University of Kentucky
UKnowledge

Theses and Dissertations--Computer Science | Computer Science

2018

NOVEL COMPUTATIONAL METHODS FOR SEQUENCING DATA ANALYSIS: MAPPING, QUERY, AND CLASSIFICATION

Xinan Liu
University of Kentucky, hitcslxa@gmail.com
Digital Object Identifier: https://doi.org/10.13023/ETD.2018.121

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Recommended Citation

Liu, Xinan, "NOVEL COMPUTATIONAL METHODS FOR SEQUENCING DATA ANALYSIS: MAPPING, QUERY, AND CLASSIFICATION" (2018). *Theses and Dissertations--Computer Science*. 63.
https://uknowledge.uky.edu/cs_etds/63

This Doctoral Dissertation is brought to you for free and open access by the Computer Science at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Computer Science by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

المنارة للاستشارات
www.manaraa.com

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Xinan Liu, Student

Dr. Jinze Liu, Major Professor

Dr. Miroslaw Truszczynski, Director of Graduate Studies

المنارة للاستشارات
www.manaraa.com

NOVEL COMPUTATIONAL METHODS FOR SEQUENCING DATA ANALYSIS: MAPPING, QUERY, AND CLASSIFICATION

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for the
degree of Doctor of Philosophy in the
College of Engineering at the
University of Kentucky

By
Xinan Liu
Lexington, Kentucky

Director: Dr. Jinze Liu, Associate Professor of Computer Science
Lexington, Kentucky 2018

Copyright© Xinan Liu 2018

للاستشارات المنارة
www.manaraa.com

ABSTRACT OF DISSERTATION

NOVEL COMPUTATIONAL METHODS FOR SEQUENCING DATA ANALYSIS: MAPPING, QUERY, AND CLASSIFICATION

Over the past decade, the evolution of next-generation sequencing technology has considerably advanced the genomics research. As a consequence, fast and accurate computational methods are needed for analyzing the large data in different applications. The research presented in this dissertation focuses on three areas: RNA-seq read mapping, large-scale data query, and metagenomics sequence classification.

A critical step of RNA-seq data analysis is to map the RNA-seq reads onto a reference genome. This dissertation presents a novel splice alignment tool, MapSplice3. It achieves high read alignment and base mapping yields and is able to detect splice junctions, gene fusions, and circular RNAs comprehensively at the same time. Based on MapSplice3, we further extend a novel lightweight approach called iMapSplice that enables personalized mRNA transcriptional profiling. As huge amount of RNA-seq has been shared through public datasets, it provides invaluable resources for researchers to test hypotheses by reusing existing datasets. To meet the needs of efficiently querying large-scale sequencing data, a novel method, called SeqOthello, has been developed. It is able to efficiently query sequence *k*-mers against large-scale datasets and finally determines the existence of the given sequence. Metagenomics studies often generate tens of millions of reads to capture the presence of microbial organisms. Thus efficient and accurate algorithms are in high demand. In this dissertation, we introduce MetaOthello, a probabilistic hashing classifier for metagenomic sequences. It supports efficient query of a taxon using its k-mer signatures.

KEYWORDS: RNA-seq, mapping, splice junction, Othello, query, classification

Author's signature: Xinan Liu

Date: May 1, 2018

المنارة للاستشارات

www.manaraa.com

NOVEL COMPUTATIONAL METHODS FOR SEQUENCING DATA ANALYSIS: MAPPING, QUERY, AND CLASSIFICATION

By
Xinan Liu

Director of Dissertation: Jinze Liu
Director of Graduate Studies: Miroslaw Truszczynski
Date: May 1, 2018

www.manaraa.com

ACKNOWLEDGMENTS

First, I would like to express my deepest gratitude to my advisor, Dr. Jinze Liu, for her continuous support and encouragement throughout my Ph.D. study. She is not only a great researcher with immense knowledge, but also an excellent advisor that drives students to the right direction. Without her patient guidance, I wouldn't have been able to finish this dissertation.

I would also like to thank Dr. James N. MacLeod. He not only taught me a lot of knowledge in biology but also helped me learn to work with researchers of other backgrounds. I am also grateful to Dr. Jerzy W. Jaromczyk, and Dr. Zongming Fei, for their invaluable comments and constructive suggestions on my dissertation.

I am fortunate to have many great friends in Lexington. Thanks for making my life enjoyable. Last but not the least, I thank my parents for their unconditional love and support all these years.

www.manaraa.com

TABLE OF CONTENTS

Acknowledgments iii

Table of Contents iv

List of Figures vi

List of Tables xi

Chapter 1 Introduction 1
  1.1 Biological background 1
  1.2 Next generation sequencing and RNA-seq 3
  1.3 Current computational methods for Next Generation Sequencing data analysis 4
    1.3.1 The typical RNA-seq data analysis pipeline 5
    1.3.2 RNA-seq read alignment and splice junction detection 7
    1.3.3 Gene fusion and circular RNA detection using RNA-seq 12
    1.3.4 Large scale RNA-seq data query 14
    1.3.5 Taxonomic classification of metagenomics sequencing data 16
  1.4 Dissertation Statement 18
  1.5 Contributions of this dissertation 18

Chapter 2 Comprehensive RNA-seq read alignment with MapSplice3 22
  2.1 Introduction 22
  2.2 Overview of MapSplice3 algorithm 26
  2.3 Global alignment with semi-maximal mappable prefix search 27
  2.4 Context-aware local alignment 30
    2.4.1 Alignment to splicing context 31
    2.4.2 Alignment to mutation context 34
    2.4.3 Pooled context from multiple datasets. 35
  2.5 Experimental results 36
    2.5.1 Performance comparison on regular read alignment 36
    2.5.2 Fusion transcript prediction results 53
    2.5.3 Circular RNA detection results 56
    2.5.4 Running time and memory requirements 56
  2.6 Conclusion 58

Chapter 3 Individualized RNA-seq read alignment 63
  3.1 Introduction 63
  3.2 iMapSplice algorithm 66
    3.2.1 SNP-mer generation 69
  3.3 Experimental results 71
    3.3.1 Improvement in general read mapping 72

www.manaraa.com

3.3.2 Reduction of biases in allelic ratio 75
3.3.3 Discovery of personal splice junctions 78
3.3.4 Super fast and space-saving indexing strategy 81
3.4 Conclusion 82

Chapter 4 Scalable query over large scale sequencing datasets 85
4.1 Introduction 85
4.2 Sequencing data *k*-mer content 86
4.2.1 Compressed *k*-mer 87
4.2.2 Mapping between *k*-mer to occurrence map within a frequency bucket 88
4.2.3 Sequence query on SeqOthello 90
4.3 Transcript query over 148 equine RNA-seq datasets 91
4.4 Gene fusion transcript query against TCGA Pan-Cancer RNA-seq datasets . 101
4.5 Conclusion 106

Chapter 5 Efficient taxonomic classification of metagenomics read 107
5.1 Introduction 107
5.2 MetaOthello algorithm 108
5.2.1 *K*-mer taxon signatures 108
5.2.2 Taxonomic classification of sequencing reads 109
5.3 Experimental results on simulated datasets 112
5.3.1 Classification accuracy and taxon-specific *k*-mer relative abundance 112
5.3.2 Comparison with the state-of-the-art tools 113
5.4 Metagenomic classification of real datasets 118
5.4.1 Human Microbiome Project data 118
5.4.2 Zika virus detection 118
5.5 Conclusion 120

Chapter 6 Conclusion 121

Bibliography 125

Vita 143

www.manaraa.com

LIST OF FIGURES

1.1 Central dogma of molecular biology. The gene shown above contains four exons sparated by three introns. In the first step of transcription, pre-mRNA is created. Second, the pre-mRNA is spliced into mRNA transcript, and this step is called splicing, in which all exons are retained and introns are removed. Third, the mRNA transcript is translated into protein. 2
1.2 Overview of a typical RNA-seq experiment. (1) RNA molecules are fragmented firstly; (2) $3'$ fragment end is utilized to make single stranded cDNA; (3) $5'$ fragment end is selected and double stranded cDNA is created during second-strand synthesis; (4) fragments are categorized based on their size and those with proper size will be used further for sequencing; (5) fragments are sequenced and single-end (sampled from one end) or paired-end reads (sampled from both ends) are generated. (Figure partially adpated from Wikipedia (www.wikipedia.com)) 4
1.3 Origins of reads reveal mRNA transcript structure and abundance. There are two types of reads according to their origins. The first type of reads are continuous ones and the other type of reads are spliced ones. The bases covered by read sequences are all exonic ones, and the splice sites could be obtained from spliced reads. The density of reads also reflects the abundance of mRNA transcript. The more reads fall into the corresponding genomic regions, the higher expression the mRNA transcript has. 6
1.4 Typical RNA-seq data analysis pipeline. Read alignment is the first and fundamental step of all the four common RNA-seq data analysis applications. 7
1.5 An illustration of alignment difficulty caused by the presence of pseudogene. . 8
1.6 Aligners timeline since 2001. DNA mappers are plotted in blue, mRNA aligners (spliced aligners) are plotted in red and miRNA mappers in green and bisulphite mappers in purple. Grey dotted lines connect related mappers (extensions or new versions). 9
1.7 Fusion transcripts are formed from two separate transcripts. In this figure, three exons from the pre-mRNA in Gene 1 and two exons from the pre-mRNA in Gene 2 formed the fusion transcript. (Figure partially adapted from Wikipedia (www.wikipedia.com)) 13
1.8 An example of circular RNA comprised with two exons from the regular transcript. 14
1.9 Origins of back-splicing reads on circular RNA and their alignment on reference genome. 14

2.1 An overview of MapSplcie3 algorithm. MapSplice3 algorithm contains two tiers: global alignment with semi-maximal mappable prefix search (1$\sim$4), and context-aware local alignment to splicing structures, genomic sequences with mutations, and reference genome (5 and 6). 28

www.manaraa.com

2.2 Examples to illustrate how splicing context and mutation context are derived. (A) An example of splicing context. Splice site sequence pairs represented with green bars are examined for junction consolidation. A splice junction will be excluded if two paired sequences are identical or similar to each other. (B) An example of mutation context. Two of the five mismatch positions are identified as candidate SNPs because they have high read coverage concordance. 32
2.3 (A) The regular mode of MapSplice3 that utilizes unshared context: datasets are processed separately, and context derived from each dataset will only be used during the context-aware local alignment of reads from the same dataset. (B) The alternative mode of MapSplice3 that utilizes shared context: context from multiple datasets are pooled, and can be utilized for reads from any datasets. 36
2.4 Read alignment yield in simulated datasets. Shown is the number of reads in ground truth and mapped by each aligner. Mapped reads are categorized according to the number of splice junctions detected. There is no splice junction in “Unspliced” reads. “SingleSpliced” and “MultiSplcied” reads contain one and more than one splice junctions respectively. 38
2.5 Read alignment yield in real datasets. Shown is the number of reads mapped by each aligner. Mapped reads are categorized according to the number of splice junctions detected. There is no splice junction in “Unspliced” reads. “SingleSplcied” and “MultiSplcied” reads contain one and more than one splice junctions respectively. 38
2.6 Distribution of base mapping rates along read sequence in simulated data. Total percentages of mapped bases for each base along the read sequence are counted and plotted for each aligner. 39
2.7 Distribution of base mapping rates along read sequence in real data. Total percentages of mapped bases for each base along the read sequence are counted and plotted for each aligner. 40
2.8 True positive rate (TPR) vs. false positive rate (FPR) (ROC-curve) for discovered splice junctions on simulated datasets. TPR is definded the same as sensitivity, which is the fraction of successfully detected true junctions among all true junctions. While FPR is defined as the fraction of falsely detected splice junctions among the total splice junctions. 41
2.9 The number of annotated splice junctions vs. the number of unannotated splice junctions detected by each aligner on the real datasets. 42
2.10 Correlation of splice junction supporting read numbers between aligners’ results and ground truth in all the six simulated data sets. Each point in the scatter plot represents supporting numbers for one splice junction in aligners’ results (X axis) and ground truth (Y axis). 43
2.11 Summary of splice junction discovery and read mapping rates for aligners with 10 datasets from Geuvadis RNA sequencing project. (A) (B) Number of annotated splice junctions detected by each aligner on the 10 datasets. (C) Number of unannotated splice junctions detected by each aligner on the 10 datasets. (D) Read mapping rates for each aligner on the 10 datasets. 48

www.manaraa.com

2.12 (A) Base mapping rates along the read sequence reported by each aligner in both untrimmed data and trimmed data. (B) Distribution of total mismatches in alignments. . . . 51
2.13 Number of supporting reads identified by different tools for each back-splice junction. Both MapSplice3 and CIRI discovered all the 24 experimentally validated back splice junctions, while STAR missed 3 of them (the 12th, 21th, and 24th in the figure). . . . 57
2.14 Million bases mapped per second by each aligner on (A) simulated datasets, R101, R262, and (B) ten datasets from Geuvadis RNA sequencing project. (C) Memory requirements of spliced alignment software. . . . 58

