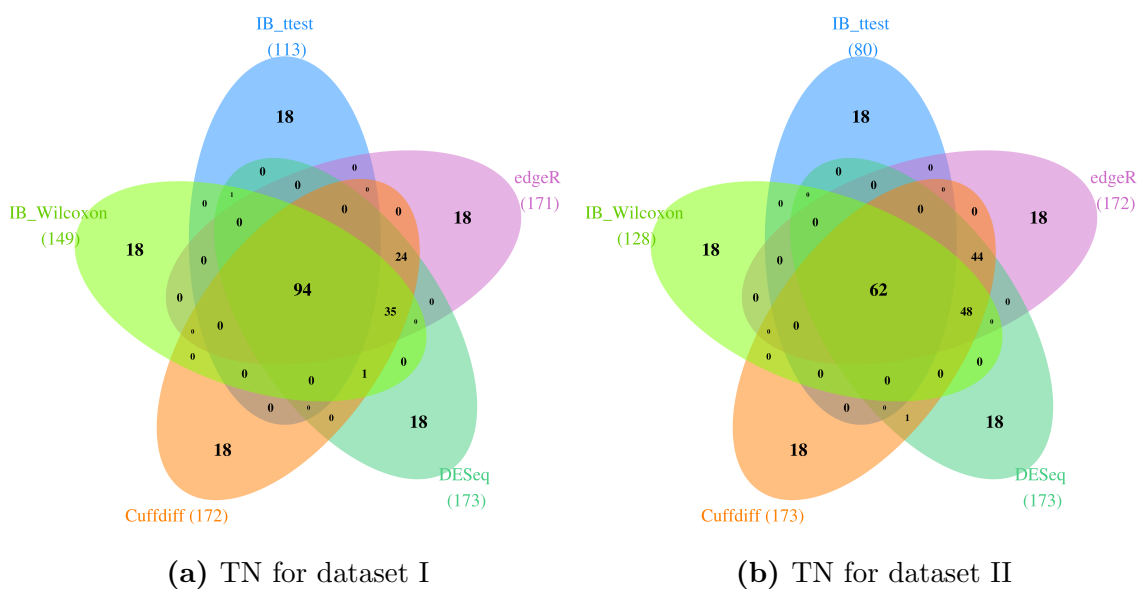


Figure 5.10: Overlap between true non-DE genes found by each method.

5.4 Conclusion

In this Chapter, we presented a novel approach for detecting differential expression in genome regions that does not rely on genomic annotations. The key idea of this approach is the segmentation methodology in which individual islands of expression are constructed based on windowed read counts and compared across experimental conditions to determine differential island expression. We illustrated how this approach is used to detect differences in expression without requiring any prior knowledge of isoforms where the only input to this approach is the raw data (short read sequences). To assess the performance of our method, we conducted a differential expression analysis using two benchmark MAQC RNA-Seq datasets. To detect DEGs, Fisher’s method for combining the significance of multiple tests was used. The performance of IBSeq approach was evaluated by comparing its results to three widely used methods for differential expression analysis. IBSeq was able to detect a high number of true DE genes using $p\text{-value} \leq 0.05$ and performed the best among the four methods based on ROC analysis. However, in detecting the true non-DE genes, IBSeq did not perform as well as expected. Although the approach has detected a reasonable number of the true non-DE genes, it was not as high as the other methods considered. Considering the results obtained, IBSeq performs well in terms of detecting true positives. However, it still leaves room for improvement in detecting true non-DE genes which is intensively discussed in Chapter 7.

CHAPTER 6

COMBINING THE SIGNIFICANCE OF GENOMIC REGIONS - A COMPARATIVE STUDY

6.1 Introduction

One of the motivations of developing the IBSeq approach is that summarizing read counts on the gene level tend to result in inaccurate detections since most genes consist of multiple exons and therefore the distribution of read counts in exons for a single gene can be different [158]. By taking into account the significance of different regions in the gene, IBSeq can break down the gene region (or any genetic feature) into multiple small regions and test for differential expression across those regions. Then for each gene, we combine the significance of genomic regions overlapping with that gene using well-known combined p -value methods. Figure 6.1 describes this process. By doing that, each region in the overall gene region will participate in the

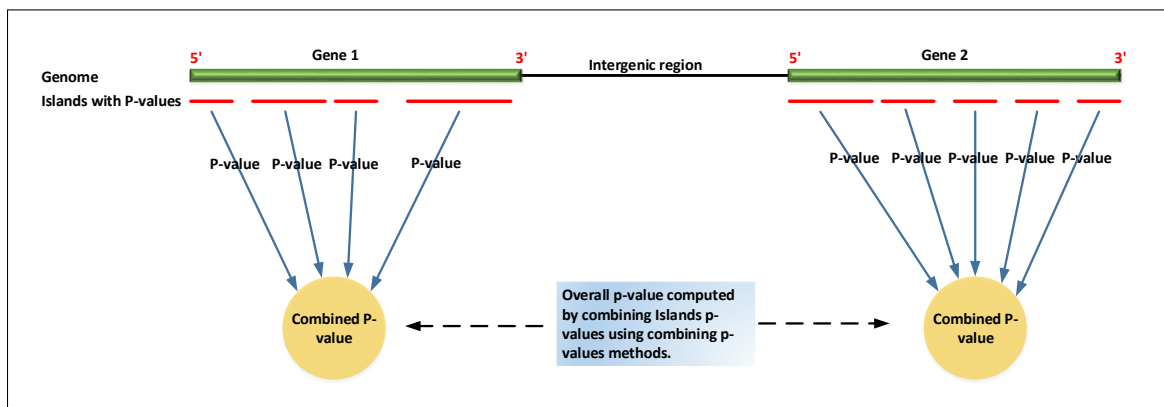


Figure 6.1: Example of combining p -values from multiple genome regions.

computation of the overall gene significance based on its degree of importance. This will ensure that regions in the gene will not be treated equally.

The concept of combining significance (p -values) from multiple tests has been intensively discussed in meta-analysis techniques from different fields. In biological experiments for instance, these approaches have been used to integrate results from multiple studies to detect which genes (or any genetic features) are differentially expressed across samples. As an example, Chapman and Whittaker [22] used several combined methods to integrate results of multiple single nucleotide polymorphisms (SNPs) tests in a gene or region. In differential expression analysis using microarrays, the technique has been used to combine p -values from probe level tests of significance. Hess *et al.* [62] proposed using Fisher’s method to combine the significance of probe level tests to identify DE genes using Affymetrix arrays. Li and Tseng [83] proposed an adaptively weighted statistics method to combine multiple genomic studies for detecting differentially expressed genes.

Since IBSeq is based on combining the significance of islands corresponding to each feature, to determine the performance of the implemented combined p -value methods, we conducted a comparative analysis study to compare six combined p -value methods using publicly available RNA-Seq datasets. The framework applied here is similar to the one in microarray studies where specific methods are used to combine the significance of probe sets for individual genes. Similarly, we used combined p -value methods to aggregate the significance of islands corresponding to a chromosomal region (e.g. gene, exon, transcript) as shown in Figure 6.1.

In this chapter, we present the results of this study. We first applied IBSeq to test for island differential expression and compute the p -values for each “island” using four MAQC datasets [16, 142, 113] and Marioni’s liver and kidney dataset [100]. In the next step, the significance of islands corresponding to each gene is combined

using six p -value methods: Fisher’s, [40], Stouffer’s Z-transform [148], Weighted z -test [110, 95], Minimum p -value [153], Logit [111], and Weighted-sum. To evaluate the performance of each method, ROC curves were generated for each MAQC dataset and auROC was used as a performance metric. On the liver and kidney dataset, we evaluated the performance of each method by looking at the number of detected genes that overlap with the original results presented in Marioni’s paper [100].

6.2 Methods

6.2.1 Combined P-value Methods

IBSeq was first applied to test for differentially expressed islands. We first tested the null hypothesis H_0 that an island has the same expression level across samples versus the alternative hypothesis H_1 that an island has a significant difference between samples. As a result, for each island, a test statistic t and p -value p were computed. To detect which genes were differentially expressed between samples, the p -value of the islands overlapping with each gene in the annotation were combined using six combined p -value methods. The use of combined p -value methods is based on the assumption that p -values p_1, p_2, \dots, p_n are independent for given samples [158].

In this study, six combined p -value methods (Table 5.2), Fisher’s, Stouffer’s z -score, Weighted z -test, Minimum p -value, Logit, and Weighted-sum were used. The first five methods are widely used methods for combining the significance from multiple tests. The sixth method (the Weighted-sum) is our proposed method.

6.2.1.1 Fisher’s Method

Fisher’s method [40] combines the significance by using p -values from k independent tests using the test statistic:

$$\chi_F^2 = -2 \sum_{i=1}^k \ln(p_i)$$

where p_i is the p -value of the i^{th} island and k is the number of islands corresponding to the tested gene. Thus, when none of the islands corresponding to a specific gene are DE indicating the null hypothesis H_0 is true for all k islands, the test statistic χ_F^2 will have a *chi*-squared distribution with $2k$ degrees of freedom [162].

6.2.1.2 Z-transform Method

The Z-transform method (sometimes called Stouffer's Z-Score, *z-test*, or *Normal test*) [148, 95] was proposed by Stouffer *et al.* in 1949. In this method, p -values are first transformed to z -values $Z_i = \Phi^{-1}(1 - p_i)$ where Φ is the cumulative distribution function of standard normal distribution [23]. The Z_i values are then combined using:

$$Z = \frac{\sum_{i=1}^k Z_i}{\sqrt{k}}$$

Therefore, when none of the islands corresponding to a specific gene are DE (H_0 is true for all k islands), Z will have a standard normal distribution [158].

6.2.1.3 Weighted z -test

The Weighted z -test [110] is a generalized form of the Z-transform method explained above proposed by Mosteller and Brush [110] in 1954 and Liptak [95] in 1958. In this method, a nonnegative weight w is assigned to each z -value. The weighted Z_w is then computed using:

$$Z_w = \frac{\sum_{i=1}^k w_i Z_i}{\sqrt{\sum_{i=1}^k w_i^2}}$$

If the weights of all tests are equal, this method is reduced to the Z-transform method. Similar to Z-transform, if none of the islands within a specific gene are DE, then Z_w

still has a standard normal distribution. Determining appropriate weights is an open issue. However, it has been shown that the sample size of each test can be used as the weight [95, 158]. Thus, we chose the square root of the length of each island l_i to be the weight assigned ($w_i = \sqrt{l_i}$).

6.2.1.4 Minimum p -value Method

The minimum p -value statistic [153] is another method for combining p -values proposed by Tippett in 1931. In this method, if $p_{[1]}$ is the minimum of p_1, p_2, \dots, p_n , then $p_{[1]}$ has a beta distribution with parameters 1 and n in case none of the islands corresponding to a specific gene are DE [153, 158]. Tippett's test procedure using the smallest $P_{[1]}$ is computed as follows:

$$p = 1 - (1 - p_{[1]})^n$$

He suggested that the combined null hypothesis H_0 should be rejected at level α if $p_{[1]} < 1 - (1 - \alpha)^{\frac{1}{n}}$.

6.2.1.5 Logit Method

The logit test [111] was proposed by Mudholkar and George in 1979. Each p -value in this method is transformed to a logit, $\ln(\frac{P}{1-P})$ and the combined logits are computed using a statistic:

$$T = - \sum_{i=1}^k \ln\left(\frac{P_i}{1-P_i}\right)$$

Under the null hypothesis H_0 , Mudholkar and George showed that T is logically equivalent to:

$$\sqrt{\frac{k\pi^2(5k+2)}{3(5k+4)}}$$

and has standard Student's t -distribution with $5k + 4$ degrees of freedom.

6.2.1.6 Weighted-sum Method

The Weighted-sum is a method we propose in this study. In this method, we simply multiply each p -value p by the island length l and divide the total of multiplications by the total length of all islands. Thus, if l_1, l_2, \dots, l_k represents the islands lengths corresponding to a specific gene and p_1, p_2, \dots, p_k are the island p -values, then the combined p -value is:

$$P = \frac{\sum_{i=1}^k l_i p_i}{\sum_{i=1}^k l_i}$$

6.2.2 Datasets

In this study, five publicly available RNA-Seq datasets were obtained from NCBI’s Sequence Read Archive (SRA). Among those, four datasets are related to the MicroArray Quality Control Project [16, 142, 113] and the fifth dataset is the widely used Marioni liver and kidney dataset [100]. All datasets are single-end reads generated by Illumina GA/GAII . A summary of the datasets used in this study is shown in Table 6.1.

Table 6.1: Summary of RNA-Seq datasets used in this study.

Dataset	Acession Number (SRA)	Read Length	Total Reads
MAQC2 [16]	SRX016359	35bp	81,250,500
	SRX016367	35bp	92,524,365
MAQC2 [113]	SRX027129	50bp	53,238,798
	SRX027130	50bp	59,561,348
MAQC2 [16]	SRX016366	35bp	81,250,481
	SRX016368	35bp	92,524,400
MAQC3-UHR [16]	SRX016369-SRX016372	35bp	183,797,505
Liver and kidney [100]	SRX000571	36bp	69,618,202
	SRX000605	36bp	66,404,506

6.2.2.1 MAQC Datasets

The experiments in the four MAQC datasets analyze two biological samples: (1) Ambion’s human brain reference (Brain) and (2) Stratagene’s human universal reference RNA (UHR). In MAQC2, each sample was prepared using one library preparation and sequenced on seven lanes (7 technical replicates each) on two flow-cells using an Illumina Genome Analyzer II (GAIIx). In the MAQC3 dataset, four different UHR library preparations were sequenced on 14 lanes and distributed across two flow cells [16].

We used the same qRT-PCR datasets described in Chapter 5 Section 5.3.1.2 as a ”gold-standard” to evaluate the performance of the six combined p -value methods with a slight change of selecting the genes. Since all genes in this dataset were classified as present (P) if they were detected above threshold and absent (A) if they were not, only genes with a ”P” flag across the four replicates of each sample were used in this study. We also removed genes that do not correspond to unique RefSeq identifiers. The expression level of each gene was computed the same as we did in Section 5.3.1.2. As a result, 313 genes fall in the true DE set and 128 in the true non-DE set. Genes with absolute log-fold change > 0.2 and < 2 are discarded and not used in this study.

6.2.2.2 Marioni’s Liver and Kidney Dataset

This data, generated by Marioni *et al.* [100], is widely used for evaluating the performance of RNA-Seq developed approaches. The goal of Marioni’s study was to assess the technical variance within and between runs by estimating gene expression differences between human liver and kidney RNA samples using multiple technical replicates. Each sample was sequenced in seven lanes distributed across two runs of the machine and two different cDNA concentrations (1.5pM, 3pM) using an Illu-

mina Genome Analyzer. Only data sequenced at 3pM (five lanes per sample) cDNA concentration was used in this study. Table 6.1 shows more information about this data. The raw data for both liver and kidney along with 17,708 Ensembl transcripts, mapped with the array probes, were obtained. Ensembl transcripts that are expired or do not exist in the most current annotation version were removed resulting in 17,001 Ensembl transcripts. To improve the quality of this data, raw 36bp reads were trimmed to 32bp before mapping as advised by the authors.

6.2.3 Differential Expression

The IBSeq approach was used to detect DEGs between the two samples for each dataset. First, all raw sequencing reads in the five datasets were mapped to the indexed reference genome (hg19) using **Bowtie** version 1.0.1 [79] with the default parameters allowing for two mismatches, given that individual reads are ≤ 50 bp. Note that in instances with longer reads, **Bowtie2** provide a more optimal approach. We applied **Bowtie2** as well with no significant differences (results not shown). For each dataset, the SAM alignment files resulting from mapping were converted into BAM format and fed into the IBSeq for island differential expression. The IBSeq was then used to test for island DE between the two samples (Brain and UHR for MAQC data and liver and kidney for Marioni) in each dataset with the following parameters:

```
--window 35 --t-pvalue 0.05 --p-value 0.05 --gap 1
```

where **--window** is the window size, **--t-pvalue** is the p -value used to calculate the classification threshold, **--p-value** is the p -value for determining significant differential expression, and **--gap** is the gap size.

To detect for gene DE, the p -values for islands corresponding to each gene in the annotation files were combined using the six combined p -value methods described in Section 6.2.1. For the MAQC datasets, the portion of qRT-PCR data we selected

in Section 6.2.2.1 with 313 genes in the true DE set and 128 in the true non-DE was used to compute the overall significance of each gene. For liver and kidney data, the Ensembl transcripts with 17,001 genes used in the original paper were used to detect for gene DE. The motivation for using the same Ensembl transcripts was to be able to conduct a valid comparison with the results of the original paper and assess the performance of each combined method.

6.3 Results

6.3.1 Results from Liver and Kidney

We evaluated the performance of the combined p -value methods using the liver and kidney dataset. As suggested by the authors to improve the data quality, four bases of each read sequence were trimmed resulting in a total of 32bp for each read. Even with trimming, the alignment rate for the two samples were not as good as expected with 57 % and 59 % for liver and kidney respectively. This low alignment was also reported in the original paper. Marioni [100] has conducted a gene differential expression analysis between the two samples using multiple sequencing replicates generated by Illumina GA and compared the results to Affymetrix arrays results using the same RNA samples. In their study, a set of 17,708 probe sets mapping uniquely to 17,708 genes (out of 32,000) obtained from Ensembl database v.48 were identified. By comparing five lanes of each sample, they identified 11,493 DE genes at FDR 0.1% from the Illumina sequencing data and 8,113 (81% of those were also detected from the Illumina) from the Affymetrix arrays. Given the fact that the alignment rate is low for the two samples, in our opinion, these numbers seem too large. however, our motivation of using this data is that it is widely used and has detailed information about the expressions in both Illumina and Affymetrix. To compare our results with the results presented in the paper, we first used IBSeq to

compare the two samples and compute the overall p -value for the 17,001 (700 genes were eliminated from the set because of their expiration in the database) genes in the Ensembl transcripts using the six combined p -value methods. Table 6.2 shows the number of differentially expressed genes detected by each method and the number of overlapping genes with both Illumina and Affymetrix results using p -value < 0.001 . Weighted-sum was eliminated from this analysis due to its poor performance.

Table 6.2: Differentially expressed genes detected by each method using p -value < 0.001 . Overlap column represents the overlap with both Illumina and Affymetrix results. The Overlap(%) column indicates the percentage of overlap out of the detected genes.

Method	DE Genes	Overlap	Overlap(%)	Novel
Marioni Affymetrix	7942	N/A	N/A	N/A
Marioni Illumina	10133	N/A	N/A	N/A
Fisher's	3734	2891	77%	843
Z-transform	1911	1464	76%	447
Weighted z -test	4733	3649	77%	1084
Minimum p -value	2414	1882	77%	532
Logit	2541	1951	76%	590

As expected, the Weighted z -test method performed the best with the highest number of detected genes and highest overlap with Illumina and Affymetrix results outperforming Fisher's method. This supports the argument of Chen [23] that the weighted z -test is superior to both Z-transform and Fisher's method. The minimum p -value and logit methods perform very similar with a slight improvement for logit. Z-transform on the other hand did not perform as well as the others. To enhance our conclusions and look at the performance of another approach, we ran Cuffdiff [154] on the same data. Cuffdiff only detected 302 genes. Of those, 207 (68%) were also detected by the other methods. Venn diagrams (Figure 6.2) show the overlap

between each combined p -value method and Marionni Illumina, Marioni Affymetrix, and Cuffdiff.

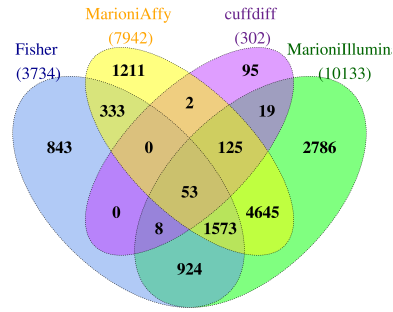
To see if the novel genes detected by each combined p -value method (Table 6.2, column 5) may have differential alternative splicing events between the two samples, we conducted an alternative expression analysis of liver and kidney using MISO (Mixture of Isoforms) [74]. As a result, 701 mutually exclusive exons (MXE) DE events were identified. To determine whether the novel detected genes are differentially spliced genes between the two samples, the 701 identified events were overlapped with the novel genes detected by each method. Table 6.3 shows the number of genes determined to be differentially spliced genes among the novel genes detected by each method.

Table 6.3: Differentially spliced genes for each method.

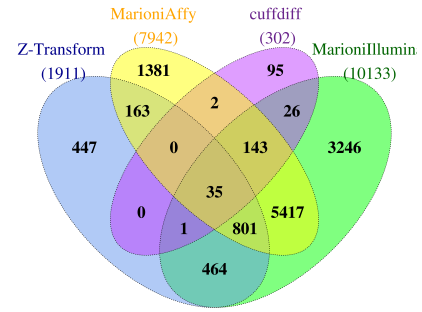
Method	Novel Genes	Differentially Spliced Genes
Fisher's	843	146
Z-transform	447	54
Weighted z -test	1084	148
Minimum p -value	532	98
Logit	590	102

6.3.2 Results from MAQC Datasets

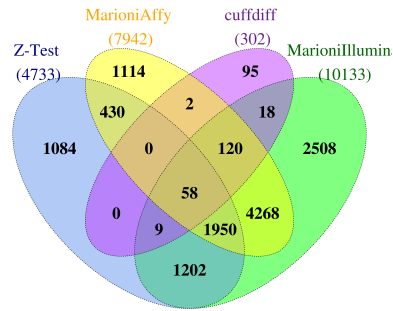
Next, using the four MAQC datasets along with qRT-PCR data, we evaluated the performance of the six combined p -value methods. For each dataset, we conducted a comparison analysis between the Brain and UHR samples. Using IBSeq, the islands for the two samples in each dataset were tested and a p -value was computed for each. Using the qRT-PCR data with 313 genes in the true DE set and 128 in the true non-DE set as a "gold-standard", each method was evaluated based on the number



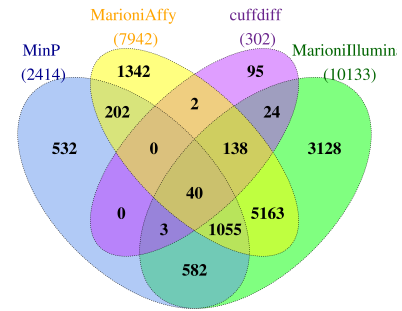
(a) Fisher's



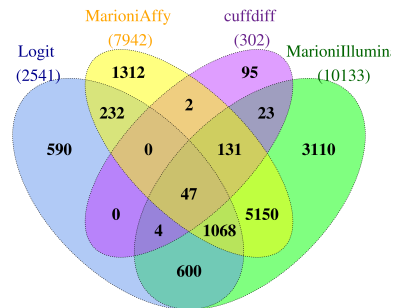
(b) Z-transform



(c) Weighted z -test



(d) Minimum p -value



(e) Logit

Figure 6.2: Overlap between the number of genes detected by each combined p -value method and Marioni's and Cuffdiff results.

of truly detected genes in both sets. Figure 6.3 shows bar plots for the true detection of each method for the four MAQC datasets.

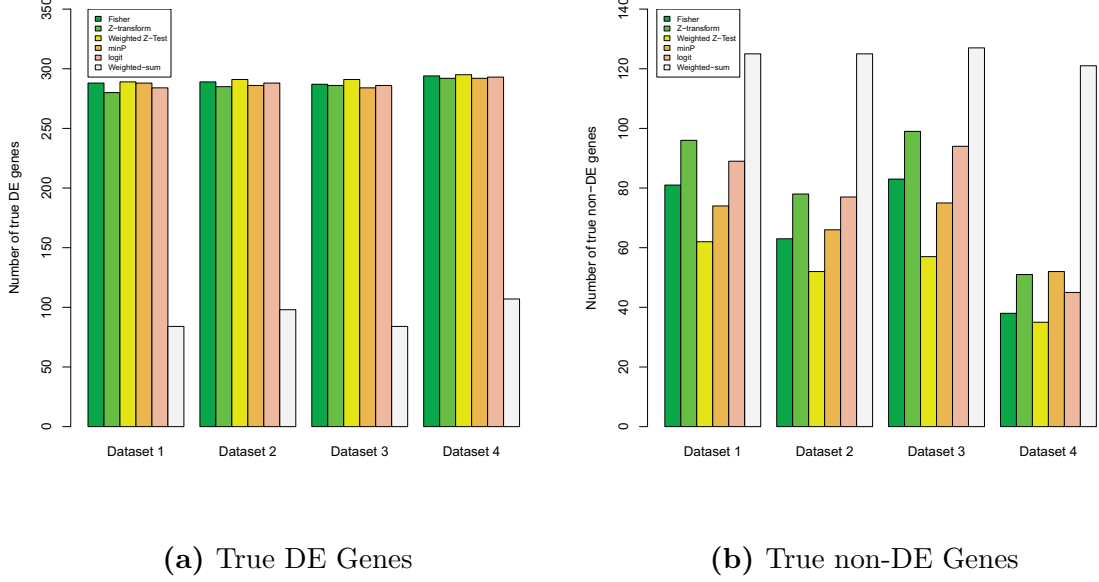


Figure 6.3: Number of true DE and true non-DE genes detected by each method for the four MAQC datasets using p -value < 0.05 .

As shown in Figure 6.3a, all compared methods except the Weighted-sum performed similarly in detecting the true DE genes for the four datasets with a slight outperformance by the Weighted z -test method. Out of 313 genes (the true DE set selected in Section 6.2.2.1), the five methods were able to detect between 89%-94% (280-295 genes) in the four datasets. In contrast, the Weighted-sum was too conservative and performed poorly with only 84, 98, 84, 107 true DE detected genes respectively. For detecting the true non-DE genes, Figure 6.3b shows a clear outperformance by the Weighted-sum method over the other methods. Out of 128 genes (the true non-DE set), the method was able to detect between 97%-99% (125-127 genes) in the four datasets. Surprisingly, the Weighted z -test and Fisher's method, which performed the best in detecting the true DE genes, did not perform as well in non-DE

detection. The Z-transform method had a better performance and outperforms the Weighted z -test and Fisher's method.

We then looked at the Receiver Operating Characteristic (ROC) to evaluate the relationship between sensitivity (TPR) and specificity (FPR) of each method. By using different p -value thresholds, we computed the true positive rate (TPR) and false positive rate (FPR) for each method in the four datasets. Thus, genes with p -values smaller than a given threshold are considered DE genes and genes with p -values greater than or equal to the threshold are considered non-DE genes. For each dataset, ROC curves (Figure 6.4) were generated for the six methods using different p -value cutpoints. The area under the ROC curve (AUC), shown in each plot and calculated using the trapezoidal rule, was used to measure the accuracy of each method and evaluate the performance for detecting DE genes.

As shown in Figure 6.4, the ROC curves show a similar performance for each of the six methods. By looking at the AUC of each method (Table 6.4), we observe that the performance was similar with a slight advantage of a certain method in each dataset. That outperformance was not significant enough to conclude that a specific method is the best among others. For example, in dataset 1, the logit method performed better than others with an AUC of 0.867. The performance of Fisher's method and Weighted-sum was very similar to the logit method with an AUC of 0.842 and 0.848, respectively. In dataset 2, Weighted-sum along with Fisher's performed the best with an AUC of 0.835 and 0.831 respectively, outperforming the Z-transform slightly. For dataset 3, the performance of Weighted-sum was still the best with an AUC 0.858, outperforming the Z-transform (AUC=0.837) and logit (AUC=0.830). Finally, for dataset 4, Z-transform performed the best with an AUC of 0.828. Therefore, the compared methods performed similarly with a slight advantage to the Weighted-sum method.