3.1 (A) An example illustrating the challenge when mapping a RNA-seq read to the reference genome in the presence of SNPs. (B) An example illustrating how iMapSplice algorithm may resolve spliced alignment with SNPs as well as the basic steps of the alignment. . . . 67
3.2 Aggregated reference allelic ratio distribution of iMapSplice and “reference only” methods (MapSplice, HISAT2, and STAR) on RNA-seq datasets from individuals NA12812, NA12749, NA07056, NA06994, and NA12275. . . . 76
3.3 Aggregated reference allelic ratio distribution of iMapSplice and “mask-based” methods (MapSplice MASK, HISAT2 MASK, and STAR MASK) on RNA-seq datasets from individuals NA12812, NA12749, NA07056, NA06994, and NA12275. . . . 76
3.4 Aggregated reference allelic ratio distribution of iMapSplice and “SNP-aware” methods (HISAT2 SNP and HISAT2 POP) on RNA-seq datasets from individuals NA12812, NA12749, NA07056, NA06994, and NA12275. . . . 77
3.5 Eamples of mutations and their corresponding impact on expression from the creation or loss of canonical splice junctions. Bases in black are the reference nucleotides, and those in red are alternate bases at SNP positions. The numbers in the figure show supporting read counts for the corresponding splice junctions. 81

4.1 Left: A toy example illustrating steps of sequence query over SeqOthello structure. Right: Two-layer hierarchical structure of SeqOthello (note that *k*-mer occurrence map is not shown). For any given query, SeqOthello first enumerates its *k*-mers and maps them to corresponding frequency buckets using the first layer Othello. Secondly, the associated frequency buckets in the second layer will be loaded into memory to extract the occurrence map for each *k*-mer falling in its bucket. In the last step, occurrence maps of all *k*-mers will be assembled to identify samples with the presence of *k*-mers more than a certain threshold θ. . . . 92
4.2 *k*-mer frequency distribution in 148 equine RNA-seq datasets used in the experiment section. . . . 92
4.3 Comparison of transcript sequence query results between SBT and SeqOthello implementation variants. Transcripts with TPM$\geq$1 were considered as present in a sample. Transcripts with TPM$<$1 were considered as absent in a sample. . . 96

المنارة للاستشارات
www.manaraa.com

4.4 Comparison of transcript sequence query results between SBT and SeqOthello implementation variants. Transcripts with TPM$\geq$100 were considered as present in a sample. Transcripts with TPM$<$1 were considered as absent in a sample. . 97
4.5 Accuracy in terms of ROC-curves for transcript sequence query results reported by SeqOthello-Exact as a function of expression levels. 98
4.6 Accuracy in terms of ROC-curves for transcript sequence query results reported by SeqOthello-Exact as a function of *mincov*, min number of reads supporting a *k*-mer. 99
4.7 Accuracy in terms of ROC-curves for transcript sequence query results reported by SeqOthello-Exact as a funciton of *k*-mer length. 100
4.8 An illustration of fusion junction sequence constructed for fusion query using SeqOthello. Each fusion junction sequence consists of 20 bases from donor exon in one gene and 20 bases from acceptor exon in the other gene. Each 21-mer within this 40-base sequence spans the fusion junction. The query of a fusion sequence using SeqOthello may return a maximum of 20 k-mer hits for each RNA-seq Experiments indexed by SeqOthello. 102
4.9 An illustration of fusion calling criteria using SeqOthello's fusion junction sequence query results against TCGA pan-cancer RNA-seq data. (a) and (b) are examples of *k*-mer hit distribution as a result of fusion junction sequence query using SeqOthello. The presence of a small set of *k*-mers in large fraction of samples indicates background noise as a result of these *k*-mers being repetitive. For each fusion, we use σ^{98th}, the *k*-mer hit at 98th percentile as an estimation of background noise. (a) Histogram of *k*-mer hits by querying junction sequence spanning chr21:42880008-chr21:39956869 connecting gene pair TMPRSS2-ERG. The background noise is estimated at σ^{98th}=2. (b) Histogram of *k*-mer hits querying junction sequence spanning chr5:134688636-chr5:179991489 connecting gene pair H2AFY-CNOT6. The background noise is estimated at σ^{98th}=6. (c) The comparison of performance in recovering database-known fusion occurrences and detecting novel occurrences between noise-aware approach and SBT-like approach using θ-based containment query. Here μ is minimum number of *k*-mer hits required beyond the fusion-specific noise level used in noise-aware approach; θ is the minimum fraction of *k*-mer hits required to call the presence of a query as used in SBT containment query. (d) The distribution of the actual *k*-mer hits of all called fusion occurrences called with noise-aware approach. 104
4.10 Top ten recurrent gene fusion detection results across 29 tumor types. Bar plots show occurrence number of top ten recurrent gene fusions detected by SeqOthello over different tumor types. Occurrences of each fusion on each tumor type are classified into novel occurrences (not in TCGA fusion database) and annotated occurrences (already curated by TCGA Gene Fusion Database). . 105

5.1 Illustration of MetaOthello algorithm. A: an example of taxonomy with reference sequences in the leaf nodes. 3-mers that are signatures to each node are highlighted in red colors with different shades. B: a two-step approach in read classification. 110

المنارة للاستشارات
www.manaraa.com

5.2 Correlation between species-specific *k*-mer signatures and classification accuracy when *k*=20 (A) and *k*=31 (B). In each panel, the central figure depicts the correlation between species-specific *k*-mer proportion and read-classification accuracy for all species; the top histogram shows the distribution of species as a function of species-specific *k*-mer proportion, and the right histogram shows the distribution of classification accuracy for all species. 113
5.3 Billion bases processed per minute by each tool with three *k*-mer length settings using 8 threads. 117

المنارة للاستشارات
www.manaraa.com

LIST OF TABLES

1.1 Genome sizes of different organisms. 1

2.1 Overview of simulated and real datasets. 37
2.2 Overview of metrics for aligner performance assessment. 37
2.3 Fusion detection results on synthetic datasets (5X, 20X, 50X). Symbol * marks the top 3 tools in terms of F_1 score. 54
2.4 Fusion detection results on synthetic datasets (100X, 200X). Symbol * marks the top 3 tools in terms of F_1 score. 54
2.5 Fusion detection results on real datasets. Symbol * marks the top three tools in terms of F_1 score. 55
2.6 Summary of splice junction discovery performance, read / base mapping rates, and read coverage correlation at SNPs in simulated datasets reported by MapSplice3 with three different configurations. 60
2.7 Summary of splice junction discovery performance, read / base mapping rates, and read coverage correlation at SNPs in simulated datasets reported by MapSplice3 with three different configurations. 61
2.8 Summary of splice junction discovery performance and read / base mapping rates in real datasets reported by MapSplice3 and five other aligners. 62

3.1 The set of tools and their index configuration used in performance comparison. 72
3.2 The comparison of five baseline tools in terms of the number of accurate unique alignments out of 20 million synthetic reads in each of the four datasets with different read length and error profiles. 73
3.3 Alignment categories that are improved by iMapSplice over MapSplice. 74
3.4 Sensitivity, specificity, and F_1 score for splice junction discovery on simulated datasets. 75
3.5 Summarized comparison in terms of mean and skewness for reference allelic ratio distributions, and number of SNPs covered by at least ten reads. 78
3.6 Detected personal splice junctions changing from noncanonical to canonical with increasing numbers of supporting individuals and reads. 79
3.7 Number of splice junctions falling into each category of average coverage changes between the individuals with and without mutations in splice sites (a). mutated from noncanonical to canonical; (b) mutated from canonical to noncanonical 80
3.8 Indexing file size and running time to align the 74 RNA-seq dataset reads. Note: indexing file storage usage and run time for HISAT2 SNP and rPGA are estimated according to their performance on a small subset of the datasets. . . . 82

4.1 Disk space, peak memory, and CPU time required by each tool for index construction and sequence query. 101

www.manaraa.com

5.1 Accuracy of read taxonomic classification in terms of precision, sensitivity, and F_1 score. 116
5.2 The proportion of reads classified into the top-five genera by each algorithm using different k-mer lengths. 116
5.3 Data sizes and numbers of reads classified as Zika reads by MetaOthello with two alternative databases. 120

www.manaraa.com

Chapter 1 Introduction

1.1 Biological background

The hereditary information instructing the development and functioning of organisms is stored in deoxyribonucleic acid (DNA). It is a long and double-stranded molecule, and each strand consists of nucleotides or bases. There are four types of nucleotides or bases, denoted as adenine(A), cytosine(C), guanine(G), and thymine(T). The genome refers to the entire set of unique DNAs. As shown in Table1.1, the size of known genomes varies largely. The human genome contains about 3 billion bases.

Table 1.1: Genome sizes of different organisms.

Organism	Common Name	GenomeSize
Arabidopsis thaliana	Mouse-ear cress	100,00,000
Drosophila melanogaster	Fruit fly	165,000,000
Fugu rubripes	Japanese pufferfish	400,000,000
Homo sapiens	Human	3,000,000,000
Mus musculus	Mouse	3,000,000,000
Schizosaccharomyces pombe	Fission yeast	14,000,000

Genes are regions on genomes that encode functional RNAs or protein produces. Each gene is located in a specific location on a chromosome and responsible for a particular function. So far, around 20,000-25,000 protein-coding genes exist in the human genome. They vary in size from a few hundred bases to more than 2,000,000 bases.

Proteins are important functional molecules in living organisms. The synthesis of proteins are guided by genes but not in a direct way. The Figure1.1 shows how genes guide

المنارة للاستشارات
www.manaraa.com

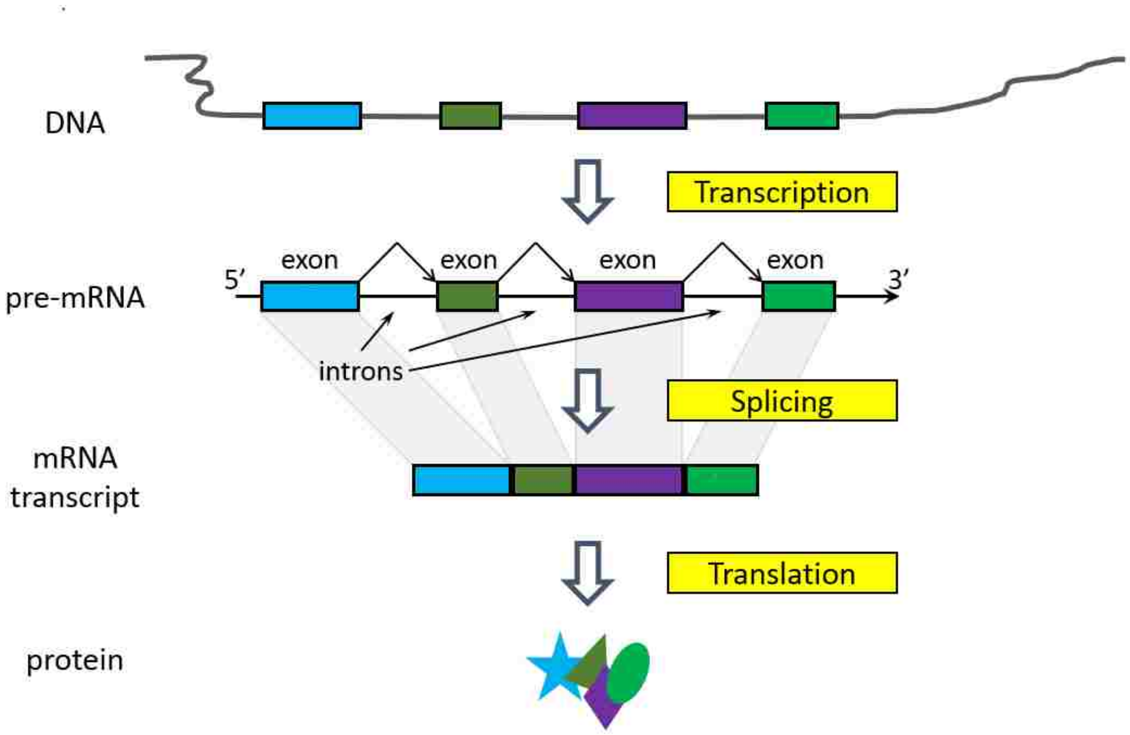


Figure 1.1: Central dogma of molecular biology. The gene shown above contains four exons sparated by three introns. In the first step of transcription, pre-mRNA is created. Second, the pre-mRNA is spliced into mRNA transcript, and this step is called splicing, in which all exons are retained and introns are removed. Third, the mRNA transcript is translated into protein.

the process of protein products creation. The first step is transcription, in which genes are read by an enzyme called polymerases and transcribed into primary RNA transcripts (pre-mRNAs). In the next step of splicing, exons are retained and joined. Introns are excised and mRNA transcripts are generated. Then protein products are finally created after translation of mRNA transcripts.