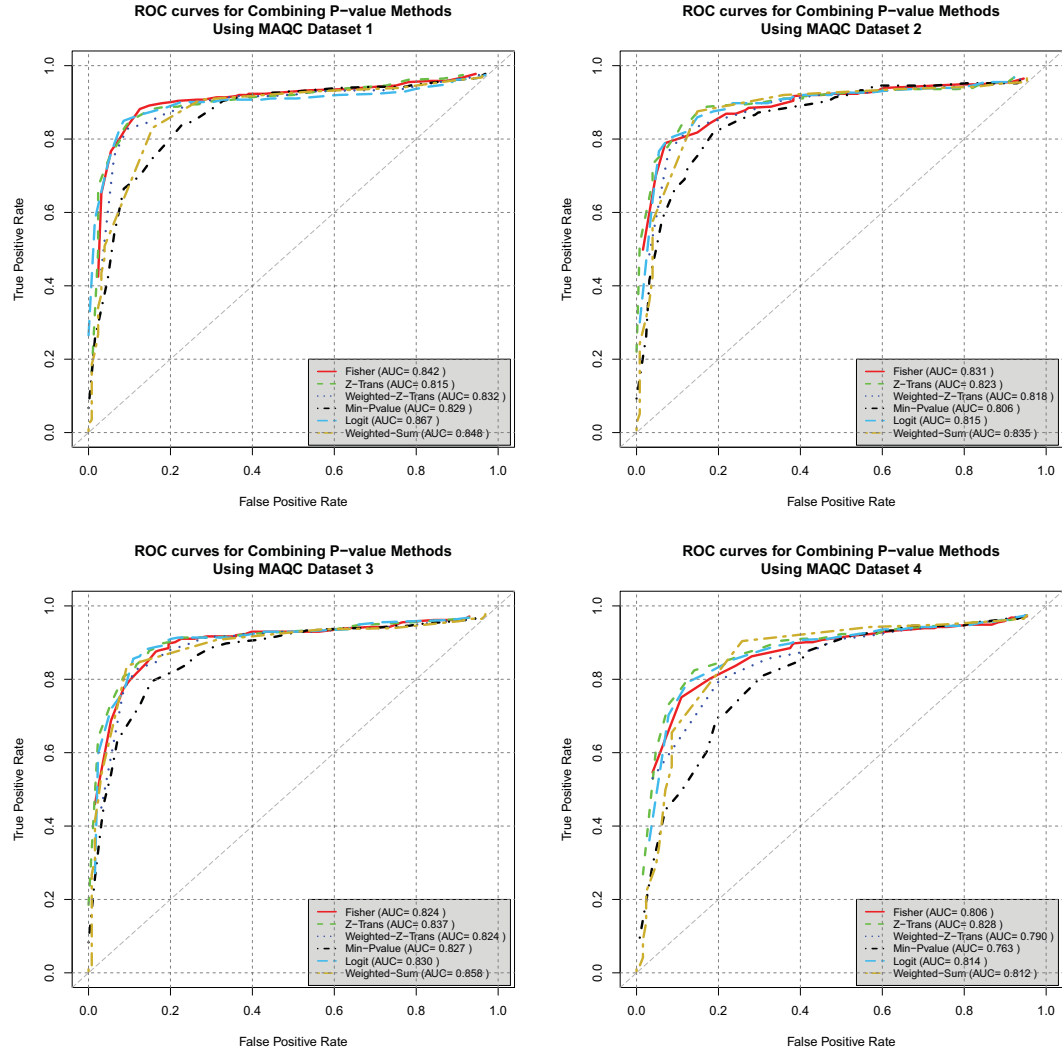


Figure 6.4: ROC curves for the six combined p -value methods on the MAQC datasets.

Table 6.4: AUC for each method on the four MAQC datasets.

Methods	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Fisher's	0.842	0.831	0.824	0.806
Z-transform	0.815	0.823	0.837	0.828
Weighted z -test	0.832	0.818	0.824	0.790
Minimum p -value	0.829	0.806	0.827	0.763
Logit	0.867	0.815	0.830	0.814
Weighted-sum	0.848	0.835	0.858	0.812

6.4 Conclusion

In this comparative study, the performance of a number of combined p -value methods on RNA-Seq data were assessed. Using five publicly available RNA-Seq datasets, we compared the ability of six combined p -value methods: (1) Fisher's, (2) Z-transform, (3) Weighted z -test, (4) Minimum p -value, (5) Logit, and (6) Weighted-sum methods for detecting differentially expressed genes. Applying the six methods on MAQC datasets shows a similar performance for detecting the true DE genes with an exception of the Weighted-sum method's poor performance. Only the Weighted z -test slightly outperformed the other methods. In contrast, the Weighted-sum method performed the best in detecting the true non-DE, clearly outperforming the other methods. When looking at the AUC in Figure 6.4, we see that the Weighted-sum method was at or near the best performance. Unexpectedly, the Weighted z -test did not perform as well as Fisher's, Z-transform, and logit methods. However, with the liver and kidney dataset, the Weighted z -test has performed the best among others and has reported the highest number of detected genes and overlap with Marioni Illumina and Affymetrix results.

CHAPTER 7

DISCUSSION AND CONCLUSIONS

In this dissertation, a novel Island-Based approach, IBSeq, for RNA-Seq differential expression analysis was developed as an attempt to mitigate some of the limitations associated with the current state of the art DE methods. IBSeq was developed in a way that no prior information of transcripts is needed and only raw data (short read sequences) is required. However, with IBSeq, we still have the option of using the annotation if needed. The core process in the IBSeq is the segmentation methodology where individual genomic regions (islands) of expression are constructed based on windowed read counts and compared across biological conditions to determine differential island expression. To determine if biological features are significantly differentially expressed across samples, the significance of islands corresponding to each feature are combined using six combined p -value methods. We presented a detailed description of this approach and illustrated how IBSeq is used to detect differences in expression without requiring any form of annotation. To assess the performance of this approach, we conducted a gene differential expression analysis and compared the results to a number of current DE methods using several publicly available benchmark RNA-Seq datasets. Using ROC curves and the area under the ROC curve (AUC) as performance metrics, IBSeq was able to perform better than other methods particularly detection of true DE events. However, in detecting the true non-DE genes, IBSeq did not perform as well as the other methods, generating more false positive detections (type I error). This has led us to conduct more investigation and look for

possible improvements. Examples of these investigations (presented below) are the parameters determination and further island segmentation.

7.1 Parameters-Determination Analysis

Since IBSeq is based on the use of several parameters and thresholds, the detection accuracy will consequently be based on the values of those parameters. Thus, choosing the optimal values would provide the best performance. Examples of IBSeq parameters that need more investigation are:

1. The window size used for splitting the genome.
2. The p -value used for computing the classification threshold used to classify genome regions into *high* and *low* density regions.
3. The number of *low* density regions included in the construction of *high* density island (the gap size).

In order to improve the detection accuracy, a complete analysis to determine the best value of each parameter is required. In this section, we present the analysis we conducted to determine the best values for the three parameters mentioned above using the same four MAQC RNA-Seq datasets described in Table 6.1 and the qRT-PCR dataset described in Section 5.3.1.2.

7.1.1 Window Size Analysis

To determine the optimal window size for IBSeq, we performed a window-size analysis using seven different window sizes, 10, 20, 30, 35, 50, 60, and 100. Figure 7.1 shows the ROC curves generated for this analysis.

Generally, the best window size is the one that detects a high number of truly DE genes with fewer false positives. Therefore, using the four MAQC datasets, results

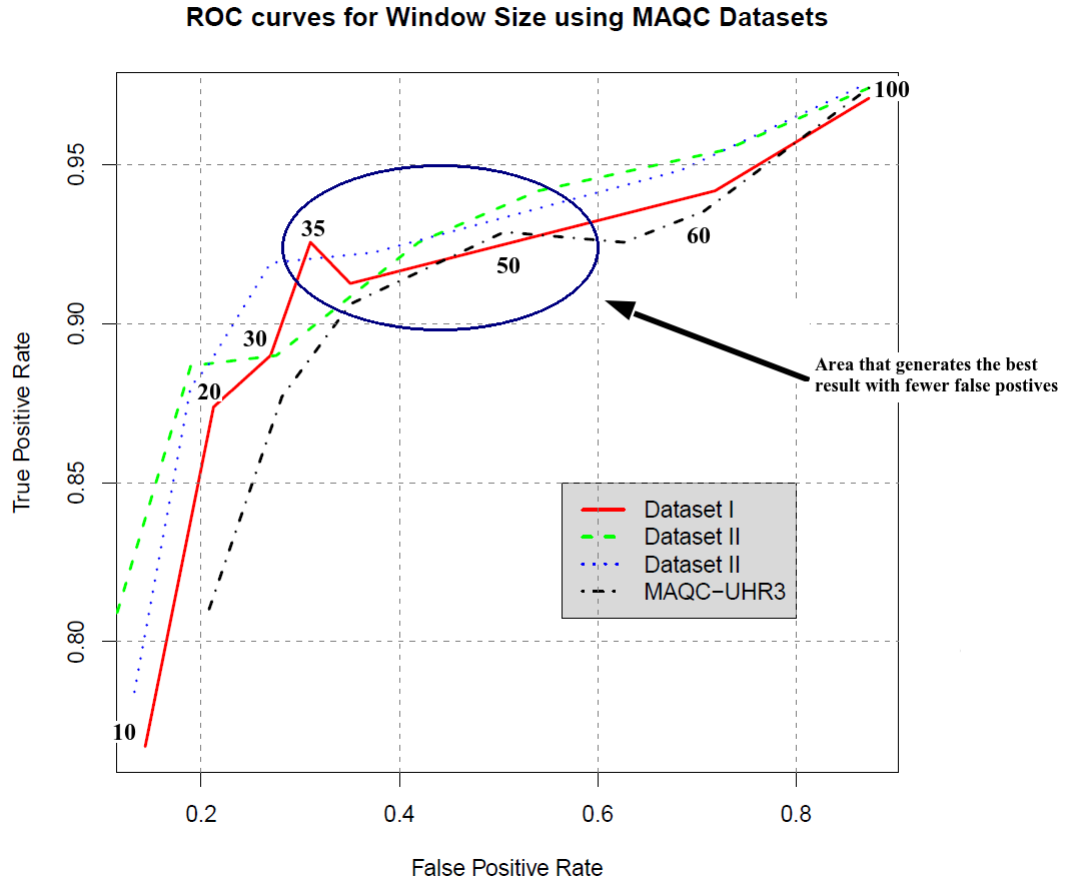


Figure 7.1: ROC curves for different window sizes. The blue oval represents the region that provides the best result.

show a window size in the range of 35-50bp (the blue oval region in Figure 7.1) works best. This may indicate a correlation between the window size and the read length since most NGS sequencers generate reads with a length close to this range.

7.1.2 Classification Threshold P -value Analysis

As discussed in Section 5.2, in order to compute the classification threshold to classify regions into high or low density regions, the user needs to provide a p -value. The computed threshold value will be based on this p -value. To determine what is the best p -value, we conducted a similar analysis to the one for window size using the

same MAQC datasets and seven different p -values, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2, and 0.5. Figure 7.2 shows for each MAQC dataset the true DE and non-DE genes detected by IBSeq.

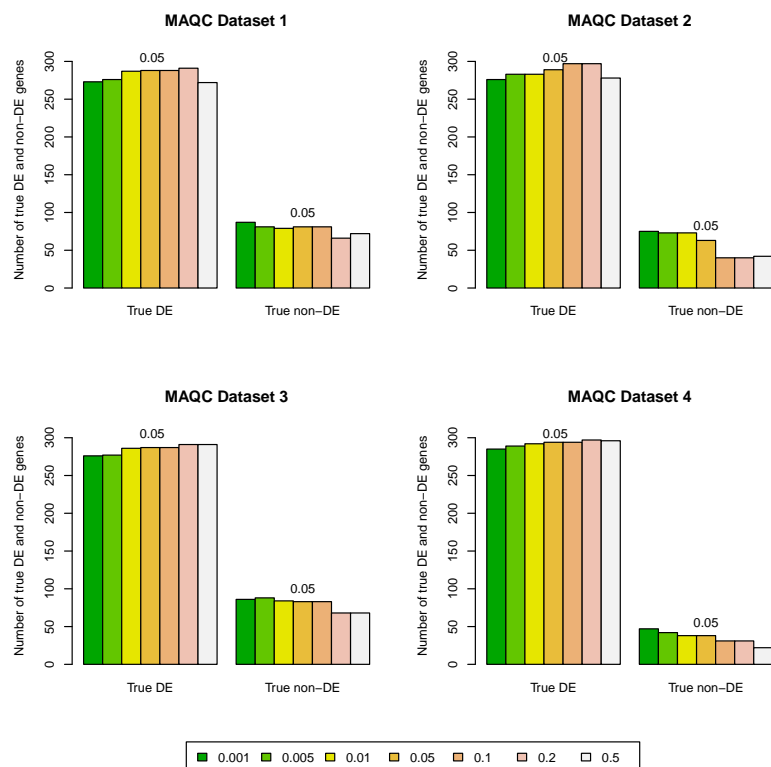


Figure 7.2: True DE and true non-DE detections using different p -values.

Obviously, the best p -value is the one that generates a threshold that provides the optimal trade-off between the true DE and true non-DE detections. This means it should detect a high number of true DE genes without increasing the false positive rate. From Figure 7.2, it is clear that a p -value of 0.05 provides the trade-off since using a p -value above 0.05 does not provide significant improvement for detecting the true DE but it does increase the false positive rate.

7.1.3 Gap Size Analysis

In order to determine how many low density islands are included in the construction of high density islands (the gap size), it is needed to run the IBSeq with different gap sizes. Therefore, we conducted a gap size analysis using the same MAQC datasets and five different gap sizes, 0, 1, 2, 3, and 4. Figure 7.3 shows the result of this analysis.

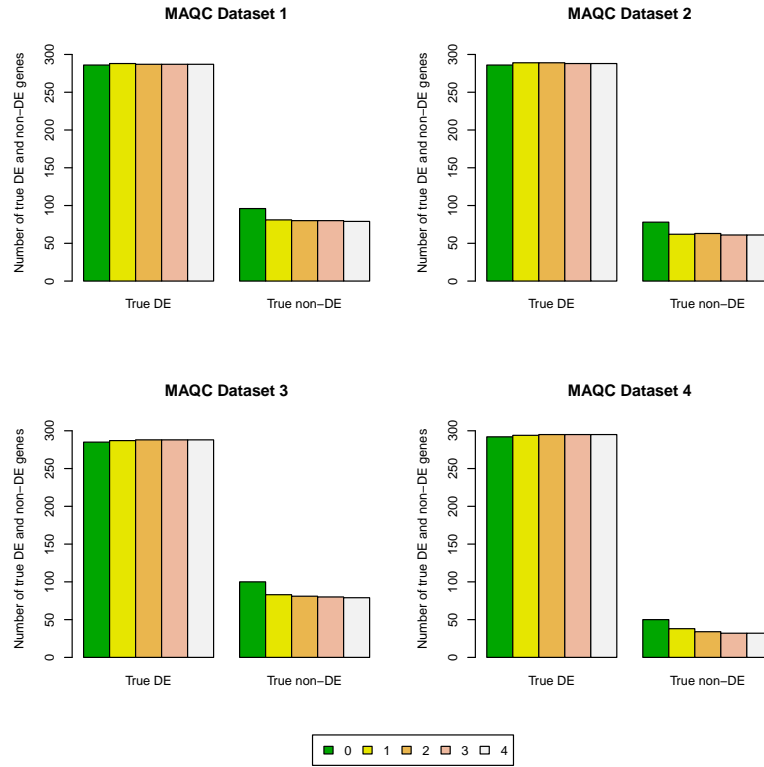


Figure 7.3: True DE and true non-DE detections using different gap sizes.

It is obvious from Figure 7.3 that the best result is generated when we do not include any low density regions (the gap size is 0) in the construction of high density islands. Once the gap size is increased (gap size ≥ 1), we see a clear increase of the false positive rate.

7.2 Further Island Segmentation

Island segmentation produces a variety of island sizes, with some belonging to a long category (2000-500,000 bases) while many others are small (< 2000 bases). After performing several experiments, it is observed that one reason for having high false positive rates is that long islands tend to overlap with more than one feature (e.g. gene). In order to alleviate this issue, we performed further segmentation on all islands exceeding 1000 bases.

The further segmentation algorithm uses the concept of standard deviation (SD) σ as follows. First, σ is computed for the initial segment (the first 10 regions in the island, R_1 - R_{10}). Second, the read counts of the next region (the eleventh region, R_{11}) is compared to 2σ . If the region count is greater than 2σ and the segment size is 10 (the segment should include at least 10 regions to be considered for further segmentation), we construct a new island from the initial segment and starts a new segment from R_{11} . If the condition is not true, the region R_{11} is added to the initial segment and a new σ is computed. This process continues until we reach the last region in the island. Figure 7.4 shows the algorithm flowchart and Algorithm 2 represents the corresponding pseudocode. To illustrate this process further, Figure 7.5 shows an example of further island segmentation using a 30bp window size. Applying this algorithm on the IBSeq approach using the same four MAQC datasets, results show a significant improvement in reducing the false discovery rate without effecting the true DE detections as shown in Figure 7.6.

Another important issue associated with IBSeq that needs more investigation is the combined p -value method used to combine the significance of islands corresponding to each feature. This process is very important since the final result of detecting differentially expressed features between samples is based on the combined p -value method used. To evaluate the performance of the six combined p -value meth-

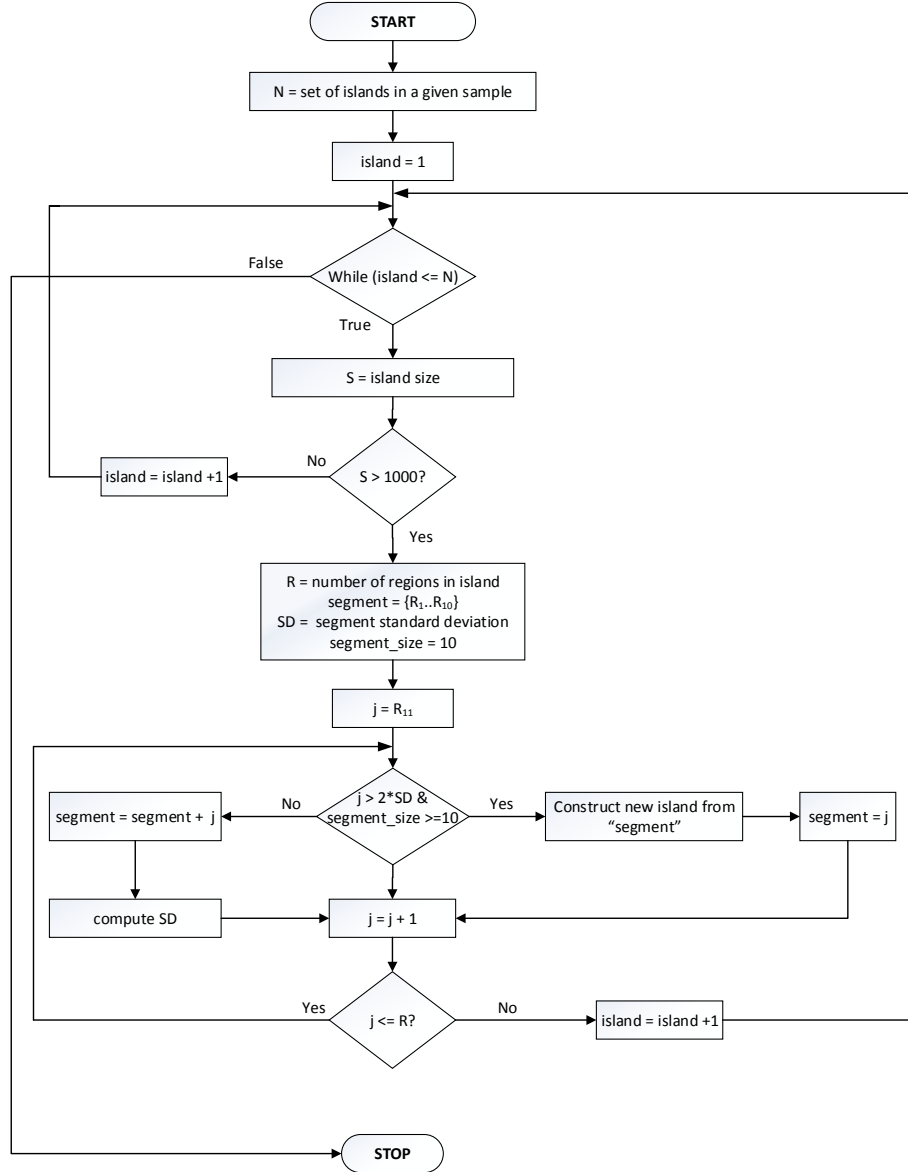


Figure 7.4: Further segmentation algorithm flowchart.

ods, we conducted a comparative study presented in Chapter 6. As a result of this study, we can conclude that the different combined p -value methods in general perform similarly. However, given the slight increase in performance of the Weighted z -test method for detecting the true DE on MAQC data and the best performance on liver and kidney data, we determined that it is important to assign weights to the combined tests because of their significant effect on the process of combining indepen-

Algorithm 2 Further Island Segmentation Algorithm

Input: a set of m islands I for n samples S

$$I = \{\{I_{s_{1,1}}, \dots, I_{s_{1,m}}\}, \{I_{s_{2,1}}, \dots, I_{s_{2,m}}\}, \dots, \{I_{s_{n,1}}, \dots, I_{s_{n,m}}\}\}$$

Output: A set of further segmented islands I across samples S .

```
1: while  $I \neq \emptyset$  do
2:   for each sample  $s_i \in S$  do
3:      $z \leftarrow I_{s_i}.size$ 
4:     if  $z > 1000bp$  then
5:        $R \leftarrow$  number of regions in island  $I_{s_i}$ 
6:        $segment = \{R_1..R_{10}\}$ 
7:        $sd =$  standard deviation of  $segment$ 
8:        $segSize \leftarrow 10$ 
9:       for  $i \leftarrow R_{11}$  to  $R$  do
10:        if  $R_i > 2*sd$  and  $segSize \geq 10$  then
11:          construct a new island from  $segment$ 
12:           $segment = R_i$ 
13:        else
14:           $segment = segment + R_i$ 
15:           $sd = sd$  of  $segment$ 
16:        end if
17:      end for
18:    else
19:      continue
20:    end if
21:  end for
22: end while
```

dent tests. This conclusion has led researchers to propose extensions [48] to Fisher's method to include weights for the tests.

Each combined p -value method has its own advantages and drawbacks. For instance, Fisher's method suffers from a significant drawback in an asymmetric sensitivity to small p -values compared to large p -values [162]. Other methods may have this drawback as well. To handle this issue, we considered approaches to minimize the influence of outliers. These approaches include (1) trimming p -values using a quantile approach, and (2) capping individual p -values with a minimum p -value threshold. Preliminary results suggest that only considering a median percentile provides marked

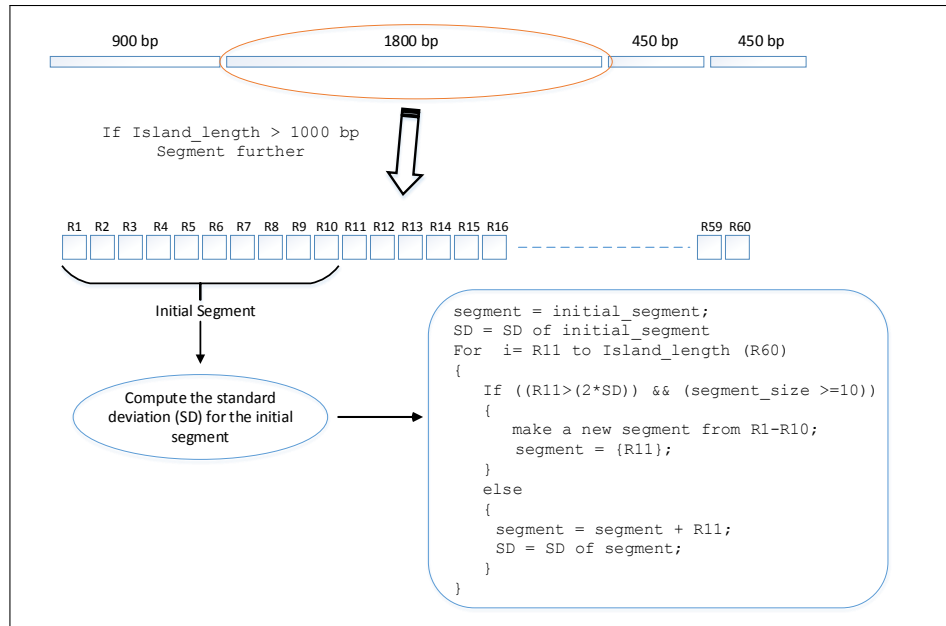


Figure 7.5: Further island segmentation example ($w = 30$ bp).

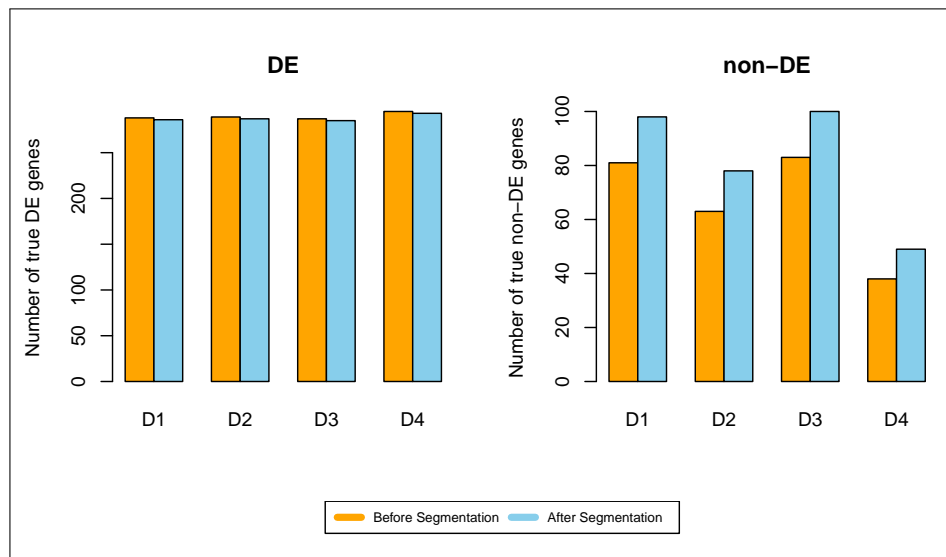


Figure 7.6: The performance of further island segmentation algorithm.

improvements in reducing the false discovery rate (Figure 7.7) with a slight reduction in detecting true positives for the Fisher's method. In contrast, thresholding the p -value provides a minimal improvement reducing the false discovery rate by 4-6 genes.

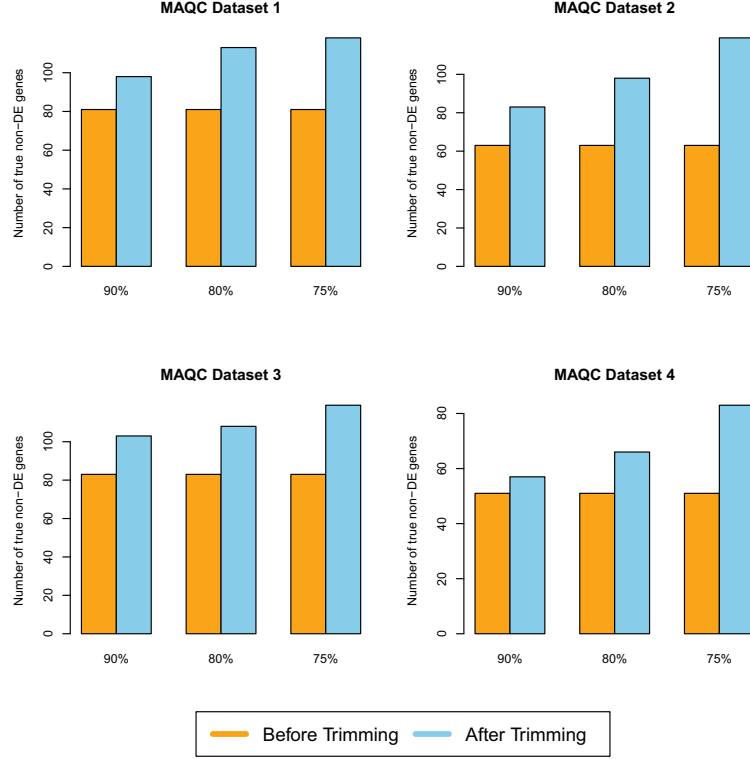


Figure 7.7: Improvement of detecting the true non-DE for Fisher’s method using median percentile approach.

One of the key assumptions with the combined p -value methods investigated in this study is that the p -values for a given sample (in this case, a gene) are independent. However, since an individual gene is composed of several islands and their expression (partially at the exon level) are likely to be correlated, at least in terms of the biological results, their p -values are not strictly independent. Thus, methods that can combine dependent p -values, such as Brown’s [15] and Kost’s [77] methods, may yield more consistent results.

All of the combined p -value methods considered in this study provide a number of false positive results (not shown). One potential reason for non-trivial false positive rates is that the datasets used in this study look at DE at a whole transcript level. However, by breaking up the transcripts into additional islands, it is possible to detect

alternative splicing events, which are likely to make up a significant amount of the false positive data. However, neither the MAQC project nor Marioni’s Liver and Kidney dataset provide any validation of alternative isoforms, and therefore, this was excluded from additional interrogation.

Although the main aim of this dissertation is to develop an RNA-Seq differential expression analysis method that provides a solution for the limitations associated with the current methods, there are still other factors (we refer to them here as *bioinformatics challenges*) which may effect the results regardless of which DE method is used. Thus, despite the advantages NGS technologies have brought to the -omics community, particularly in the transcriptomic realm, a number of new challenges have been introduced as well. Below, we list some of the challenges that have been introduced to the RNA-Seq community (there may be other challenges in other areas but since the focus of this dissertation work is limited to RNA-Seq, we focus only on the challenges in this area).