The transcriptome consists of all messenger RNAs transcribed from the genome of a cell or a population of cells. The elucidation of the transcriptome profiles associated with biological processes and developmental mechanisms is a critical area of biology. Anal-

المنارة للاستشارات
www.manaraa.com

ysis of mRNA transcriptome consisting of mRNA transcripts provides a great chance of studying and revealing the linkage from genotype to phenotype.

1.2 Next generation sequencing and RNA-seq

Recent years, Next Generation Sequencing (NGS), or High Throughput Sequencing, has emerged as one of the main technologies that advance the study of molecular biology. It generates a large amount of short sequence data of genome (Whole Genome Sequencing), exome (Whole Exome Sequencing), transcriptome (RNA Sequencing, or RNA-seq), etc.

RNA-seq directly samples and sequences from the mRNA transcriptome without dependence on predetermined sequence templates. This enables the detection of novel transcripts as no pre-knowledge of gene/transcript sequences are required. At the same time, this sequencing technique pictures transcriptome with a high resolution on base level and makes the accurate quantification of transcripts possible.

As shown in Figure1.2, typically, in an RNA-seq experiment, the probed RNA molecules in the target transcriptome are firstly fragmented into shorter pieces and synthesized into cDNAs, then those cDNA pieces with a proper size would be selected for sequencing. The output of the RNA-seq experiment is the single-end (sampled from one end) or paired-end reads (sampled from both ends), typically of length 50 - 300. Nowadays, to improve sampling coverage at the same sequencing capability, the paired-end sequencing strategy has been widely adopted.

المنارة للاستشارات
www.manaraa.com

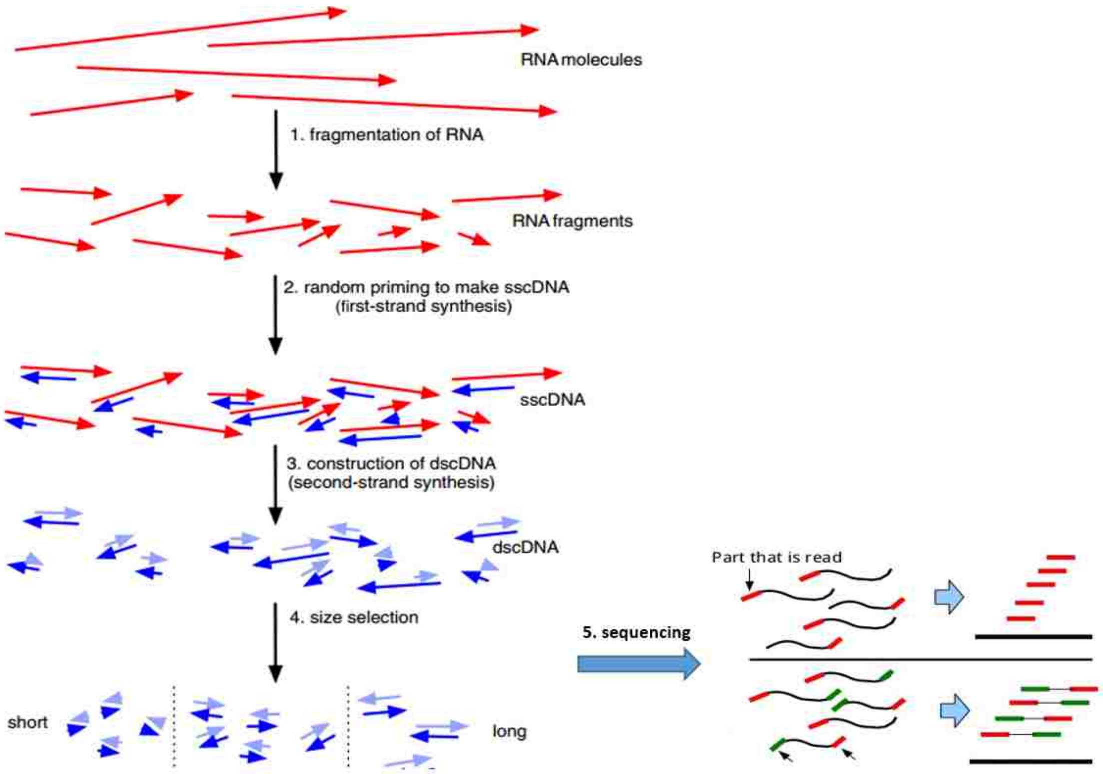


Figure 1.2: Overview of a typical RNA-seq experiment. (1) RNA molecules are fragmented firstly; (2) $3'$ fragment end is utilized to make single stranded cDNA; (3) $5'$ fragment end is selected and double stranded cDNA is created during second-strand synthesis; (4) fragments are categorized based on their size and those with proper size will be used further for sequencing; (5) fragments are sequenced and single-end (sampled from one end) or paired-end reads (sampled from both ends) are generated. (Figure partially adpated from Wikipedia (www.wikipedia.com))

1.3 Current computational methods for Next Generation Sequencing data analysis

As Next Generation Sequencing technology (NGS) plays a more and more prominent role in biological studies, many computational methods have been developed for the analysis of NGS data.

المنارة للاستشارات
www.manaraa.com

1.3.1 The typical RNA-seq data analysis pipeline

In the past decade, RNA-seq has established itself as the major technology in transcriptome profiling. Its capability in deep sequencing RNA transcripts provides an unprecedented detail in identifying the exact variety and quantity of transcript isoforms in a transcriptome. This allows researchers to uncover important gene expression changes on a transcript isoform level. These would have been overlooked in traditional studies where gene expression was measured as a function of transcript products for a particular gene locus. As is shown in Figure 1.3, two types of reads are generated by sequencing machines based on their origins: continuous reads and spliced reads. Continuous reads will fall into a single exon, while the spliced ones would span two or more exons. For a simulated human RNA-seq dataset with typical read length 100bp and realistic parameters [Kim et al., 2015], we would expect around 38% of reads span two or more exons. And this proportion increases significantly from 19% to 46% as read length increases from 50bp to 150bp. The bases covered by read sequences are all exonic ones, and the splice sites could be obtained from spliced reads. The density of reads also reflects the abundance of mRNA transcript. Thus obtaining the origins of reads can help reveal mRNA transcript structure and abundance.

The typical analysis pipeline of RNA-seq data is described in Figure 1.4. The input is RNA-seq datasets, which usually contain tens to hundreds of millions of reads with a typical length of 50-300bp. Four regular applications based on RNA-seq data are listed in the figure, including transcript reconstruction, transcript quantification, differential transcription analysis, and differential gene expression analysis. Many computational software programs have been developed for each of those four applications. For example, Trin-

المنارة للاستشارات
www.manaraa.com

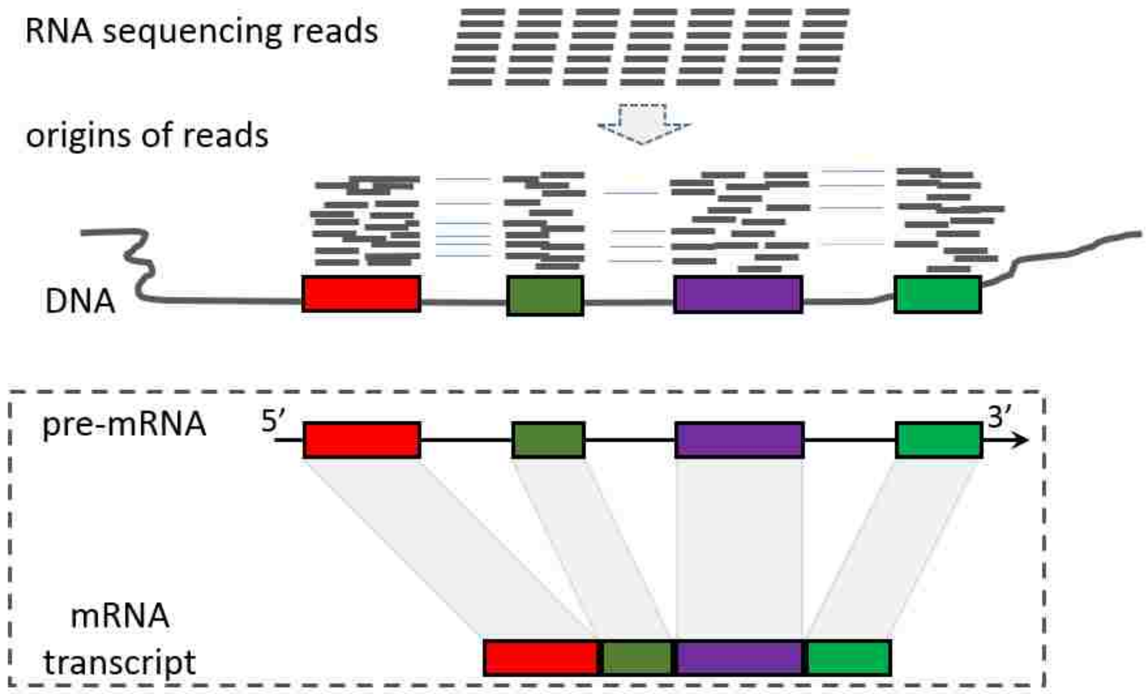


Figure 1.3: Origins of reads reveal mRNA transcript structure and abundance. There are two types of reads according to their origins. The first type of reads are continuous ones and the other type of reads are spliced ones. The bases covered by read sequences are all exonic ones, and the splice sites could be obtained from spliced reads. The density of reads also reflects the abundance of mRNA transcript. The more reads fall into the corresponding genomic regions, the higher expression the mRNA transcript has.

ity [Grabherr et al., 2013] and Cufflinks [Trapnell et al., 2010] can be used to assemble aligned reads into transcripts. RSEM [Li and Dewey, 2011a] and eXpress [Roberts et al., 2011] are two widely used tools in transcript quantification - transcript abundance prediction. Differential transcription events such as alternative splicing and isoform switching can be detected by DiffSplice [Hu et al., 2013] and EBseq [Leng et al., 2013]. For differential gene expression analysis, DESeq2 [Love et al., 2014] and edgeR [Robinson et al., 2010] are the two most popular software.

المنارة للاستشارات
www.manaraa.com

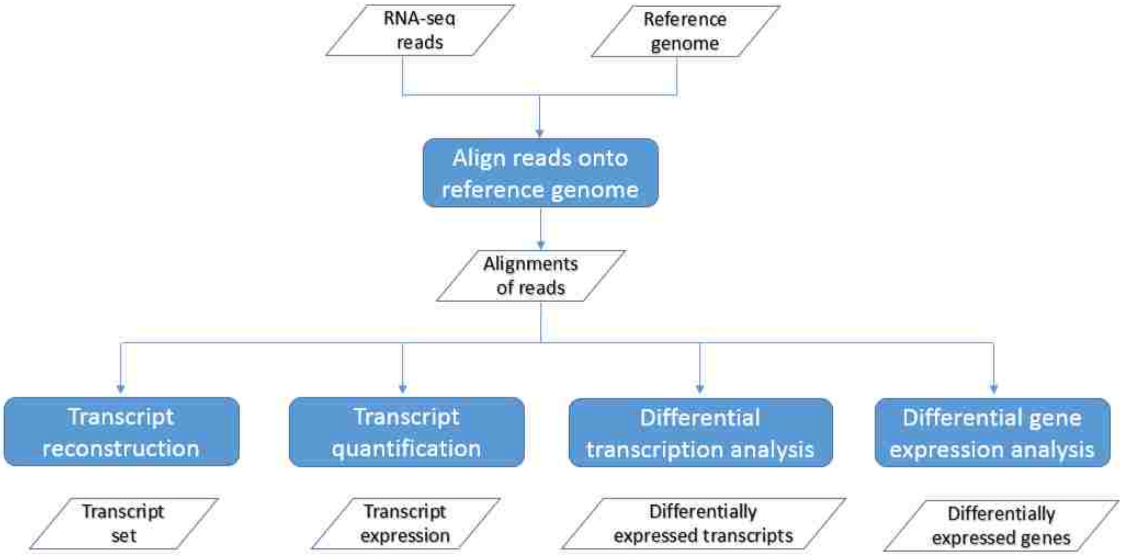

RNA-seq reads
Reference genome
Align reads onto reference genome
Alignments of reads
Transcript reconstruction
Transcript quantification
Differential transcription analysis
Differential gene expression analysis
Transcript set
Transcript expression
Differentially expressed transcripts
Differentially expressed genes

Figure 1.4: Typical RNA-seq data analysis pipeline. Read alignment is the first and fundamental step of all the four common RNA-seq data analysis applications.