1. **Mapping Uncertainty:** considering short read sequences, some reads will map equally well to multiple locations on the reference genome which may effect the analysis results. As an example, non-unique regions within genes (such as domains or conserved family features) may show up as arbitrary under or over represented due to mapping.
2. **Transcriptome Reconstruction:** as many eukaryotic genes can produce different transcripts that encode for different isoforms and considering the reads are short, it is hard to determine which reads originate from which isoforms. In addition, low expressed genes (genes represented by a few reads) will be, in most cases, discarded and their transcripts will not be assembled. Furthermore, identifying mature transcripts is a difficult task since some reads originate solely from exons [44].

3. **Sequencing Depth:** to detect and measure RNA-Seq transcripts, one needs to decide on the sequencing depth. This is an issue of coverage versus cost. To provide better expression estimate, more coverage, which requires more sequencing depth, is needed but this will result in more cost. Sequencing depth is very important factor that needs more attention since additional reads results in the identification of low-abundance regions or transcripts, and provides a more accurate picture of the actual dynamic range of expressed transcripts.

7.3 Computational and Space Complexity

To measure the efficiency and feasibility of any computer algorithm, the computational complexity in terms of time (the amount of *time* required to run the algorithm) and space (the amount of *memory* required to run the algorithm) has to be measured and estimated. This is critical in considering scalability, particularly with the likelihood that the input size is large as the case with the next-generation sequencing data IBSeq uses. Generally, a complexity analysis is based on counting primitive operations an algorithm needs to perform. Thus, the number of steps IBSeq takes as a function of the input size needs to be measured and assessed in terms of its efficiency by measuring the upper bound amount of time and space required to execute the algorithm. Please refer to Algorithm 1 and Figure 5.2 throughout this discussion as they represent the IBSeq steps discussed here.

Let us first define the notations involved in this analysis as follows:

- N : number of samples (range from 2-10).
- M : number of replicates (range from 3-5).
- R_{nm} : reads per file (range from 30-80 million).
- I_{nm} : number of pre-islands in sample n , replicate m .
- G : genome size (~ 3 billion base pairs).
- L : read length (range from 35-100).
- W : window size (range from 30-50bp).

For most RNA-Seq studies, there will be a number of N samples where each sample may have M replicates, R reads, and a genome of length G as an input. Note that in most cases, the genome size is constant, on the order of 3 billion base pairs (bp). While the sample size R and the genome length G may not be constants, for most RNA-Seq projects, R will range from 30-80 million reads, while G is typically around 3 billion base pairs (bp). Thus, in the worst case scenario (the upper bound on algorithm performance), the *mapping* process will run in $NM \times (LR + LG)$ time depending on the number of reads R and therefore the upper bound time complexity of this step is on the order of $NM \times L(R + G)$ or $\mathcal{O}(NM \times L(R + G))$. Given that $M \approx N$ and L is a constant which can be discarded, the time complexity is roughly $\mathcal{O}(N^2(RG))$. In addition, the genome length G can be discarded since it is a constant in most cases as discussed above. Therefore, the time complexity of the mapping process function is $\mathcal{O}(RN^2)$. As for the space complexity, this step will be roughly based on the genome size G . Although the mapping step is not part of IBSeq, the computational and space complexity of this step is discussed here since it is an essential process of any RNA-Seq analysis and it is assumed to be performed before using IBSeq. Refer to [139] for a more in-depth discussion about mapping complexity.

The *per base count* step (step 1 of IBSeq) will run in $(NM \times G)$ time depending only upon the genome size G . Similar to mapping, given that $M \approx N$ and G is a constant, the time complexity of this step can be $\mathcal{O}(N^2)$ (approximately quadratic time). Similarly, the *region construction* process (step 2) will run in $(NM \times G/W)$ time. Considering G and W are constants and can therefore be discarded and $M \approx N$, the time complexity of this step is on the order of N^2 or $\mathcal{O}(N^2)$. The space complexity of the two steps clearly depends on the genome size G . The *island construction* process (step 3) will be treated the same as region construction step and will run in $(NM \times G/W)$ times. Given the constants G and W , the time complexity of island

construction is also a quadratic on the order of N^2 or $\mathcal{O}(N^2)$.

In the second part of the IBSeq algorithm where all possible pairwise comparisons between the samples are performed to test for *island differential expression* (step 4), the process runs in $I \times \left(\frac{NM(NM-1)}{2} \right)$ time. The complexity function here is $I \times \left(\frac{1}{2}(NM)^2 - \frac{1}{2}NM \right)$ but since $\frac{1}{2}NM$ is much smaller than $\frac{1}{2}(NM)^2$ and has no significant effect on the running time, this term can be discarded. In addition, we can also discard the constant $\frac{1}{2}$ since it takes a constant amount of time and insignificant for the growth function. Therefore, the time complexity of this part in the worst case scenario is on the order of $I \times (NM)^2$ or $\mathcal{O}(I(NM)^2)$ which is a quadratic complexity. This is clearly the larger than the time complexity of all other previous steps.

The last step which *combines the significance of DE islands* corresponding to each feature (step 5) will run in the same time as step 4 with an order of $\mathcal{O}(I(NM)^2)$ with an additional parameter C_{sig} which represents the complexity of the combined p -value method used. Thus, the time complexity in this context will be $\mathcal{O}(I(NM)^2 \times C_{sig})$ where C_{sig} represents steps in the combined p -value method used. Similar to the previous steps, the space complexity of step 4 and 5 will be in most cases dependent on the genome size.

Overall, the time complexity of IBSeq algorithm can be written as a total of the time complexities of all steps as the following:

$$\underbrace{\mathcal{O}(RN^2)}_{\text{preprocessing}} + \underbrace{\mathcal{O}(N^2)}_{\text{Step 1}} + \underbrace{\mathcal{O}(N^2)}_{\text{Step 2}} + \underbrace{\mathcal{O}(N^2)}_{\text{Step 3}} + \underbrace{\mathcal{O}(I(NM)^2)}_{\text{Step 4}} + \underbrace{\mathcal{O}(I(NM)^2) \times C_{sig}}_{\text{Step 5}}$$

preprocessing: mapping read sequences to the reference genome.

Step 1: compute per-base count.

Step 2: region construction.

Step 3: island construction.

Step 4: island differential expression.

Step 5: combining the significance of islands.

It is obvious that the time complexity of IBSeq is a summation of quadratic complexities which shows a potential room for improvement. However, it can be acceptable and feasible in our case because of two reasons: (1) most RNA-Seq studies use a small number of samples N (due to sequencing cost), (2) since the number of samples is a small (range from 2-10), the growth rate of the complexity will not grow fast where it will start increasing only with larger values of N (e.g. $N > 50$). The space complexity for almost all IBSeq steps will roughly be based upon the genome size.

The complexity analysis discussed above is an asymptotic upper bound estimation of IBSeq efficiency and does not indicate the actual performance of the algorithm. In order to perform an actual measurement, IBSeq needs to be run using real data and the amount of both CPU time and memory space required to perform each step has to be recorded. Therefore, we performed an actual assessment of the IBSeq requirements of time and space using four RNA-Seq datasets described in Table 7.1.

Table 7.1: Description of the datasets used in the complexity analysis.

Dataset	# of Samples	Read Length	Total Reads	Organism
MAQC2 [16]	2	35bp	hbr: 81,250,500 uhr: 92,524,365	Human
MAQC2 [113]	2	50bp	hbr: 53,238,798 uhr: 59,561,348	Human
Marioni's data [100]	2	36bp	Liver: 69,618,202 Kidney: 66,404,506	Human
Petruska's data [59]	3	59bp	T0: 63,623,836 x 2 T7: 48,485,234 x 2 T14: 41,050,145 x 2	Rat

IBSeq was implemented using Perl version v5.14.2 and run on a Dell Alienware Area-51 Gaming workstation with Intel(R) Core(TM) i7 CPU 930 @ 2.80GHz

(8 CPUs/4 cores) and 12 GB RAM. The CPU time and memory usage were recored for each step. Table 7.2 shows the amount of time and space usage for each step of the IBSeq algorithm while Figure 7.8 represents the corresponding charts.

Table 7.2: The amount of time and space recorded for each step in IBSeq using four RNA-Seq datasets (D=Dataset).

Process	CPU Time (Minutes)				Memory (Kilobytes)			
	D1	D2	D3	D4	D1	D2	D3	D4
Mapping	27	55	24	34.5	3613000	3369000	3530000	2664000
Per-base count	154	168	154	153	1902000	1902000	1902000	2045000
Region construction	99	101	99	92	7520	7516	7524	7524
Island construction	18	17	15	18	1453000	1452000	1455000	1357000
Island DE	82	56	138	52	3095000	3095000	3095000	3342000
Combine significance	165	135	219	115	14960	14820	15708	25648

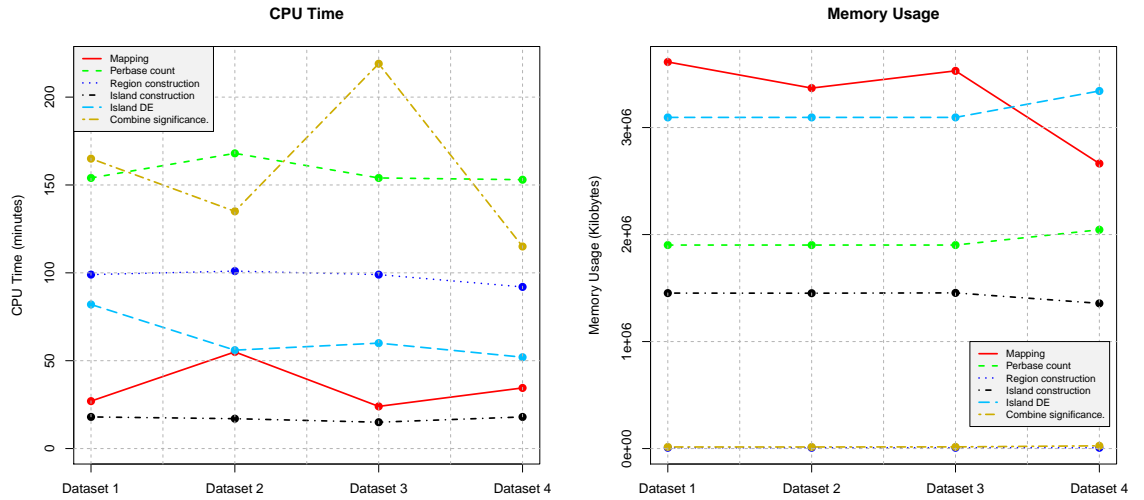


Figure 7.8: The amount of time and space utilized by each step in IBSeq using four RNA-Seq datasets.

By looking at the charts above, it is clear that the island construction step took the lowest running time which was expected since this step does not require

intensive computations. In terms of memory utilization, the performance of this step utilized only a small fraction of the memory (12%). Although the mapping step took the second lowest running time, it utilized the largest space of memory ranging from 22%-30%. This was expected since this process needs to load the indexed reference genome, which is usually in GBs, into the memory. In terms of the best process that took the smallest memory space, the region construction step along with combining the significance utilized the smallest space with only 0.1%. However, combining the significance along with per base count step took the largest running time. Although the running time for island DE process was small, it utilized the largest memory space along with the mapping with 25% of the physical memory. In conclusion, it is observed that the per base count and combining the significance processes require the largest running time. This result is expected since the per base count is based on the length of reference genome which is in most cases in the range of 3 billion bases. Similarly, the computation of combining the significance of genetic features is based on the number of annotated features which is usually large (ranging from 18000-80000). In contrast, mapping and island DE processes require the largest memory space. This observation is to be expected since the mapping process requires the large reference genome index to be loaded into the memory. Similarly for island DE, in order to perform the DE test, we need to load the large region files into the memory to extract all island counts. Therefore, these processes that take the largest running time and largest memory space are considered for further optimization in order to minimize the requirement of both time and space.

7.4 Future Directions

There are several exciting directions for future research inspired by this dissertation. In this chapter, we summarize briefly some of the potential research directions.

7.4.1 Potential IBSeq Extensions

Since IBSeq is in its initial version, a number of potential extensions need to be implemented in the near future to improve the efficiency and robustness including:

1. *Alternative splicing detection*: currently, IBSeq does not support the detection of alternative splicing events. Since most genes composed of multiple exons in eukaryotic have multiple isoforms, alternative splice detection is important in order to understand subtle differences that occur at a transcript level. As an initial approach to detect alternative splicing events, transcripts could be broken up into additional islands which are then stitched together to check for expression differences between isoforms within the same gene.
2. *Biological variation*: IBSeq as for now is capable of detecting island differential expression using technical replicates only. We plan in the future to adapt the current statistical model in order to accommodate for the biological variation of genetic features across biological replicates (IBSeq_bio).
3. *Visualization*: the current form of IBSeq results is a tab-delimited file containing all statistical and expression information which can be visualized on one of the genome visualization tools such as UCSC genome browser, Integrative Genomics Viewer (IGV), or GBrowse. In the future, we plan to develop a visualization package adapted specifically for IBSeq results.

7.4.2 Combining the Significance of Islands

As discussed in Section 6.4, one of the major assumptions with the combined p -value methods studied is that the p -values for a given sample (in this case a gene) are independent. But since an individual gene is composed of several islands, there is a strong likelihood of correlation between expression of islands and therefore their

p -values are not strictly independent. In the future, we plan to consider methods that can combine dependent p -values such as Brown’s [15] and Kost’s [77] which may yield more consistent results. In addition, we also plan to continue exploring additional weighted measures in order to provide realistic p -value combinations. Furthermore, We hope to continue further analysis to look at the effects of each of the combined p -value methods on alternative splice detection in the future.

7.4.3 Comparison to Transcriptome Assemblers

The main focus of this research was to extend the knowledge of differentially expressed regions outside of known annotations. While this may be a fruitful approach for *de novo* transcriptome discovery, we have yet to compare it to *de novo* transcriptome assemblers such as Trans-Abyss [130], Oases [137], or Trinity [49]. This is due to the fact that our IBSeq approach as currently constructed is a mapping-based methodology in contrast to these assembly-based methods. In the future, we will consider an IBSeq-based methodology to *de novo* transcript assembly.