1.3.2 RNA-seq read alignment and splice junction detection

For all of those four applications, the first and key step is mapping reads back to the reference genome, i.e. spliced alignment. The existence of spliced reads makes the mapping problem for RNA-seq reads turn out to be more challenging compared to that of DNA-seq reads from two aspects. First of all, RNA-seq spliced aligners have to deal with gapped alignments with very large gaps (introns). Typically, in mammalian genomes, introns span a wide range of length from 50 to 500,000 bases. Secondly, pseudogenes with introns removed would cause many exon-spanning reads to map incorrectly. As shown in Figure 1.5, the pseudogene contains the same exons as the gene with the intron removed. In that case, reads crossing the inron are very likely to be mapped incorrectly to the pseudogene. Besides of the positions of alignments, spliced aligner also reports the differences (mismatches, insertions, and deletions) between the sequencing reads and the reference

المنارة للاستشارات
www.manaraa.com

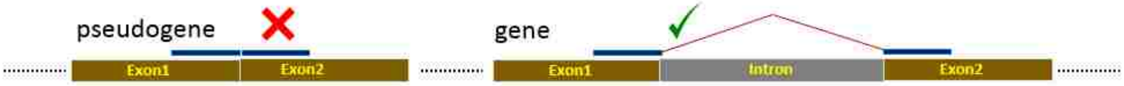


Figure 1.5: An illustration of alignment difficulty caused by the presence of pseudogene.

genome.

In the past few years, as shown in Figure 1.6, numerous next generation sequencing read aligners have been developed, including but are not limited to Tophat [Trapnell et al., 2009, Kim et al., 2013], MapSplice1/2 [Wang et al., 2010], GSNAP [Wu and Nacu, 2010], STAR [Dobin et al., 2013], HISAT [Kim et al., 2015] and some other.

Before the advent of spliced aligners designed specifically for RNA-seq reads, some general alignment tools like BLAST [Altschul et al., 1990] and the BLAST-like tools such as BLAT [Kent, 2002] are used to do RNA-seq read alignment. However several problems limit their application on aligning RNA-seq reads against the reference genome. First of all, RNA-seq reads are too short, and they can not handle reads with that size properly. Secondly, splicing signals (flanking strings for most splice site are “GT/AG”) are not considered in the algorithms, which impairs the accuracy of splice junction detection. Thirdly, they are far way too slow to finish the alignment task of tens to hundreds of millions of reads in a limited time.

The first set of spliced alignment tools including Tophat, GSNAP, and MapSplice were developed around the year 2010. They are mainly designed for the very short reads (36 to 50bp) generated by the very early sequencing machines. Their performance dramatically drops when applied to relatively longer reads with the length more than 200bp. Among

المنارة للاستشارات
www.manaraa.com

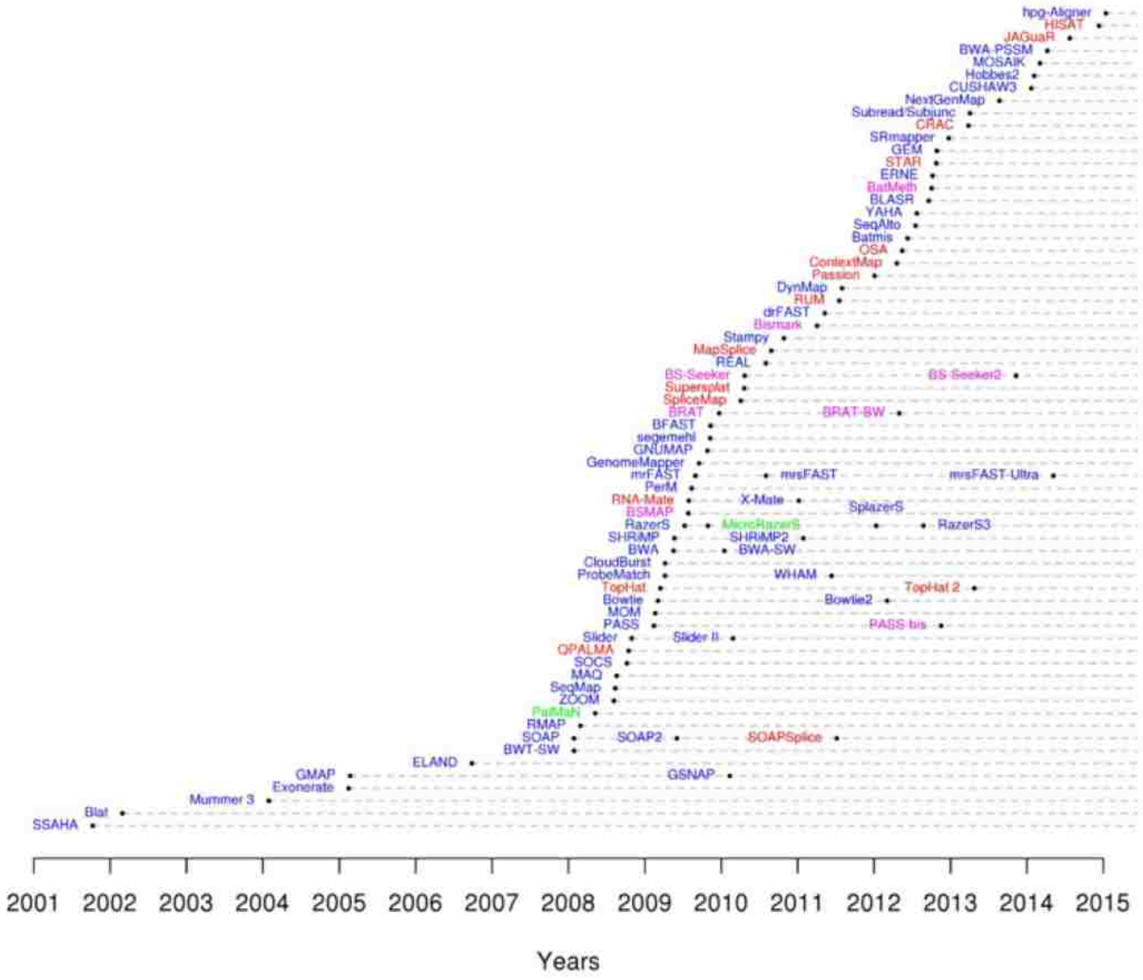


Figure 1.6: Aligners timeline since 2001. DNA mappers are plotted in blue, mRNA aligners (spliced aligners) are plotted in red and miRNA mappers in green and bisulphite mappers in purple. Grey dotted lines connect related mappers (extensions or new versions).

them, Tophat and MapSplice firstly perform exonic continuous alignment with pre-split read segments based on unspliced aligner Bowtie [Langmead and Salzberg, 2012]. The indexing data structure used by Bowtie is FM index of Burrows-Wheeler Transformer (BWT). GSNAP is a seed-extend spliced aligner and uses all 12-mers in the read to identify candidate mapping locations. It would first extract the seeds of reads and scan in the hashtable with exact matching positions in reference genome returned. Then these seeds

www.manaraa.com

would be extended until the whole read sequence is covered. Partially due to the inefficient indexing structure, these tools are also not scalable enough to handle the large and ever-increasing sequencing datasets. As RNA-seq technology evolves, the sequencing throughput has increased dramatically to 100-500 million reads per run. As mentioned before, obtaining the alignments of reads is the first step in many RNA-seq applications, but this time-consuming may take several days to process a regular RNA-seq experiment data with these first generation spliced aligners.

Recently, two aligners STAR and HISAT are introduced to provide a probability of performing super-fast read alignment. STAR uses uncompressed suffix array as the index to accelerate the whole algorithm. Though the size of uncompressed suffix array is much larger than that of BWT-FM index (STAR requires 32 GB RAM for alignment against human genome), it is still acceptable as more and more high performance computing resources are used in genomics research and the price of RAM becomes lower and lower. STAR firstly perform Maximal Mappable Prefix search iteratively from the beginning to the last base of read sequences. Then those aligned maximal prefixes are clustered and stitched to generate alignments as well as the corresponding alignment score. HISAT adopts a new indexing scheme of hierarchical structure based on Burrows-Wheeler transform and FM-index. It consists of one global FM index and around 48,000 local FM indexes for the human genome, and each local FM index represents a genomic region of 64,000bp. It also uses Bowtie to handle exonic alignment as Tophat1/2. To map a read sequence onto the reference genome, HISAT starts with applying exonic alignment with selected segments from the read sequence to locate the candidate mapping regions based on the global index. Then the local indexes of the candidate regions are used to finalize the alignment. The

www.manaraa.com

far smaller size of the local index allows itself to fit in the cache as a whole, which considerably reduces the cache misses that frequently happen when applying exact matching against the global index.

Here, we categorized existing published spliced aligners based on types of the index they are using. Three kinds of indexes are widely used, including *k*-mer hash table, BWT-FM index, and suffix array.

***K*-mer hash table:** In *k*-mer hash table, the keys are the k-mer sequences, and the values are the positions they occur in the reference genome. The length *k* of the *k*-mer sequences has a large impact on the performance of that kind of methods. The smaller *k* is, the more sensitive the algorithms would be, and the more computational time would be required. Most tools in this category adopt the strategy of seed-extension. The algorithms would first extract subsequences of reads and scan in the hashtable with exact matching positions in reference genome returned. Then these seeds would be extended until the whole read sequence is covered. Different tools come up with different seeding methods: some of them use non-overlapped seeds while the others use overlapped seeds; some of them require exact matching between seeds and reference genome while some of them allow internal mismatches. GSNAP [Wu and Nacu, 2010], OLego [Wu et al., 2013], and Subread [Liao et al., 2013] are the representatives in that category.

BWT-FM: FM index of Burrows-Wheeler Transformer (BWT) [Salson et al., 2008] are used in many commonly used unspliced aligners like BWA [Li and Durbin, 2009] and Bowtie [Langmead and Salzberg, 2012]. Some spliced aligners including Tophat [Trapnell

www.manaraa.com

et al., 2009]), MapSplice [Wang et al., 2010], and HISAT [Kim et al., 2015] make use of those unspliced aligners for read segment alignment. BWT-FM index is typically 0.5 to 2 bytes per nucleotide, depending on details of implementation. For example, human genome takes 2 to 8 GB of memory.

Suffix array: As more and more high performance computing resources are used in genomics research and the price of RAM becomes lower and lower, tools with larger memory requirements are emerging as most of them come up with higher mapping speed. Many of them are taking suffix array as indexes, such as STAR [Dobin et al., 2013] and Segmenhlp [Hoffmann et al., 2009].

There have been some published literature [Engström et al., 2013, Grant et al., 2011] performing comprehensive comparisons between existing spliced aligners. No single tool dominantly performed the best. But Tophat1/2 [Trapnell et al., 2009, Kim et al., 2013], MapSplice1/2 [Wang et al., 2010], STAR [Dobin et al., 2013], and GSNAP [Wu and Nacu, 2010] outperform others in most evaluation metrics and are also current widely used ones.