7.4.4 IBSeq Optimization

Since IBSeq consists of several steps, the amount of time and space required to execute each step varies from one to another. For instance, the computation of per base count step is very expensive in terms of both time and space since it needs to compute the number of reads mapping to each position in the genome. To achieve a drastic improvement in speed and reduce the running time of IBSeq algorithm, we plan to implement a new version of IBSeq that can be run on one of the massively parallel computing platforms such as the graphics processing unit (GPU) which provides massively parallel computational power that assures the speed of algorithms. This can be performed using CUDA (Compute Unified Device Architecture) C/C++ to

program GPUs. Since the various steps of IBSeq are performed for each sample, steps can run in parallel which is expected to speed up the algorithm in large magnitude.

REFERENCES

- [1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403-410, October 1990.
- [2] Amino acid structure. Available at: http://en.wikipedia.org/wiki/Amino_acid.
- [3] S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10):R106, October 2010.
- [4] S. Anders, A. Reyes, and W. Huber. Detecting differential usage of exons from RNA-seq data. *Genome Research*, 22(10):2008-2017, October 2012.
- [5] S. Anders, P. T. Pyl, and W. Huber. HTSeq—A Python framework to work with high-throughput sequencing data. *Bioinformatics*, 31(2):166-169, September 2014.
- [6] M. Aschoff, A. Hotz-Wagenblatt, K. H. Glatting, M. Fischer, R. Eils, and R. König. SplicingCompass: differential splicing detection using RNA-seq data. *Bioinformatics*, 29(9):1141-1148, May 2013.
- [7] P. L. Auer and R. W. Doerge. A Two-Stage Poisson Model for Testing RNA-Seq Data. *Statistical Applications in Genetics and Molecular Biology*, 10(1):1-26, May 2011.
- [8] S. Batzoglou, D. B. Jaffe, K. Stanley, J. Butler, S. Gnerre, E. Mauceli, B. Berger, J. P. Mesirov, and E. S. Lander. ARACHNE: a whole-genome shotgun assembler. *Genome Research*, 12(1):177-189, January 2002.

- [9] A. D. Baxeavanis and B. F. F. Ouellette. Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins. *John Wiley & Sons*, Inc., Second edition, 2001.
- [10] 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, 57(1):289-300, 1995.
- [11] B. Blencowe. Alternative Splicing: New insights from Global Analyses. *Cell*, 126(1):37-47, July 2006.
- [12] R. Bohnert and G. R  tsch. rQuant.web: a tool for RNA-Seq-based transcript quantitation. *Nucleic Acids Research*, 38:W348-W351, May 2010.
- [13] F. Bona, S. Ossowski, K. Schneeberger, and G. R  tsch. Optimal spliced alignments of short sequence reads. *Bioinformatics*, 24(16):i174-80, August 2008.
- [14] A. Brazma, H. Parkinson, T. Schlitt, and M. Shojatalab. A quick introduction to elements of biology-cells, molecules, genes, functional genomics, microarrays. http://www.ebi.ac.uk/microarray/biology_intro.html, (Draft), 2001.
- [15] M. B. Brown. A method for combining non-independent, one-sided tests of significance. *Biometrics*, 31(4):987-992, December 1975.
- [16] J. H. Bullard, E. Purdom, K. D. Hansen, and S. Dudoit. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, 11(1):94, February 2010.
- [17] M. Burrows and D.J. Wheeler. A Block-sorting Lossless Data Compression Algorithm. *Technical Report 124* Palo Alto, CA: Digital Equipment Corporation; 1994.

- [18] J. Butler, I. MacCallum, M. Kleber, I. A. Shlyakhter, M. K. Belmonte, E. S. Lander, C. Nusbaum, and D. B. Jaffe. ALLPATHS: de novo assembly of whole-genome shotgun microreads. *Genome Research*, 18(5):810-820, May 2008.
- [19] E. Cha, K. L. Hoblitzell, and E. C. Rouchka. Alternative Splicing Events. *University of Louisville Bioinformatics Technical Report Series*, TR-ULBL-2007-03, December 2007.
- [20] M. J. Chaisson, D. Brinza, and P. A. Pevzner. De novo fragment assembly with short mate-paired reads: Does the read length matter. *Genome Research*, 19(2):336-346, February 2009.
- [21] M. J. Chaisson and P. A. Pevzner. Short read fragment assembly of bacterial genomes. *Genome Research*, 18(2):324-330, February 2008.
- [22] J. Chapman and J. Whittaker. Analysis of multiple snps in a candidate gene or region. *Genetic Epidemiology*, 32(6):560-566, September 2008.
- [23] Z. Chen. Is the weighted z-test the best method for combining probabilities from independent tests? *Journal of Evolutionary Biology*, 24(4):926-930, April 2011.
- [24] Chromosome Structure. *Access Excellence @ the National Health Museum*. Available at: <http://www.accessexcellence.org/RC/VL/GG/chromosome.php>.
- [25] P. J. 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 Research*, 38(6):1767-1771, January 2010.
- [26] J. Commins, C. Toft, M. A. Fares. Computational Biology Methods and Their Application to the Comparative Genomics of Endocellular Symbiotic Bacteria of Insects. *Biol Proced Online*, 11(1):52-78, March 2009.

- [27] V. Costa, C. Angelini, I. De Feis, and A. Ciccodicola. Uncovering the complexity of Transcriptomes with RNA-Seq. *Journal of Biomedicine and Biotechnology*, 2010:853916, April 2010.
- [28] R. d. Cousins. Annotated Bibliography of Some Papers on Combining Significances or p -values. Available at *arXiv:0705.2209v2*, December 2008.
- [29] F. Crick. Central Dogma of Molecular Biology. *Nature*, 227(5258):561-563, August 1970.
- [30] S. Datta, S. Datta, S. Kim, S. Chakraborty, and R. S. Gill. Statistical Analysis of Next Generation Sequencing Data: A Partial Overview. *J Proteomics Bioinform*, 3(6):183-190, June 2010.
- [31] F. Denoeud, J. Aury, C. Da Silva, B. Noel, O. Rogier, M. Delledonne, M. Morgante, G. Valle, P. Wincker, C. Scarpelli, O. Jaillon, and F. Artiguenave. Annotating genomes with massive-scale RNA sequencing. *Genome Biology*, 9(12):R175, December 2008.
- [32] Y. Di, D. W. Schafer, J. S. Cumbie, and J. H. Chang. The NBP negative binomial model for assessing differential gene expression from RNA-Seq. *Statistical Applications in Genetics and Molecular Biology*, 10(1):1-28, May 2011.
- [33] DNA replication at wikimedia commons. Available at: https://commons.wikimedia.org/wiki/File:DNA_replication_split_horizontal.svg.
- [34] DNA Sequencing-Maxam-Gilbert Method. Available at: <http://www.dnasequencing.org/maxam-gilbert>.
- [35] J. C. Dohm, C. Lottaz, T. Borodina, and H. Himmelbauer. SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. *Genome Research*, 17(11):1697-1706, November 2007.

- [36] B. K. Dredge, A. D. Polydorides, and R. B. Darnell. The splice of life: alternative splicing and neurological disease. *Nature Reviews Neuroscience*, 2(1):43-50, January 2001.
- [37] ENCODE Project Consortium. A user’s guide to the encyclopedia of DNA elements (ENCODE). *PLoS Biology*, 9(4):e1001046, April 2011.
- [38] B. Ewing and P. Green¹. Base-Calling of Automated Sequencer Traces Using Phred.II. ErrorProbabilities. *Genome Research*, 8(3):186-194, March 1998.
- [39] P. Ferragina and G. Manzini. Opportunistic data structures with applications. *In Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, IEEE, 390-398, 2000.
- [40] R. A. Fisher. Statistical methods for research workers. *Oliver and Boyd*, Edinburgh, London, 1970.
- [41] P. Flicek and E. Birney. Sense from sequence reads: methods for alignment and assembly. *Nature Methods*, 6, S6-S12, October 2009.
- [42] D. Gao, J. Kim, H. Kim, T. L. Phang, H. Selby, A. C. Tan, and T. Tong. A survey of statistical software for analysing RNA-seq data. *Human Genomics*, 5(1):56-60, October 2010.
- [43] J. Garbe. RNA-Seq Tutorial - Central dogma of molecular biology. *Minnesota Supercomputing Institute, University of Minnesota*, March 2012. Available at: <https://www.msi.umn.edu/sites/default/files/RNA-Seq%20Module%201.pdf>
- [44] M. Garber, M. G. Grabherr, M. Guttman, C. Trapnell. Computational methods for transcriptome annotation and quantification using RNA-seq. *Nature Methods*, 8(6):469-477, June 2011.

- [45] Genetic code. Available at: <http://en.wikipedia.org/wiki/File:RNA-codons.png>.
- [46] Genetics Home Reference. Available at: <http://ghr.nlm.nih.gov/handbook/basics/gene>.
- [47] P. Glaus, A. Honkela, M. Rattray. Identifying differentially expressed transcripts from RNA-seq data with biological variation. *Bioinformatics*, 28(13):1721-1728, July 2012.
- [48] I. J. Good. On the weighted combination of significance tests. *Journal of the Royal Statistical Society Series B*, 17(2):264-265, 1955.
- [49] M. G. Grabherr, B. J. Haas, M. Yassour, J. Z. Levin, D. A. Thompson, I. Amit, X. Adiconis, L. Fan, R. Raychowdhury, Q. Zeng, Z. Chen, E. Mauceli, N. Hacohen, A. Gnirke, N. Rhind, F. di Palma, B. W. Birren, C. Nusbaum, K. Lindblad-Toh, N. Friedman, and A. Regev. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology*, 29(7):644-652, May 2011.
- [50] R. P. Grant. Computational Genomics: Theory and Application. *Horizon Bioscience*, pp. 225-249, September 2004.
- [51] E. D. Green. Strategies for the systematic sequencing of complex genomes. *Nature Reviews Genetics*, 2(8):573-583, August 2001.
- [52] M. Griffith, O. L. Griffith, J. Mwenifumbo, R. Goya, A. S. Morrissy, R. D. Morin, R. Corbett, M. J. Tang, Y. C. Hou, T. J. Pugh, G. Robertson, S. Chittaranjan, A. Ally, J. K. Asano, S. Y. Chan, H. I. Li, H. McDonald, K. Teague, Y. Zhao, T. Zeng, A. Delaney, M. Hirst, G. B. Morin, S. J. Jones, I. T. Tai, and M.

- A. Marra. Alternative expression analysis by RNA sequencing. *Nature Methods*, 7(10):843-847, September 2010.
- [53] R. Gupta, I. Dewan, R. Bharti, and A. Bhattacharya. Differential Expression Analysis for RNA-Seq Data. *ISRN Bioinformatics*, 2012:817508, August 2012.
- [54] A. E. Guttmacher and F. S. Collins. Genomic Medicine-A Primer. *N Engl J Med*, 347(19):1512-1520, November 2002.
- [55] M. Guttman, M. Garber, J. Z. Levin, J. Donaghey, J. Robinson, X. Adiconis, L. Fan, M. J. Koziol, A. Gnirke, C. Nusbaum, J. L. Rinn, E. S. Lander, and A. Regev. Ab initio reconstruction of cell typespecific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. *Nature Biotechnology*, 28(5):503-510, May 2010.
- [56] B. J. Haas and M. C. Zody. Advancing RNA-Seq analysis. *Nature Biotechnology*, 28(5):421-423, May 2010.
- [57] J. Halvardson, A. Zaghlool, and L. Feuk. Exome RNA sequencing reveals rare and novel alternative transcripts. *Nucleic Acids Research*, 41(1):e6, August 2013.
- [58] T. J. Hardcastle and K. A. Kelly. baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics*, 11(1):422, August 2010.
- [59] B. J. Harrison, R. M. Flight, C. Gomes, G. Venkat, S. R. Ellis, U. Sankar, J. L. Twiss, E. C. Rouchka, and J. C. Petruska. IB4-binding sensory neurons in the adult rat express a novel 3'UTR-extended isoform of CaMK4 that is associated with its localization to axons. *Journal of Comparative Neurology*, 522(2):308-336, February 2014.

- [60] J. Harrow, F. Denoeud, A. Frankish, A. Reymond, C. K. Chen, J. Chrast, J. Lagarde, J. G. Gilbert, R. Storey, D. Swarbreck, C. Rossier, C. Ucla, T. Hubbard, S. E. Antonarakis, and R. Guigo. GENCODE: producing a reference annotation for ENCODE. *Genome Biology*, 7(Suppl 1):S4, August 2010.
- [61] J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski, B. L. Aken, D. Barrell, A. Zadissa, S. Searle, I. Barnes, A. Bignell, V. Boychenko, T. Hunt, M. Kay, G. Mukherjee, J. Rajan, G. Despacio-Reyes, G. Saunders, C. Steward, R. Harte, M. Lin, C. Howald, A. Tanzer, T. Derrien, J. Chrast, N. Walters, S. Balasubramanian, B. Pei, M. Tress, J. M. Rodriguez, I. Ezkurdia, J. van Baren, M. Brent, D. Haussler, M. Kellis, A. Valencia, A. Reymond, M. Gerstein, R. Guigó, and T. J. Hubbard. GENCODE: The reference human genome annotation for The ENCODE Project. *Genome Research*, 22(9):1760-1774, September 2012.
- [62] A. Hess and H. Tyer. Fisher’s combined p -value for detecting differentially expressed genes using Affymetrix expression arrays. *BMC Genomics*, 8(1):96, April 2007.
- [63] L. W. Hillier, G. T. Marth, A. R. Quinlan, D. Dooling, G. Fewell, D. Barnett, P. Fox, J. I. Glasscock, M. Hickenbotham, W. Huang, V. J. Magrini, R. J. Richt, S. N. Sander, D. A. Stewart, M. Stromberg, E. F. Tsung, T. Wylie, T. Schedl, R. K. Wilson, and E. R Mardis. Whole-genome sequencing and variant discovery in *C. elegans*. *Nature Methods*, 5(2):183-188, February 2008.
- [64] N. Homer, B. Merriman, and S. F. Nelson. BFAST: An Alignment Tool for Large Scale Genome Resequencing. *PLoS ONE*, 4(11):e7767, November 2009.
- [65] C. Howald, A. Tanzer, J. Chrast, F. Kokocinski, T. Derrien, N. Walters, J. M. Gonzalez, A. Frankish, B. L. Aken, T. Hourlier, J. H. Vogel, S. White, S. Searle,

- J. Harrow, T. J. Hubbard, R. Guigó, and A. Reymond. Combining RT-PCR-seq and RNA-seq to catalog all genic elements encoded in the human genome. *Genome Research*, 22(9):1698-1710, September 2012.
- [66] Y. Hu, Y. Huang, Y. Du, C. F. Orellana, D. Singh, A. R. Johnson, A. Monroy, P. F. Kuan, S. M. Hammond, L. Makowski, S. H. Randell, D. Y. Chiang, D. N. Hayes, C. Jones, Y. Liu, J. F. Prins, and J. Liu. DiffSplice: the genome-wide detection of differential splicing events with RNA-seq. *Nucleic Acids Research*, 41(2):e39, January 2013.
- [67] X. Huang and A. Madan. CAP3: A DNA sequence assembly program. *Genome Research*, 9(9):868-877, September 1999.
- [68] Human Genome Project. Available at: http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml.
- [69] L. hunter. Molecular Biology for computer Scientists. Artificial Intelligence and Molecular Biology. *MIT Press*, Cambridge, MA, 1-46. Available at: <http://compbio.ucdenver.edu/Hunter/01-Hunter.pdf>, 1993.
- [70] Illumina, Inc. RNA-Seq Data Comparison with Gene Expression Microarrays. *White paper*, Pub. No 470-2011-004, April 2011.
- [71] W. R. Jeck, J. A. Reinhardt, D. A. Baltrus, M. T. Hickenbotham, V. Magrini, E. R. Mardis, J. L. Dangl, and C. D. Jones. Extending assembly of short DNA sequences to handle error. *Bioinformatics*, 23(21):2942-2944, November 2007.
- [72] H. Jiang. Computational and statistical approaches in RNA sequencing analysis. PhD dissertation, *Institute for Computational and Mathematical Engineering*, Stanford university, 2009.