1.3.3 Gene fusion and circular RNA detection using RNA-seq

Fusion genes are hybrid genes resulting from breakage and rejoining of two previously separate genes, which are always related to chromosome rearrangements, translocation, and inversion. Though fusion transcripts appear to be rare compared to regular transcripts, they potentially undertake important biological functions. Some of the fusion genes have been proven to be cancer indicating events such as the TMPRSS2-ERG fusion in prostate cancer [Tomlins et al., 2008]. When reads spanning fusion breakpoints are mapped back

المنارة للاستشارات
www.manaraa.com

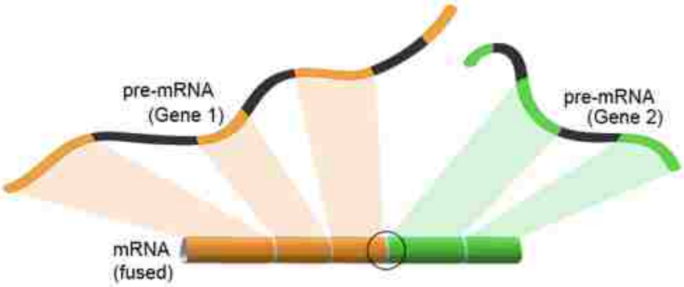


Figure 1.7: Fusion transcripts are formed from two separate transcripts. In this figure, three exons from the pre-mRNA in Gene 1 and two exons from the pre-mRNA in Gene 2 formed the fusion transcript. (Figure partially adapted from Wikipedia (www.wikipedia.com))

to reference genome correctly, they will be split into two or more inter-gene segments. Most of the existing spliced aligners lack the ability of handling that kind of alignments. Though some fusion transcript detection tools like Tophat-Fusion [Kim and Salzberg, 2011], deFuse [McPherson et al., 2011], and SoapFuse [Jia et al., 2013] have been developed and published in recent years, many problems remain unsolved. First of all, most of those tools take regular spliced alignment results as input, but this “post-processing” methods are biased naturally as many of fusion reads would be forced to map onto reference genome collinearly which impairs the sensitivity. Secondly, their performance largely depends on how well the embedded regular spliced aligner works. But as mentioned previously, many of the regular spliced aligners are not ready for the data from the evolving sequencing platforms. Last but not the least, false discovery rates for the current tools are still too high, partly due to the “whole genome spread” features of fusion reads.

Reads from circular RNAs are also always be neglected by existing spliced aligners. A previous study shows that circular RNAs are the predominant transcript isoform for hundreds of human genes in diverse cell types [Salzman et al., 2012]. The key point of circular RNA detection is the correct alignment of back splicing reads. In Figure 1.8, exon2

المنارة للاستشارات
www.manaraa.com

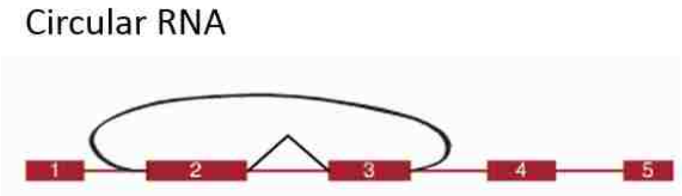


Figure 1.8: An example of circular RNA comprised with two exons from the regular transcript.

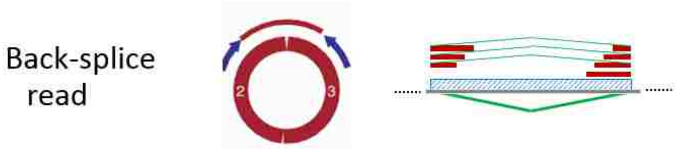


Figure 1.9: Origins of back-splicing reads on circular RNA and their alignment on reference genome.

and exon3 comprise the circular RNA, and can be identified with the back-splice reads and their alignments in Figure 1.9. In the past two years, though circular RNA detection tools are emerging such as CIRI [Gao et al., 2015] and KNIFE [Szabo et al., 2016], most of them not only share the same problem with fusion detection tools – high dependency on regular spliced aligners, but also are limited in their functionality – only back-splices between annotated exons are considered [Gao et al., 2015].

1.3.4 Large scale RNA-seq data query

The past decade has witnessed an explosion of sequencing data fuelled by a significant drop in sequencing cost. The growth of sequencing data has surpassed Moore's law - the expected growth of the computation speed - i.e., more than doubling every year [Kahn, 2011]. The Sequence Read Archive (SRA) has established itself as an invaluable sequencing data repository. It contains over 40,000 RNA-seq samples from more than 7,000 projects, with petabytes of data from a broad range of species, experiment conditions, and sequencing

المنارة للاستشارات
www.manaraa.com

technologies.

Sharing data through SRA ensures reproducibility. More importantly, by reusing and re-purposing existing datasets, researchers can generate and test new hypotheses without having to repeat similar sequencing experiments, which are often time-consuming and cost-inefficient. However, accessing the sequencing data through SRA is not as convenient as one may prefer. The search function in SRA only supports metadata queries, providing links to relevant sequencing datasets. Obtaining biological results still requires downloading dozens of gigabytes of data, if not more, and tremendous bioinformatics support to reanalyze the data.

An emerging computational problem with an urgent but unmet need is how to query large transcriptome sequencing data, which are highly diverse in different tissues, diseases, and experimental conditions besides species. For example, a biologist has discovered some novel transcript isoforms in his/her experiment, and he/she may want to explore how commonly this previously uncharacterized transcript may occur in existing SRA datasets. As of now, such a population-level query cannot possibly be answered without downloading and reanalyzing all sequencing datasets, which is infeasible. Computational algorithms capable of supporting such routine query will bridge a considerable gap between biologists and the sequencing data and have potential to impact how biologists conduct research every day.

In the past year, numerous efforts have been pioneered to facilitate queries over large-scale RNA-seq data. These efforts can be roughly divided into two categories: supporting query over pre-analyzed results and enabling query against raw sequencing reads. The first approach often provides a search engine on top of comprehensive genomic features summarized from the analysis of each sample using state-of-the-art bioinformatics pipelines.

المنارة للاستشارات
www.manaraa.com

One such example is Snaptron [Wilks et al., 2018], which supports the query of splice patterns generated from large-scale pre-analyzed human RNA-seq data. The latter approach, however, aims to build indexes that allow for the query against raw sequencing data. It is alignment-free, reference-free, and the most faithful to the sequencing data. SBT (Sequence Bloom Tree) is the pioneering approach in making large-scale sequence search feasible [Solomon and Kingsford, 2016]. SBT can be considered as a binary tree with each node implemented as a bloom filter. Each bloom filter is a compact, probabilistic data structure for storing a large set of *k*-mers present in one or more samples. Within the tree, each leaf node stores the information of a single sequencing sample, while each internal node summarizes all the *k*-mer information of all samples in each branch. Sequence query over SBT is a depth-first traversal of the tree - visiting relevant nodes in the tree one after the other - and avoids the need to load the entire SBT into the memory at once. Although SBT owns a binary tree structure, it does not naturally lend itself to anything like the regular binary search. This is because the content of any two dividing branches is not disjoint. A query over the tree may require the traversal of the entire tree rather than a single path as in typical binary search.

1.3.5 Taxonomic classification of metagenomics sequencing data

Metagenomics is the study of genomic content obtained in bulk from an environment of interest, such as the human body [Huttenhower and Human Microbiome Project Consortium, 2012], seawater [Venter et al., 2004], or acidic mine drainage [Tyson et al., 2004]. Metagenomics studies often generate tens of millions of sequencing reads to capture the presence of microbial organisms and quantify their relative abundances, rendering the classification

المنارة للاستشارات
www.manaraa.com

and analysis of these data a logistical challenge.

One of the major computational challenges in the analysis of metagenomic data is the classification of each sequencing read into the most-specific biological taxon to which sequence conservation supports its assignment. Specifically, a read is classified as belonging to a taxon if it has high sequence similarity with the reference genomes collected for that taxon, a process made possible by the large deposits of reference sequences collected in recent years for a variety of microbial species. In 2014 alone, more than 10,000 sequence records were newly added to the NCBI RefSeq database thanks to the accessibility of NGS technology.

Existing classification methods can be divided into two broad categories: alignment-based and alignment-free. The former approach, implemented most popularly as BLAST [Altschul et al., 1990], assigns each read to the taxon that affords the best alignment with its reference genomes. Several methods, including MEGAN [Huson et al., 2007], PhymmBL [Brady and Salzberg, 2009], and NBC [Rosen et al., 2011], apply additional machine-learning techniques to BLAST results to increase classification accuracy. These methods are often slower than BLAST alone, rendering them computationally prohibitive for large-scale analysis of many millions of short reads. However, the recent development of Centrifuge [Kim et al., 2016] has significantly improved the scalability of the alignment-based algorithm using FM-index. Besides using genomic sequences as the reference, the recently published tool Kaiju [Menzel and Krogh, 2015] performs alignments towards protein sequences, achieving faster classification speed than existing tools.

The other line of work, pioneered by LMAT [Ames et al., 2013] and Kraken [Wood and Salzberg, 2014], classifies a read using exact *k*-mer matches between the read and ref-

www.manaraa.com

erence sequences belonging to the target taxon, thereby avoiding inefficient base-by-base alignment while maintaining sensitivity and specificity comparable to the alignment-based approach. This approach is faster than alignment-based methods and allows for greater flexibility in reference material because it requires only the collection of *k*-mers extracted from reference sequences belonging to each taxon. Thus *k*-mers extracted from DNA or RNA sequencing data can be included as reference material without being assembled, increasing the sensitivity of the algorithm in capturing natural variants that are often missed using reference genomes alone.

1.4 Dissertation Statement

This dissertation aims to build novel computational methods for Next Generation Sequencing data analysis. It covers three different but closely related areas, including: accurate and comprehensive RNA-seq read sequence mapping for transcriptome characterization, scalable sequence query over large-scale RNA-seq data, and efficient taxonomic classification of metagenomics sequencing data.

1.5 Contributions of this dissertation

We have developed a series of novel computational methods for Next Generation Sequencing data analysis, including sequence mapping, query, and classification. And the performance of each of those methods is assessed using a number of simulated and real datasets. The experiments demonstrate their advantages on sensitivity and accuracy of sequence analysis, and computation-wise performance (speed, storage, and memory usage) com-

المنارة للاستشارات
www.manaraa.com

pared to other state-of-the-art methods. This dissertation may contribute to the following four areas.

Accurate alignment of RNA-seq reads and detection of splice junctions, gene fusions, and circular RNAs The major innovation of MapSplice3 is its context-aware alignment through self-learning. This improves both the sensitivity and specificity of read alignment as well as the mapping completeness of the whole read. Context-aware alignment refers to subject-specific transcriptome features which include genome-wide transcriptome splicing structure and structural variation. Importantly, Context is learned from the intermediate alignment generated in the first phase of MapSplice3. These intermediate alignments may be incomplete, but are reliable because relatively long segments or clusters of segments are required to avoid random alignments in that phase. These reads are then used collectively to derive transcriptome features of the subject. Additional filtering approaches are applied to minimize errors in the context. Using that context data overcomes the difficulty of completing and solidifying alignments with a pure reference genome. In addition to regular transcript read alignment, MapSplice3 simultaneously detects gene fusion and circular RNA transcripts accurately. Different from the other gene fusion/circular RNA detection approaches that serve as a separate and post-processing step, MapSplice3 provides a systematic and unified solution to align reads from both regular and irregular transcripts.

Individualized RNA-seq read alignment for personal splice junction detection and unbiased reference allele ratio estimation Existing algorithms that utilize a standard reference genome as a template sometimes have difficulty in mapping reads that carry ge-

www.manaraa.com

nomic variants. These problems can lead to allelic ratio biases and the failure to detect splice variants created by splice site mutations. To tackle those challenges, we have developed a novel approach called iMapSplice that enables personalized mRNA transcriptome profiling. The algorithm makes use of personal genomic information and performs an unbiased alignment towards genome indices carrying both reference and alternative bases. Importantly, this breaks the dependency on reference genome splice site dinucleotide motifs and enables iMapSplice to discover personal splice junctions created through splice site mutations. Also, the experiments demonstrate that iMapSplice significantly alleviates allelic ratio biases with minimal overhead in computation time and storage.

Super fast transcript sequence query over large scale datasets The third computational method described in this dissertation is one of the first tools that support super-fast large-scale sequence query against thousands of datasets. It is two orders of magnitude faster than current state-of-the-art tools without loss of accuracy and compromises in memory usage. It achieves great compression ratio and can support query over tens of terabytes raw sequencing data using a standalone desktop computer. We conducted two experiments: annotated protein-coding transcript query over 148 equine RNA-seq datasets and a global survey of gene fusion transcript sequences over the whole TCGA Pan-Cancer RNA-seq datasets. Both of the two experiments demonstrate its capacity of providing accurate and efficient reference-free, alignment-free, and parameter-free sequence queries against a large number of datasets.

المنارة للاستشارات
www.manaraa.com

Efficient taxonomic classification of metagenomics sequencing reads Metagenomic read classification is a critical step in the identification and quantification of microbial species sampled by high-throughput sequencing. Although many algorithms have been developed to date, they suffer significant memory and computational costs. Due to the growing popularity of metagenomic data in both basic science and clinical applications, as well as the increasing volume of data being generated, efficient and accurate algorithms are in high demand. We introduce MetaOthello, a probabilistic hashing classifier for metagenomic sequencing reads. The algorithm employs a novel data structure, called l-Othello [Yu et al., 2017], to support efficient query of a taxon using its k-mer signatures. MetaOthello is an order-of-magnitude faster than the current state-of-the-art algorithms Kraken and Clark and requires only one-third of the RAM. In comparison to Kaiju, a metagenomic classification tool using protein sequences instead of genomic sequences, MetaOthello is three times faster and exhibits 20-30% higher classification sensitivity.