- [73] H. Kae. Genome projects: Uncovering the blueprints of biology. *In The Science Creative Quarterly*, August 2003.
- [74] Y. Katz, E. T. Wang, E. M. Airoidi, and C. B. Burge. Analysis and design of RNA sequencing experiments for identifying isoform regulation. *Nature Methods*, 7(12):1009-1015, November 2010.
- [75] W. J. Kent. BLAT-The BLAST-Like Alignment Tool. *Genome Research*, 12(4):656-664, April 2002.
- [76] H. Keren, G. Lev-Maor, and G. Ast. Alternative splicing and evolution: diversification, exon definition, and function. *Nature Reviews Genetics*, 11(5):345-355, May 2010.
- [77] J. T. Kost and M. P. McDermott. Combining dependent p-values. *Statistics & Probability Letters*, 60(2):183-190, November 2002.
- [78] V. M. Kvam, P. Liu, and Y. Si. A comparison of statistical methods for detecting differentially expressed genes from RNA-seq data. *American Journal of Botany*, 99(2):248-256, February 2012.
- [79] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25, March 2009.
- [80] B. Langmead, K. D. Hansen, and J. T. Leek. Cloud-scale RNA-sequencing differential expression analysis with Myrna. *Genome Biology*, 11(8):R83, August 2010.
- [81] Lawrence Livermore National Laboratory . Microarray technology. Available at: <https://www.llnl.gov/str/JulAug03/Wyrobek.html>.

- [82] N. Leng, J. A. Dawson, J. A. Thomson, V. Ruotti, A. I. Rissman, B. M. G. Smits, J. D. Haag, M. N. Gould, R. M. Stewart, and C. Kendzierski. EBSeq: An empirical Bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics*, 29(8):1035-1043, April 2013.
- [83] B. J. Li and G. C. Tseng. An adaptively weighted statistic for detecting differential gene expression when combining multiple transcriptomic studies. *The Annals of Applied Statistics*, 5(2A):994-1019, July 2011.
- [84] R. Li, Y. Li, K. Kristiansen, and J. Wang. SOAP: short oligonucleotide alignment program. *Bioinformatics*, 24(5):713-714, March 2008.
- [85] H. Li, J. Ruan, and R. Durbin. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research*, 18(11):1851-1858, November 2008.
- [86] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754-1760, July 2009.
- [87] R. Li, C. Yu, Y. Li, T. Lam, S. Yiu, K. Kristiansen, and J. Wang. SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*, 25(15):1966-1967, August 2009.
- [88] R. Li, H. Zhu, J. Ruan, W. Qian, X. Fang, Z. Shi, Y. Li, S. Li, G. Shan, K. Kristiansen, S. Li, H. Yang, J. Wang, and J. Wang. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*, 20(2):265-272, February 2010.
- [89] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078-2079, August 2009.

- [90] B. Li, and C. N. Dewey. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, 12:323, August 2011.
- [91] J. Li, D. M. Witten, I. M. Johnstone, R. Tibshirani. Normalization, testing, and false discovery rate estimation for RNA-sequencing data. *Biostatistics*, 13(3):523-538, July 2012.
- [92] J. Li and R. Tibshirani. Finding consistent patterns: a nonparametric approach for identifying differential expression in RNA-Seq data. *Statistical Methods in Medical Research*, 22(5):519-536, October 2011.
- [93] H. Lin, Z. Zhang, M. Q. Zhang, B. Ma, and M. Li. ZOOM! Zillions Of Oligos Mapped. *Bioinformatics*, 24(21):2431-2437, November 2008.
- [94] D. J. Lipman and W. R. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227(4693):1435-1441, March 1985.
- [95] T. Liptak. On the combination of independent tests. *Magyar Tud Akad Mat Kutato Int Kozl*, 3:171-197, 1958.
- [96] L. Liu, Y. Li, S. Li, N. Hu, Y. He, R. Pong, D. Lin, L. Lu, and M. Law. Comparison of Next-Generation Sequencing systems. *Journal of Biomedicine and Biotechnology*, 2012:251364, April 2012.
- [97] Q. Liu. RNA-Seq data analysis. *Department of Biomedical Informatics, Vanderbilt University School of Medicine*. Available at: <http://fenchurch.mc.vanderbilt.edu/bmif310/2012/RNA-Seq%20data%20analysis.pdf>
- [98] S. P. Lund, D. Nettleton, D. J. McCarthy, and G. K. Smyth. Detecting differential expression in RNA-sequence data using quasi-likelihood with shrunken

- dispersion estimates. *Statistical Applications in Genetics and Molecular Biology*, 11(5):8, October 2012.
- [99] E. R. Mardis. Next-Generation DNA Sequencing Methods. *Annual Review of Genomics and Human Genetics*, 9:387-402, June 2008.
- [100] 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 Research*, 18(9):1509-1517, June 2008.
- [101] J. A. Martin and Z. Wang. Next-generation transcriptome assembly. *Nature Reviews Genetics*, 12(10):671-682, September 2011.
- [102] J. Martin, V. M. Bruno, Z. Fang, X. Meng, M. Blow, T. Zhang, G. Sherlock, M. Snyder, and Z. Wang. Rnnotator: an automated de novo transcriptome assembly pipeline from stranded RNA-Seq reads. *BMC Genomics*, 11:663, November 2010.
- [103] C.K. Mathews, K.E. Van Holde, and K.F. van Holde. Biochemistry. *Benjamin-Cummings Pub Co.*, Second Edition, 1996.
- [104] T. R. Mercer, D. J. Gerhardt, M. E. Dinger, J. Crawford, C. Trapnell, J. A. Jeddloh, J. S. Mattick, and J. L. Rinn. Targeted RNA sequencing reveals the deep complexity of the human transcriptome. *Nature Biotechnology*, 30(1):99-104, January 2012.
- [105] M. L. Metzker. Sequencing technologies - the next generation. *Nature Reviews Genetics* 11(1):31-46, January 2010.
- [106] J. R. Miller, S. Koren, and G. Sutton. Assembly Algorithms for Next-Generation Sequencing Data. *Genomics*, 95(6):315-327, June 2010.

- [107] J. R. Miller, S. Koren, and G. Sutton. Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6):315-327, June 2010.
- [108] O. Morozova, M. Hirst, and M. A. Marra. Applications of New Sequencing Technologies for Transcriptome analysis. *Annual Review of Genomics and Human Genetics*, 10:135-151, September 2009.
- [109] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 5(7):621-628, May 2008.
- [110] F. Mosteller and R. R. Bush. Selected quantitative techniques. *In: Handbook of Social Psychology*, Addison-Wesley, Cambridge, Mass, 1:289-334, 1954.
- [111] G. S. Mudholkar and E. O. George. The logit method for combining probabilities. *In Symposium on Optimizing Methods in Statistics*, 345-366, Academic Press New York, 1979.
- [112] E. W. Myers, G. G. Sutton, A. L. Delcher, I. M. Dew, D. P. Fasulo, M. J. Flanigan, S. A. Kravitz, C. M. Mobarry, K. H. J. Reinert, K. A. Remington, E. L. Anson, R. A. Bolanos, H. Chou, C. M. Jordan, A. L. Halpern, S. Lonardi, E. M. Beasley, R. C. Brandon, L. Chen, P. J. Dunn, Z. Lai, Y. Liang, D. R. Nusskern, M. Zhan, Q. Zhang, X. Zheng, G. M. Rubin, M. D. Adams, and J. C. Venter. A Whole-Genome Assembly of *Drosophila*. *Science*, 24, 287(5461):2196-2204, March 2000.
- [113] S. Nacu, W. Yuan, Z. Kan, D. Bhatt, C. S. Rivers, J. Stinson, B. A. Peters, Z. Modrusan, K. Jung, S. Seshagiri, and T. D. Wu. Deep RNA sequencing analysis of readthrough gene fusions in human prostate adenocarcinoma and reference samples. *BMC Medical Genomics*, 4:11, January 2011.

- [114] M. Nicolae, S. Mangul, I. I. Măndoiu, and A. Zelikovsky. Estimation of alternative splicing isoform frequencies from RNA-Seq data. *Algorithms for Molecular Biology*, 6(1):9, April 2011.
- [115] Northern Blot Scheme. Available at: http://en.wikipedia.org/wiki/File:Northern_Blot_Scheme.PNG.
- [116] A. Oshlack, M. D. Robinson, and M. D Young. From RNA-seq reads to differential expression results. *Genome Biology*, 11(12):220, December 2010.
- [117] F. Ozsolak and P. M. Milos. RNA-Sequencing: advances, challenges and opportunities. *Nature Reviews Genetics*, 12(2):87-98, February 2011.
- [118] L. Pachter. Models for transcript quantification from RNA-Seq. arXiv:1104.3889v2 (<http://arxiv.org/abs/1104.3889>), May 2011.
- [119] W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8): 2444-2448, April 1988.
- [120] P. A. Pevzner, H. Tang, and M. S. Waterman. An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17):9748-9753, August 2001.
- [121] J. K. Pickrell, J. C. Marioni, A. A. Pai, J. F. Degner, B. E. Engelhardt, E. Nkadori, J. Veyrieras, M. Stephens, Y. Gilad, and J. K. Pritchard. Understanding mechanisms underlying human gene expression variation with RNA sequencing. *Nature*, 464(7289):768-772, April 2010.
- [122] M. Pop. Genome assembly reborn: recent computational challenges. *Brief Bioinform*, 10(4):354-366, July 2009.

- [123] L. Pray. Discovery of DNA structure and function: Watson and Crick. *Nature Education*, 1(1):100, 2008.
- [124] W. K. Purves, G. H. Orians, and H. C. Heller. Life: The Science of Biology, 4th edition. *Sinauer Associates, Inc.*, Sunderland, MA, December 1994.
- [125] A. R. Quinlan and I. M. Hall. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841-842, March 2010.
- [126] F. Rapaport, R. Khanin, Y. Liang, M. Pirun, A. Krek, P. Zumbo, C. E. Mason, N. D. Socci, and D. Betel. Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome Biology*, 14(9):R95, September 2013.
- [127] J. Reinartz, E. Bruyns, J. Lin, T. Burcham, S. Brenner, B. Bowen, M. Kramer, and R. Woychik. Massively parallel signature sequencing (MPSS) as a tool for in-depth quantitative gene expression profiling in all organisms. *Briefings in Functional Genomics and Proteomics*, 1(1):95-104, February 2002.
- [128] RNA Splicing. *Molecular Biology Web Book*. Available at: <http://www.web-books.com/MoBio/Free/Ch5A4.htm>.
- [129] RNA-Seq workflow. Available at: <http://cmb.molgen.mpg.de/2ndGenerationSequencing/Solas/RNA-seq.html>.
- [130] G. Robertson, J. Schein, R. Chiu, R. Corbett, M. Field, S. D. Jackman, K. Mungall, S. Lee, H. M. Okada, J. Q. Qian, M. Griffith, A. Raymond, N. Thiessen, T. Cezard, Y. S. Butterfield, R. Newsome, S. K Chan, R. She, R. Varhol, B. Kamoh, A. Prabhu, A. Tam, Y. Zhao, R. A. Moore, M. Hirst, M. A. Marra, S. J. M. Jones, P. A. Hoodless, and I. Birol. De novo assembly and analysis of RNA-seq data. *Nature Methods*, 7(11):909-912, October 2010.

- [131] M. D. Robinson and G. K. Smyth. Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, 23(21):2881-2887, September 2007.
- [132] M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139-140, January 2010.
- [133] M. D. Robinson and A. Oshlack. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(3):R25, March 2010.
- [134] T. Rodden. Genetics FOR DUMMIES. *Wiley Publishing, Inc.*, First edition, 2005.
- [135] S. M. Rumble, P. Lacroute, A. V. Dalca1, M. Fiume, A. Sidow, and M. Brudno. SHRiMP: Accurate Mapping of Short Color-space Reads. *PLoS Comput Biol*, 5(5):e1000386, May 2009.
- [136] SAM Format Specification. Available at: <http://samtools.github.io/hts-specs/SAMv1.pdf>.
- [137] M. H. Schulz, D. R. Zerbino, M. Vingron, and E. Birney. Oases: Robust de novo RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics*, 28(8):1086-1092, April 2012.
- [138] Serial Analysis Of Gene Expression. Available at: <http://www.sagenet.org/findings/index.html>.
- [139] R. Shamir and R. Sharan. Algorithms in Molecular Biology. Course Lecture Notes. *Blavatnik School of Computer Science, Tel Aviv University*, January 2011. Available at: http://www.cs.tau.ac.il/~rshamir/algmb/archive/ngs_algorithms.pdf.