The open-source software packages for the algorithms developed in this dissertation are open source and publicly available to the research community.

Copyright© Xinan Liu, 2018.

www.manaraa.com

Chapter 2 Comprehensive RNA-seq read alignment with MapSplice3

A critical and fundamental step of mRNA-seq analysis is to map the RNA-seq reads onto a reference genome correctly. Evolving sequencing technologies are generating longer reads, and current software programs (aligners) are not capable of dealing with these data. In this dissertation, we introduce novel computational solutions in a new version of the splice alignment tool MapSplice. MapSplice version 3.0 (MapSplice3) achieves high read alignment and base mapping yields while providing high sensitivity and specificity of splice junction discovery. In addition to the regular mapping of reads to a reference genome, MapSplice3 performs context-aware local alignments to detect spliced structures and SNPs. A novel splice junction classification method also significantly improves the accuracy of identifying splice junctions. Comprehensive comparisons with RNA-seq datasets, which include varying read lengths and error profiles, demonstrate that MapSplice3 is an accurate and sensitive RNA-seq aligner.

2.1 Introduction

In parallel to the evolution of sequencing technologies, there has been a rapid development of computational tools to analyze high throughput transcriptome data. One of the first and most critical steps in analyzing sequencing data is the alignment of raw reads back to the selected reference genome. In the past few years, numerous software programs for the alignment of read sequence have been developed and refined. These include, but are not limited to, Tophat 1/2 [Trapnell et al., 2009, Kim et al., 2013], MapSplice1/2 [Wang et al.,

www.manaraa.com

2010], GSNAP [Wu and Nacu, 2010], Subread [Liao et al., 2013], OLego [Wu et al., 2013], STAR [Dobin et al., 2013], and HISAT [Kim et al., 2015]

While existing aligners have been quite successful in supporting traditional applications such as gene expression measurements, challenges in attaining accurate read alignments to support more advanced RNA-seq applications remains a significant unmet need in the arena of RNA-seq analyses. Research utilizing RNA-seq technology has repeatedly demonstrated that changes in biological function are often reflected in the transcriptome and can provide valuable insight into disease pathogenesis. However, an area that lacks systematic support from RNA-seq aligners is the detection of transcripts that break the normal and natural form of mRNAs involving upstream exons followed by downstream exons from the same gene locus. Gene fusion is one such example where transcripts from two remotely located gene loci are joined together. Fusions have been identified as biomarkers in a variety of tumor types with clinical relevance. A second example is circular RNA where downstream exons are found to be contiguous with upstream exons. Both fusion transcripts and circular RNAs are excluded from conventional spliced aligners. While dozens of tools specialized in gene fusion or circular RNA detection have been developed. Most of these tools conduct a post-processing of unmapped reads that failed to be resolved by aligners. Such approaches, however, cause many issues. First, the set of unmapped reads from different aligners may vary significantly as their mapping sensitivities are never the same, and as a result, so are the parameters to dispense unmapped reads. Secondly, non-linear

www.manaraa.com

transcripts co-exist with their regular counterparts and often share a significant portion in sequence as well as splice donor and acceptor sites. Detection of them sequentially at different stages of the analysis misses the opportunity to use the information from one to disambiguate and optimize the other. Additionally, more stringent criteria are frequently applied in the detection of fusion reads to avoid false positives. This will result in lower read coverage in these junctions, biasing the downstream transcript quantification analysis.

Another area where deviation from the reference genome sequence may be problematic is in the study of allele-specific expression (AE). The analytical goal of AE is to determine the relative expression of specific transcripts from each of the two parental haplotypes. Typical AE analysis seeks to capture this information by differentiating the counts of RNA-seq reads carrying either the reference or alternative alleles at heterozygous sites in an individual. As such, accurate read alignment is critical to derive the correct relative AE. However, current aligners are often biased in favor of mapping reads that are consistent with reference genome, since nucleotide variants in the alternative alleles are considered as mismatches to the reference genome. The bias in the read alignment may ultimately affect the accuracy in quantifying AE in many sites of interest [Castel 2015].

Also, further advancement of sequencing technologies and read lengths is poised to challenge alignment software programs in the near future. We have witnessed the extension of most short-read data from 25bp to 100-150bp within the past decade. This progressive increase in read length has helped in disambiguating read alignments, particularly in repetitive regions of the genome. Today, the Illumina MiSeq platform is able to produce 300bp reads with high quality. It is expected that similar read lengths may become the standard output for the more commonly used Illumina HiSeq platform in the near future.

المنارة للاستشارات
www.manaraa.com

Importantly, these improvements will endow greater accuracy to transcript reconstruction and quantification, splice junction detection, as well as structural variation calling. However, a current limitation of state-of-the-art aligners is that they are optimized for reads 100bp in length. As read lengths increase beyond 150bp, the accuracy of these programs for both alignment and splice junction detection declines dramatically. Longer reads result in a higher chance to at least partially map the reads to a reference genome. The challenge becomes determining the exact composition of an alignment, i.e., what specific transcript features an alignment consists of. These features include splice junctions (donor and acceptor sites) and structural variations including SNPs, small indels, and fusion transcripts. Any given read may contain a combination of these features. With the advent of longer reads, resolving the complexity of these features using gapped alignment may result in increased detection of false positive junctions. This will be made more challenging if lower quality base calling is present in longer reads. As such, obtaining end-to-end complete alignments with accurate transcriptome feature composition becomes an even more difficult task. Given these challenges, an alignment program that can achieve excellent alignment accuracy and splice junction detection for RNA-seq reads longer than 100bp is an emergent need.

MapSplice3, a new version of MapSplice software, provides a systematic and unified one-stop solution to align both regular and irregular spliced reads. MapSplice3 is suited to align both short ($\leq$ 100bp) and medium-sized reads ($\leq$ 500bp). The major innovation is context-aware alignment through self-learning. Context refers to subject-specific transcriptome features that include genome-wide transcriptome splicing (regular and irregular) structure and structural variations. MapSplice3 learns the individual context during the

www.manaraa.com

alignment process and uses this information for more sensitive and unbiased mapping.

The implementation of MapSplice3 utilizes a two-level suffix array-based index structure for both global and local alignments. It is highly scalable, with speed and memory competitive to current state-of-the-art aligners for regular spliced alignment. If the detection of irregular transcripts is also considered, it is orders of magnitude faster than existing two-step pipelines. MapSplice3 is also easy to use and parameter free. In this dissertation, we compare the performance of MapSplice3 and a number of popular alignment programs using both simulated and real datasets. MapSplice3 outperformed other software tools in accuracy of splice junction detection especially on challenging datasets, such as those with longer reads and higher error rates. Such performance extends to the alignments of reads sampling gene fusion and circular junctions, as well as reads carrying alternative alleles.

2.2 Overview of MapSplice3 algorithm

MapSplice3 adopts a two-phase alignment strategy. In phase I, the program maps each read individually against the entire reference genome to identify candidate mapping positions. Transcriptome context is learned from the candidate alignments in this phase. Although these alignments may be incomplete, they are reliable because relatively long segments or clusters of segments are required to avoid random alignments in this first phase. Additional filtering approaches are applied to minimize errors in the context. In the second phase, MapSplice3 uses the transcriptome context to complete partially aligned reads from Phase I. The context serves two purposes: firstly, it guides the alignment towards a subject-specific transcriptome structure. This increases the sensitivity and completeness of the alignment. Secondly, spurious alignments that result from random sequence matches are

المنارة للاستشارات
www.manaraa.com

reduced. An overview of the MapSplice3 pipeline is shown in Figure 2.1.

2.3 Global alignment with semi-maximal mappable prefix search

MapSplice3 first looks for long segments of a read that can be mapped continuously to the reference genome, as shown in step 1, Figure 2.3. This is done by repetitively applying prefix searches starting at the beginning of a read using a suffix array index of the entire reference genome. The matching of a prefix stops at the base in a read where it does not match anywhere in the reference genome. This is typically due to mismatches, indels or splice junctions. The algorithm will then jump over this base and restart the prefix search for the rest of the read sequence until the last base of the read is included in a prefix.

The prefix search employed by MapSplice3 is not the typical maximal prefix search, in which only the mapping location of the longest match is returned. MapSplice3 returns all mapping positions with a match longer than a certain threshold (set at 30bp by default). We call this semi-maximal prefix searching, an approach chosen because the longest sequence match does not always lead to the correct alignment. For example, reads spanning two exons from a gene may map continuously to a related pseudogene locus, where the two exons are already adjoined and the intron between them absent. In this case, a maximal prefix search strategy would only return the mapping position in the pseudogene and ignore the mapping in the parent gene. In contrast, the semi-maximal prefix search will include both loci as candidate mapping locations. In general, this method is much more sensitive in recovering alignments to regions with high sequence similarity.

Iterative semi-maximal prefix searching naturally divides read sequences into segments of variable lengths. Although a minimum alignment of 20bp is required for any cluster

المنارة للاستشارات
www.manaraa.com

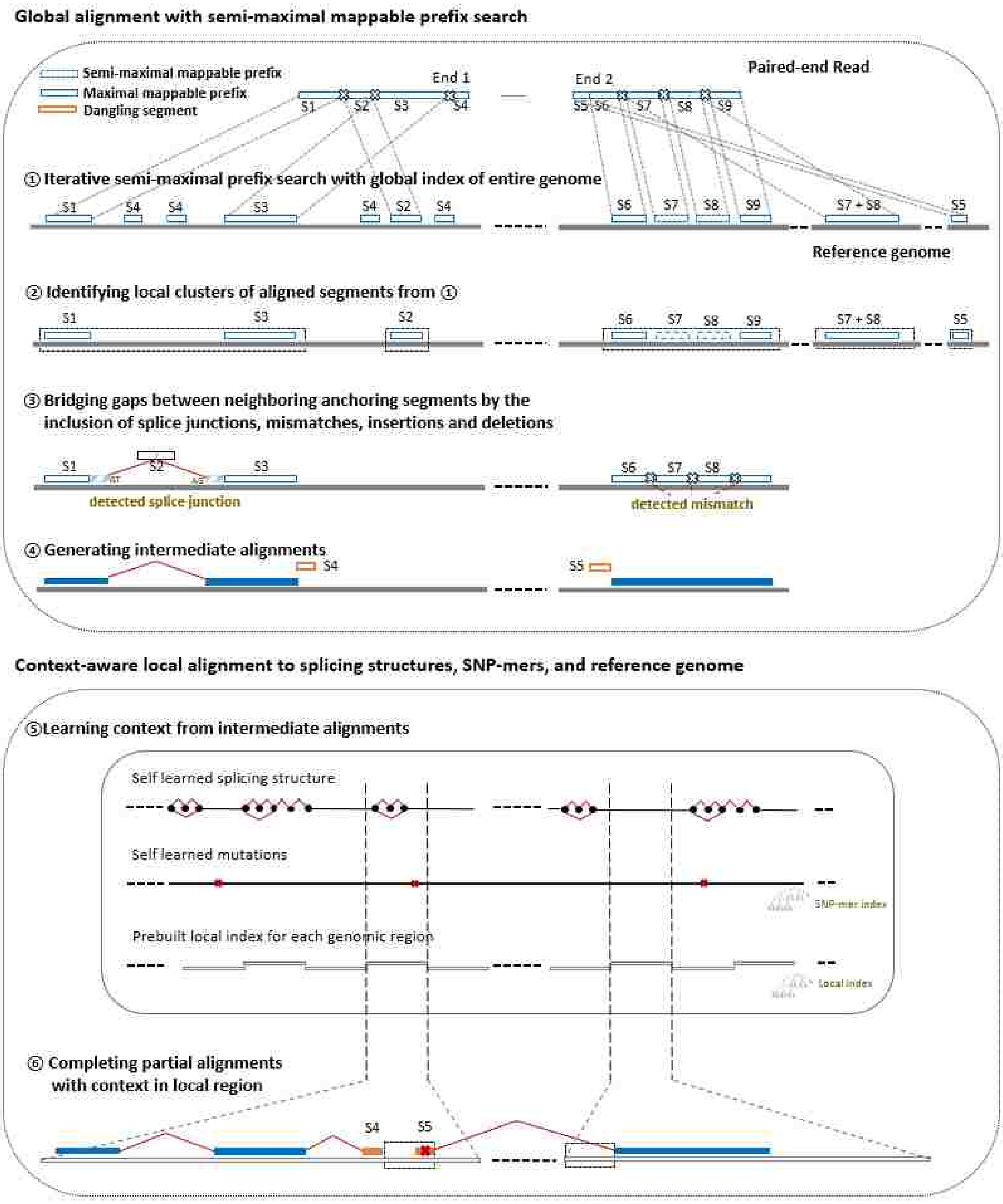

Global alignment with semi-maximal mappable prefix search
Semi-maximal mappable prefix
Maximal mappable prefix
Dangling segment
End 1
S1 S2 S3 S4
End 2
Paired-end Read
S5 S6 S7 S8 S9
① Iterative semi-maximal prefix search with global index of entire genome
S1 S4 S4 S3 S4 S2 S4
S6 S7 S8 S9 S7 + S8 S5
Reference genome
② Identifying local clusters of aligned segments from ①
S1 S3 S2
S6 S7 S8 S9 S7 + S8 S5
③ Bridging gaps between neighboring anchoring segments by the inclusion of splice junctions, mismatches, insertions and deletions
S1 S2 S3
GT AG
detected splice junction
S6 S7 S8
detected mismatch
④ Generating intermediate alignments
S4
S5
Context-aware local alignment to splicing structures, SNP-mers, and reference genome
⑤ Learning context from intermediate alignments
Self learned splicing structure
Self learned mutations
SNP-mer index
Prebuilt local index for each genomic region
Local index
⑥ Completing partial alignments with context in local region
S4 S5

Figure 2.1: An overview of MapSplcie3 algorithm. MapSplice3 algorithm contains two tiers: global alignment with semi-maximal mappable prefix search (1∼4), and context-aware local alignment to splicing structures, genomic sequences with mutations, and reference genome (5 and 6).

www.manaraa.com

to be considered further, some segments are relatively long, uniquely mapped, and more supportive of the read alignment. In contrast, other segments, especially shorter segments, can often be mapped to multiple positions in the reference genome. Segments with too many mapping possibilities (such as segment S4 in Figure 2.1) would be considered as unmapped at this step. Their alignment will be resolved in step 3 of Phase I or in Phase II.

Segment alignments are clustered together if they are closely positioned on the genome and if their relative order is consistent with that in the read (step 2, Figure 2.1). It is noteworthy that there can be more than one cluster for each read. This often corresponds to multiple candidate locations in repetitive regions of the genome. At each candidate cluster, there may be gaps between two neighboring segments. If the length of a gap falls in the range of a potential intron, a spliced alignment is attempted. The spliced alignment seeks a split position within a segment that maximizes the sequence similarity between the divided segments and the corresponding donor and acceptor sequences in the reference genome (step 3 of Phase I in Figure 2.1). In case of unusually high mismatches around a splice site, a second alignment allowing small indels or even a small exon will be conducted and accepted if a significant reduction of mismatches can be achieved. MapSplice3 uses both canonical and semi-canonical splice site flank strings (GT-AG, GC-AG, AT-AC) to help determine the split position in the read. Non-canonical splice sites are allowed in the absence of canonical or semi-canonical ones.

If a read spans two gene loci as a result of a gene fusion event, its alignment may consist of two clusters that are located at a distance on the same chromosome, on different chromosomes, or with different strand orientations. In the case of circular RNAs, the alignment of two neighboring segments will be in the reverse order of each alignment.

المنارة للاستشارات
www.manaraa.com

Based on the segment mapping in step 2, MapSplice3 is able to identify these events from the fact that two neighboring segments in the read are mapped to positions in the genome that are inconsistent with regular transcripts. With the extraction of appropriate donor and acceptor site sequences, the spliced alignment algorithm is applied to identify the exact splice site by matching the corresponding unmapped subsequences from the read to the donor and the acceptor sequences in the reference genome.

The last step in Phase I assembles aligned segments of a read and, if applicable, combines it with the alignment from the paired read at the other end to achieve a mate-pair.

2.4 Context-aware local alignment

Many reads obtain complete alignment at the end of Phase I. However, a significant proportion of reads remain partially aligned or unmapped. This is exacerbated by longer reads. In a partial alignment, the mapping of one or more parts of the read is missing. Frequently this includes a trailing prefix or suffix of a read as depicted by S4 and S5 in Step 4, Figure 2.1. Processing these segments during the global alignment phase may result in spurious alignments and significantly increase the running time. The number of short, ambiguously mapped segments is reduced by localizing them to the region identified by the aligned portion of their read and its pair. Phase II of the MapSplice3 program leverages this localized information to complete the alignment.

Phase II alignment is termed Context-Aware Local Alignment. This localized alignment involves two approaches. Firstly, MapSplice3 attempts to align unmapped sequences to the reference genome in the localized region using similar steps to Phase I. Unlike Phase I, however, where the search space is the whole genome, in Phase II, the search space is

المنارة للاستشارات
www.manaraa.com

restricted to the region where the rest of the read and the corresponding mate-pair have already been mapped. This significantly reduces the number of spurious alignments for short segments. Secondly, a short trailing segment of a read is often unmapped due to one of the following reasons: (1) it spans a splice or fusion junction, or (2) it includes a variant allele relative to the reference base. If these features were known a priori, it would be possible to target the alignment directly. Frequently, however, these features are not provided as input. Nonetheless, given the depth of the RNA-seq data, these transcriptome features may already be available from other reads that were successfully mapped in Phase I. Thus, MapSplice3 can derive transcriptome features from Phase I alignments. In comparison to the first approach, which is an ab initio alignment based entirely on sequence match, the approach in Phase II is more specific and informed by subject-specific features already determined. Overall, the MapSplice3 algorithm uses both approaches to resolve partial alignments in the second phase. Combined, this reduces the issue of false discovery of splice junctions due to short anchors while improving the sensitivity of splice site alignments.

2.4.1 Alignment to splicing context

Splice junctions derived from Phase I alignments form the basis of the splicing context. The accuracy of these splice junctions is derived from using segments at least 20bp in length on both donor and acceptor sites for their discovery. Junctions involving repetitive sequences are removed to reduce false positive splice junction reporting. For example, a splice junction where the sequence at the acceptor site has a high similarity with the downstream flanking sequence at the donor site (or a sequence at any of its alternative acceptor sites)

المنارة للاستشارات
www.manaraa.com

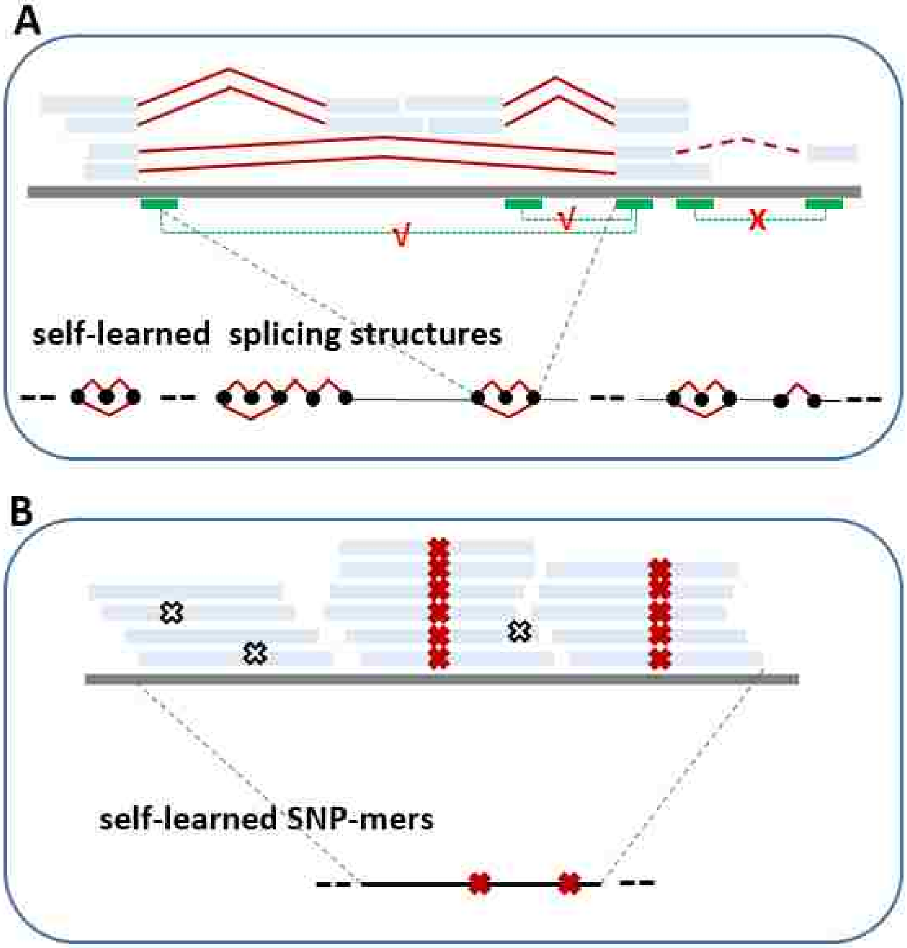


Figure 2.2: Examples to illustrate how splicing context and mutation context are derived. (A) An example of splicing context. Splice site sequence pairs represented with green bars are examined for junction consolidation. A splice junction will be excluded if two paired sequences are identical or similar to each other. (B) An example of mutation context. Two of the five mismatch positions are identified as candidate SNPs because they have high read coverage concordance.

المنارة للاستشارات
www.manaraa.com

would not be reported (Figure 2.2A). In Figure 2.2A, the green bars represent examples of sequence pairs to be examined for junction consolidation. The edit distance between the alternative acceptor site sequences is computed using the Smith-Waterman algorithm (REF). A splice junction will not be included in the splicing context if the edit distance is calculated to be very small. This filtering strategy is very effective at reducing false positive splice junction reporting. This is independent of read coverage and enables the retention of correct junctions even with low read coverage. In addition to the junctions derived from read alignments, splice junctions derived from transcript annotation can be used as input to augment the splicing context.

MapSplice3 allows the rapid interrogation of alternative splicing sites given the location of a partial alignment and the sequence of the unmapped segment. This is facilitated by using a two-level hashtable. The first level hash table supports splice site searching and the second level hashtable stores the information of paired alternative splicing sites at any individual site, either donor or acceptor. The key to access the second level hash table is a sequence segment (30 bases by default), which can uniquely identify the isoforms entering (or exiting) the splice site. The information stored includes support numbers, splicing signals, and features (splice junctions, indels, and mismatches) detected on the other end of the splice site and supporting alignments. Given the location of the existing partial alignment, donor or acceptor sequences connected to this location will be retrieved from the splicing hashtable and matched to the querying segment.

www.manaraa.com

2.4.2 Alignment to mutation context

Reads carrying SNPs relative to the reference genome reduce mapping efficiency and accuracy. This is especially true when the sequence variant is in proximity to the read end, a splice junction, or another mutation. As a consequence, the inability to align reads will cause an allelic bias, hindering downstream analysis, such as allele-specific expression. To alleviate this issue, SNP information is inferred and included as part of the context to facilitate alignment.

During the Phase I alignment, mismatched bases in all alignments are recorded. Resolving true SNPs from technical sequencing errors is a significant challenge. Only nucleotide variants supported by sufficient reads and consensus mismatches are retained as candidate SNPs. Figure 2.2B depicts an example of this event, five mismatches are shown, but only two positions illustrate high concordance identifying them as candidate variant alleles. To facilitate previously unaligned read segment mapping against genomic regions containing candidate SNPs, enhanced suffix array-based indices for SNP-mers are built. Each SNP-mer corresponds to a segment of genomic sequence carrying the alternative nucleotide positioned in the middle of the sequence. Thus, with the SNP-mer indices, segments containing variant alleles, like segment S5 in Figure 2.1, can be correctly mapped through the semi-maximal prefix searching method.

Partially aligned reads with unmapped segments are dealt with by mapping them to their neighboring region. For every single-end read, regions 300kbp downstream and upstream are considered. In the case of paired-end reads, the local region is further limited by the mapping location of its mate-pair, either downstream or upstream. Within identified

المنارة للاستشارات
www.manaraa.com

local regions of a read, the trailing segments are mapped to the reference genome with a pre-built local index using a similar approach to Phase I. The local index is built for the reference sequence of each genomic region in a fixed window size (3000K by default). A similar approach has been used in the HISAT program (REF), significantly improving the efficiency of short segment mapping position searching. This ab initio alignment strategy enables MapSplice3 to discover short overhang splice junctions which would otherwise be missed in Phase I due to the short length of the anchor segments and lead to ambiguous genome-wide mapping.

In the last step of the MapSplice3 algorithm, segments aligned in both phase I and II are assembled to generate the final alignments. Segments that cannot be aligned at this stage are soft-clipped, but included in the alignment files.