- [140] S. Shen, J. W. Park, J. Huang, K. A. Dittmar, Z. X. Lu, Q. Zhou, R. P. Carstens, and Y. Xing. MATS: a Bayesian framework for flexible detection of differential alternative splicing from RNA-Seq data. *Nucleic Acids Research*, 40(8):e61, April 2012.
- [141] J. Shendure and H. Ji. Next-Generation DNA sequencing. *Nature Biotechnology*, 26(10), 1135-1145, October 2008.
- [142] L. Shi, L. H. Reid, W. D. Jones, R. Shippy, J. A. Warrington, S. C. Baker, P. J. Collines, F. de Longueville, E. S. Kawasaki, K. Y. Lee, et al. The MicroArray Quality Control (MAQC) project shows inter-and intraplatform reproducibility of gene expression measurements. *Nature Biotechnology*, 24(9):1151-1161, September 2006.
- [143] J. T. Simpson, K. Wong, S. D. Jackman, J. E. Schein, S. J. M. Jones, and . Birol. ABySS: a parallel assembler for short read sequence data. *Genome Research*, 19(6):1117-1123, June 2009.
- [144] G. K. Smyth. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):1-25, February 2004.
- [145] SRA database growth. Available at: <http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=announcement>.
- [146] S. Srivastava and L. Chen. A two-parameter generalized Poisson model to improve the analysis of RNA-seq data. *Nucleic Acids Research*, 38(17):e170, September, 2010.
- [147] O. Stegle, P. Drewe, R. Bohnert, K. M. Borgwardt, G. Rätsch. Statistical Tests

- for Detecting Differential RNA-Transcript Expression from Read Counts. *Nature Precedings*, 713, May 2010.
- [148] S. A. Stouffer, E. A. Suchman, L. C. DeVinney, S. A. Star, and R. M. Williams. The American Soldier: Adjustment During Army Life. *Princeton University Press, Princeton*, Volume 1, 1949.
- [149] K. Sun, X. Chen, P. Jiang, X. Song, H. Wang, and H. Sun. iSeeRNA: identification of long intergenic non-coding RNA transcripts from transcriptome sequencing data. *BMC Genomics*, 14(Suppl 2):S7, February 2013.
- [150] Y. Surget-Groba and J. I. Montoya-Burgos. Optimization of de novo transcriptome assembly from next-generation sequencing data. *Genome Research*, 20(10):1432-1440, October 2010.
- [151] S. Tarazona, F. Garca, A. Ferrer, J. Dopazo, and A. Conesa. NOIseq: a RNA-seq differential expression method robust for sequencing depth biases. *EMBnet.journal*, 17(B):pp-18, 2012.
- [152] The Central Dogma of Molecular Biology. *Access Excellence @ the National Health Museum*. Available at: <http://www.accessexcellence.org/RC/VL/GG/central.php>.
- [153] L. H. C. Tippett. The methods of statistics. *Williams & Norgate*, London, 1931.
- [154] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5): 511-515, May 2010.
- [155] C. Trapnell and S. L. Salzberg. How to map billions of short reads onto genomes. *Nature Biotechnology*, 27(5):455-457, May 2009.

- [156] Typical animal, bacteria, and plant cells. Available at: <http://micro.magnet.fsu.edu/cells/index.html>.
- [157] M. A. Van De Wiel, G. G. 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*, 14(1):113-128, January 2013.
- [158] L. Wan and F. Sun. CEDER: accurate detection of differentially expressed genes by combining significance of exons using RNA-Seq. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(5):1281-1292, September-October 2012.
- [159] Z. Wang, M. Gerstein, and M. Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1), 57-63, January 2009.
- [160] 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*, 26(1):136-138, October 2010.
- [161] R. L. Warren, G. G. Sutton, S. J. M. Jones, and R. A. Holt. Assembling millions of short DNA sequences using SSAKE. *Bioinformatics*, 23 (4):500-501, February 2007.
- [162] M. C. Whitlock. Combining probability from independent tests: the weighted z-method is superior to fisher’s approach. *Journal of Evolutionary Biology*, 18(5):1368-1373, September 2005.
- [163] Z. Wu, B. D. Jenkins, T. A. Rynearson, S. T. Dyhrman, M. A. Saito, M. Mercier, and L. P. Whitney. Empirical bayes analysis of sequencing-based transcriptional profiling without replicates. *BMC Bioinformatics*, 11(1):564, November 2010.

- [164] J. H. Yang, J. H. Li, S. Jiang, H. Zhou, and L. H. Qu. ChIPBase: a database for decoding the transcriptional regulation of long non-coding RNA and microRNA genes from ChIP-Seq data. *Nucleic Acids Research*, 41(D1):D177-D187, November 2013.
- [165] J. H. Yang and L. H. Qu. deepBase: Annotation and discovery of microRNAs and other noncoding RNAs from deep-sequencing data. *Methods in Molecular Biology*, 822:233-248, 2012.
- [166] C. Zang, D. E. Schones, C. Zeng, K. Cui, K. Zhao, and W. Peng. A clustering approach for identification of enriched domains from histone modification ChIP-Seq data. *Bioinformatics*, 25(15):1952-1958, August 2009.
- [167] D. R. Zerbino and E. Birney. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, 18(5):821-829, March 2008.
- [168] J. Zhang, R. Chiodini, A. Badr, and G. Zhangd. The impact of next-generation sequencing on genomics. *J Genet Genomics*, 38(3):95-109, March 2011.
- [169] X. Zhou, L. Ren, Q. Meng, Y. Li, Y. Yu, and J. Yu. The next-generation sequencing technology and application. *Protein Cell*, 1(6):520-536, June 2010.
- [170] Y. H. Zhou, K. Xia, and F. A. Wright. A powerful and flexible approach to the analysis of RNA sequence count data. *Bioinformatics*, 27(19):2672-2678, October 2011.

APPENDIX A

LIST OF ABBREVIATIONS

ABySS	Assembly By Short Sequences
auROC	Area under Receiver Operating Characteristic
BFAST	Blat-like Fast Accurate Search Tool
BLAST	Basic Local Alignment Search Tool
BLAT	the BLAST-Like Alignment Tool
DAG	Directed Acyclic Graph
DDBJ	DNA Data Bank of Japan
DE	Differential Expression
ELAND	Efficient Large-Scale Alignment of Nucleotide Databases
EMBL	European Molecular Biology Laboratory
emPCR	Emulsion Polymerase Chain Reaction
ERANGE	Enhanced Read Analysis of Gene Expression
GTF	Gene Transfer Format
LRT	Likelihood Ratio Test
MAQ	Mapping and Assembly with Quality
MISO	Mixture of Isoform
mRNA	Messenger RNA
NGS	Next Generation Sequencing
PCR	Polymerase Chain Reaction
PGM	Personal Genome Machine
ROC	Receiver Operating Characteristic

rRNA	Ribosomal RNA
RSEM	RNA-Seq by Expectation-Maximization
SAM	Sequence Alignment/Map
SHARCGS	SHort read Assembler Based on Robust Contig Extension for Genome Sequencing
SHRiMP	SHort Read Mapping Package
SMRT	Single Molecule Teal Time
SNP	Single Nucleotide Polymorphism
SOAP	Short Oligonucleotide Alignment Program
SRA	Sequence Read Archive
SSAKE	The Short Sequence Assembly by K-mer Search and 3' Read Extension
VCAKE	Verified Consensus Assembly by K-mer Extension
ZOOM	Zillions Of Oligos Mapped

CURRICULUM VITAE

Abdallah Eteleeb

Department of Computer Engineering and Computer Science

University of Louisville

Louisville, KY, 40292

Email: abdallah.eteleeb@louisville.edu

Research Interests

Developing computational algorithms and approaches to analyze high-throughput next-generation sequencing data for different applications such as RNA-Seq, ChIP-Seq analysis, and variant detection, Data Mining, and Machine learning.

Education

- Ph.D., Computer Engineering and Computer Science (Expected: May 2015)
University of Louisville, Louisville, KY
Advisors: Dr. Eric C. Rouchka
Dissertation Topic: An Island-Based Approach for Differential Expression Analysis and Alternative Splice Detection.
- M.S., Information Systems Development, 2002-2005
HAN University of Applied Sciences, Arnhem, The Netherlands
Thesis Title: Aspects of Aggregates in Data Warehouses and Multidimensional Databases.
- B.S., Computer Science, 1992-1996
University of Aljabal Algharbi, Zentan, Libya

Positions and Employment

- Instructor, Summer 2013, 2014
Computer Engineering and Computer Science
University of Louisville
CECS 660-50: Introduction to Bioinformatics, online graduate course.

- Graduate Student Assistant (GSA), 2013-2014
Resources for Academic Achievement (REACH)
Undergraduate Affairs, University of Louisville.
- Assistant Lecturer, 2005-2008
Computer Science Department
University of Aljabal Algharbi, Zentan, Libya.
- Adjunct Lecturer, 2007-2008
Software Development Department
Higher Institute of Vocational Training, Jado, Libya.
- Adjunct Lecturer, 2007-2008
Computer Applications Lecturer
Higher Institute of Public Health, Yafren, Libya.
- Teaching Assistant, 1998-2001
Computer Science Department
University of Aljabal Algharbi, Zentan, Libya.

Work Experience

- Bioinformatics, Graduate Research Assistant, 2010-present
Bioinformatics and Biomedical Computing Laboratory
Computer Engineering and Computer Science, University of Louisville
 - Developed an Island-Based (IBSeq) method for differential expression analysis.
 - Conducting various next-generation sequencing analysis for Kentucky Biomedical Research Infrastructure Network (KBRIN) projects.
 - Provide bioinformatics research support in KBRIN in different areas such as sequence alignment, assembly, and variant calling.
- Graduate Student Assistant (GSA), 2013-2014
Resources for Academic Achievement (REACH), University of Louisville.
 - Managed and Supervised Computer Resource Centers (CRC) with around 60 PCs.
 - Manager of Computer Resource Center - iTech Zone.
 - Tutoring undergraduate student in C, C++, C#, and Java classes.
 - Supervised and trained around 21 tutors in programming courses.
 - Supervised and led tutor training sessions.
 - Conducted test review sessions for programming languages, C, C++, C#, and Java.
 - Provided technical assistance (Microsoft office, Windows, Blackboard, ULink, e-mail, Wireless Access, Password Settings) to the University students and staff.
 - REACH Webmaster. Maintained and updated the website for REACH.

Skills

- Bioinformatics Skills
 - RNA-Seq: Cufflinks/Cuffdiff, DESeq, DEXSeq, edgeR, BaySeq, MATS, and MISO.
 - ChIP-Seq: MACS, PeakAnalyzer, and SICER.
 - Sequence Mapping and Assembly: Bowtie, BWA, Maq, and Tophat.
 - Variant Calling: mpileup, Bcftools, snpEff, and SnpSift.
 - NGS data pre-processing and quality control: Samtools, FASTX-Toolkit, FastQC, BED-Tools, BamTools, and HTSeq.
 - Integrated Genome Analysis Platforms: Galaxy, GenePattern.
 - Visualization Tools: Integrative Genomics Viewer (IGV), GBrowse.
- Computer Skills
 - Programming & Scripting: C/C++, Java, C#, Perl, Python, R, JavaScript, UNIX Shell, Sed, Awk.
 - Web Programming: ASP, PHP and C#.
 - Web GUIs languages: HTML (XHTML), CSS, and JavaScript.
 - Database languages: SQL.
 - Database Management Systems: Microsoft SQL Server, Oracle, and MySQL.
 - Algorithms Design and Analysis and Data Structures.
 - Operating Systems: Linux (ubuntu, Linux Mint), Mac, and Windows.
- Information Technology
 - M.Sc in Information Systems Development.
 - Capable of doing complex databases and application development.
 - Well-known knowledge of Relational Database Management Systems and Application Servers.
 - Capable of developing Data Warehousing and OLAP Applications.

Working Papers

- **Eteleeb AM**, Flight RM, Harrison BJ, Petruska JC, Moseley HNB, Rouchka EC. “IBSeq: An Island-Based RNASeq Approach for Annotation-Free Detection of Differential Expression”. *In preparation for Nucleic Acids Research.*

Refereed Publications

- Elena Matveeva, John Maiorano, Qingyang Zhang, **Abdallah M Eteleeb**, Paolo Convertini, Stefan Stamm, Eric C Rouchka, Jiping Wang and Yvonne Fondufe-Mittendorf. “PARP-1 regulates alternative splicing.” *Nucleic Acids Research*, (Under revision).
- **Eteleeb, Abdallah M.**, Hunter N. Moseley, and Eric C. Rouchka. “A comparison of combined p-value methods for gene differential expression using RNA-seq data.” *In Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 417-425. ACM, 2014.
- **Eteleeb, Abdallah M.**, Robert M. Flight, Benjamin J. Harrison, Jeffrey C. Petruska, and Eric C. Rouchka. “An island-based approach for differential expression analysis.” *In Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, p. 419. ACM, 2013.
- **Eteleeb AM**, Rouchka EC. Differential expression analysis methods for RNA-Seq data. *OA Bioinformatics* 2013 Sep 01;1(1):3.
- Flight RM, **Eteleeb AM**, Rouchka EC. Affymetrix Mismatch(MM) Probes: Useful After All. *Proceedings of the 2012 ASE/IEEE International Conference on BioMedical Computing (BioMedCom 2012)*, pp. 561-568. December 14-16, 2012, Washington, D.C. (DOI 10.1109/SocialInformatics.2012.36)

Meeting Abstracts

- Harrison BJ, Flight RM, **Eteleeb A**, Rouchka EC, Petruska JC. ”RNA Seq profiling of UTR expression during neuronal plasticity“. *Proceedings of the Eleventh Annual UT-ORNL-KBRIN Bioinformatics Summit*. BMC Bioinformatics, 13(Suppl 12):A4, March 30-April 1, 2012, Louisville, KY.
- Rajadinakaran, G., Huifang, F., **Eteleeb, A.**, Rouchka E., Smith, M.E. Next Generation Sequencing identifies Regulation of Pathways in Zebrafish Auditory Hair Cell Regeneration. *Kentucky Academy of Sciences, the 97th Annual Meeting*, November 4-5, 2011, Murray, KY.

Presentations and Talks

- An Island-Based Approach for Differential Expression Analysis. UT-KBRIN Bioinformatics Summit 2014, April 11-13, 2014, Cadiz, KY. (oral, poster)
- An Island-Based Approach for Differential Expression Analysis. ACM Conference on Bioinformatics, Computational Biology and Biomedical Informatics (ACM-BCB 2013), September 22-25, 2013, Bethesda, MD. (oral, poster)
- A Region-Based Approach for Differential Expression Analysis and Alternative Splicing Detection. 99th Annual Meeting of the Kentucky Academy of Sciences, November 9, 2013, Morehead, KY. (oral)
- Accessing and Analyzing your data in Galaxy. KBRIN/Cofactor Genomics Next-Generation Sequencing Workshop, August 15, 2011, Louisville, KY. (oral)
- Introduction to Data Warehousing and OLAP. 1th International Symposium on Information Systems Modeling and Development, June 2006, Tripoli, Libya. (oral)