2.4.3 Pooled context from multiple datasets.

It is typical for RNA-seq experiments to include biological replicates in the sample set. Samples with the same or very similar biological contexts are likely to carry similar sets of splice junctions and genomic variants. Context derived from intermediate alignments not only have the potential to facilitate the mapping of reads from that single sample but also possibly improve the mapping of reads from biological replicates in the same experimental group. Splice junctions that cannot be resolved in one sample due to low coverage may be identified in other samples. Thus, borrowing contexts from biological replicates has the potential to facilitate the mapping of spliced reads that might otherwise be missed with independent analyses that are limited only to individual samples. MapSplice3 provides an option for processing multiple samples simultaneously with a pooled context (Figure

المنارة للاستشارات
www.manaraa.com

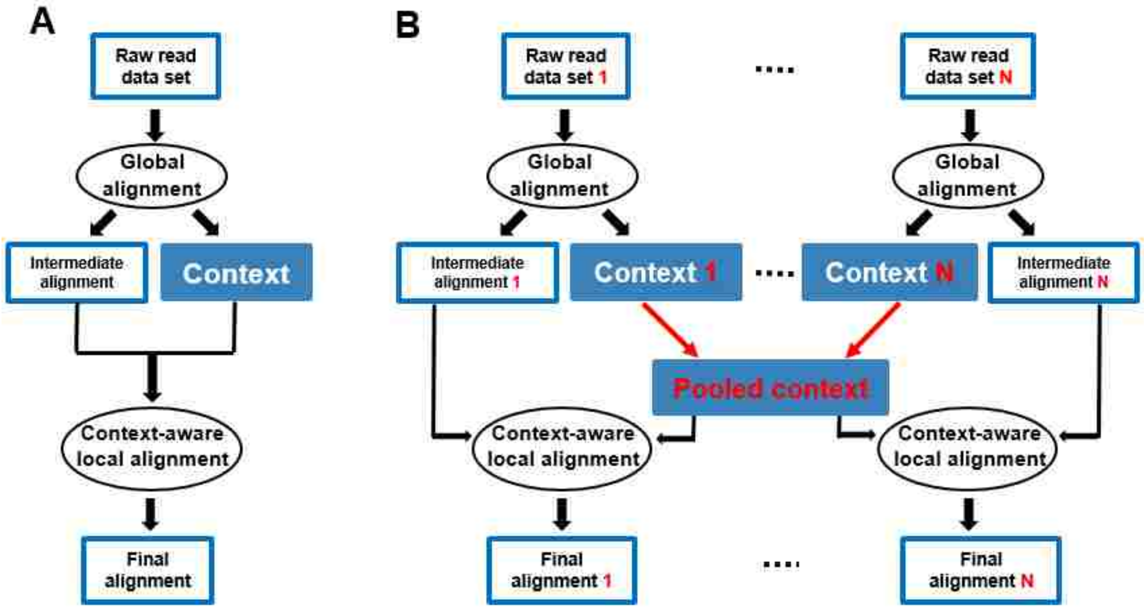


Figure 2.3: (A) The regular mode of MapSplice3 that utilizes unshared context: datasets are processed separately, and context derived from each dataset will only be used during the context-aware local alignment of reads from the same dataset. (B) The alternative mode of MapSplice3 that utilizes shared context: context from multiple datasets are pooled, and can be utilized for reads from any datasets.

2.3B). In this situation, each dataset is processed separately in the global alignment of phase I. However, context derived from intermediate alignments from each sample is then pooled and utilized for the remaining alignments in phase II.

2.5 Experimental results

2.5.1 Performance comparison on regular read alignment

Aligners, datasets, and metrics used in comparison

We performed a comprehensive evaluation of MapSplice3. A number of datasets including both synthetic and real datasets with different read lengths are used for the assessment. These datasets are detailed in Table 2.1. A variety of metrics to measure read mapping,

المنارة للاستشارات
www.manaraa.com

Table 2.1: Overview of simulated and real datasets.

Data	Total read #	Read length	Experiment
L100	20,000,000	100	Simulated
L200	20,000,000	200	Simulated
L300	20,000,000	300	Simulated
H100	20,000,000	100	Simulated
H200	20,000,000	200	Simulated
H300	20,000,000	300	Simulated
R101	190164184	avg=96	fetal lung fibroblasts
R262	7,050,382	avg=228	GM12878
10 Geuvadis datasets	324,900,974	avg=74	lymphoblastoid cell lines

Table 2.2: Overview of metrics for aligner performance assessment.

Metric		Description
Splice junction	Sensitivity	Proportion of detected true junctions among all true junctions in ground truth.
	Specificity	Proportion of detected true junctions among all detected junctions
	Coverage correlation	Pearson's correlation of junction read coverage between alignment and ground truth.
Mapping rate	Read	Proportion of mapped reads among all reads
	Base	Proportion of mapped bases among all bases
Allele coverage correlation	Reference	Pearson's correlation of allele read coverage between alignment and ground truth for reference allele at SNP coordinates.
	Alternate	Pearson's correlation of allele read coverage between alignment and ground truth for alternate allele at SNP coordinates.

splice junction discovery, as well as allelic biases are reported. The explanation of the metrics are listed in Table 2.2. We compare MapSplice3 against a number of the state-of-the-art aligners.

We next compare MapSplice3 using self-learned context with a number of State of the Art aligners. These include MapSplice2 (version 2.1.7), Tophat2 (version 2.0.12), HISAT (version 0.1.4-beta), STAR (version 2.4.0.1) and GSNAP (version 2014-10-22). The default settings were used for all those tools. For both STAR and HISAT, their 2-pass mode,

المنارة للاستشارات

www.manaraa.com

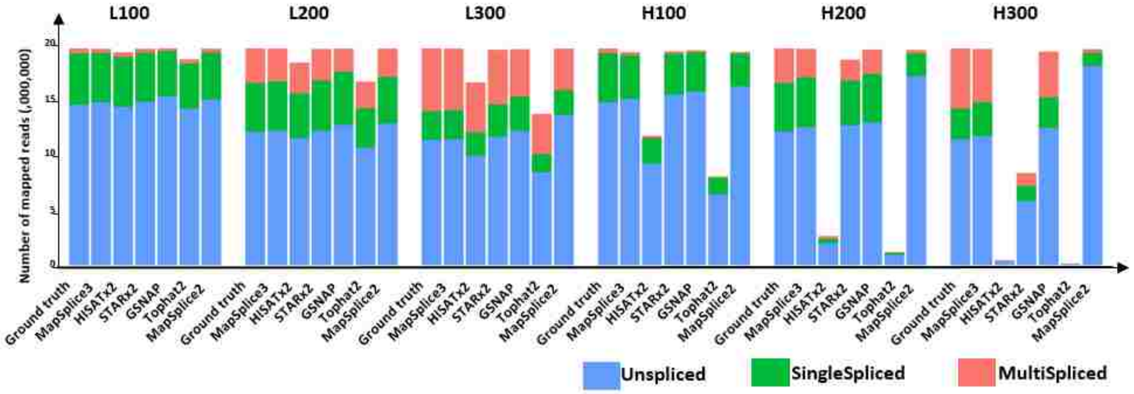


Figure 2.4: Read alignment yield in simulated datasets. Shown is the number of reads in ground truth and mapped by each aligner. Mapped reads are categorized according to the number of splice junctions detected. There is no splice junction in “Unspliced” reads. “SingleSpliced” and “MultiSplcied” reads contain one and more than one splice junctions respectively.

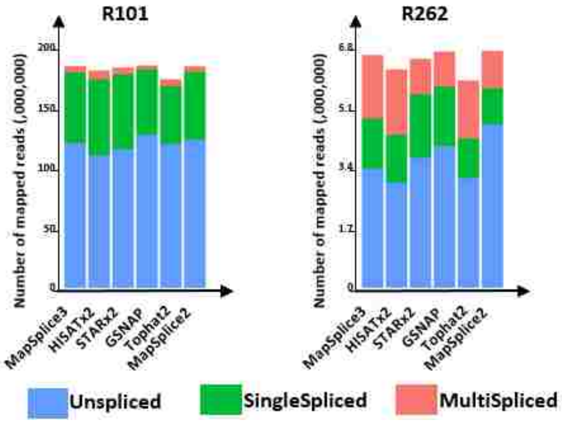


Figure 2.5: Read alignment yield in real datasets. Shown is the number of reads mapped by each aligner. Mapped reads are categorized according to the number of splice junctions detected. There is no splice junction in “Unspliced” reads. “SingleSpliced” and “Multi-Splcied” reads contain one and more than one splice junctions respectively.

namely, STAR 2-pass and HISATx2 are chosen for the comparison as they outperform their one-pass partner in accuracy and are more popularly used in real applications.

www.manaraa.com

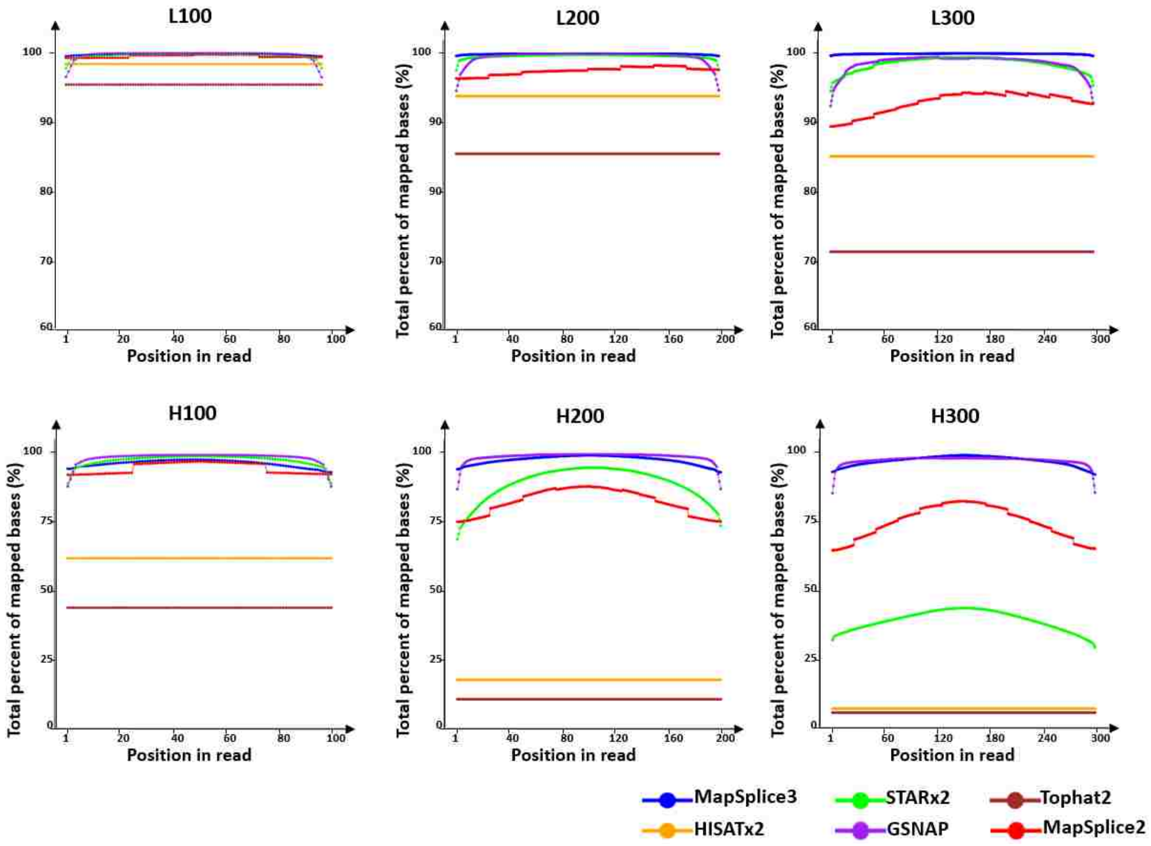


Figure 2.6: Distribution of base mapping rates along read sequence in simulated data. Total percentages of mapped bases for each base along the read sequence are counted and plotted for each aligner.

Alignment Yield

To evaluate the completeness of the alignment, we compute the read mapping rate, i.e., the proportion of the reads that are reported with an alignment to the reference genome regardless of the length of the alignment. Results are included in Table 2.7. In low error simulated data, MapSplice3 and GSNAP achieve a consistent high mapping rate over 99% in all three datasets with varying read length (L100, L200, and L300) followed by STARx2. Longer read length becomes problematic for HISAT and TopHat2, whose read mapping rates drop to around 70% with 300bp reads even in low error data. STARx2 shows conflicting results

المنارة للاستشارات
www.manaraa.com

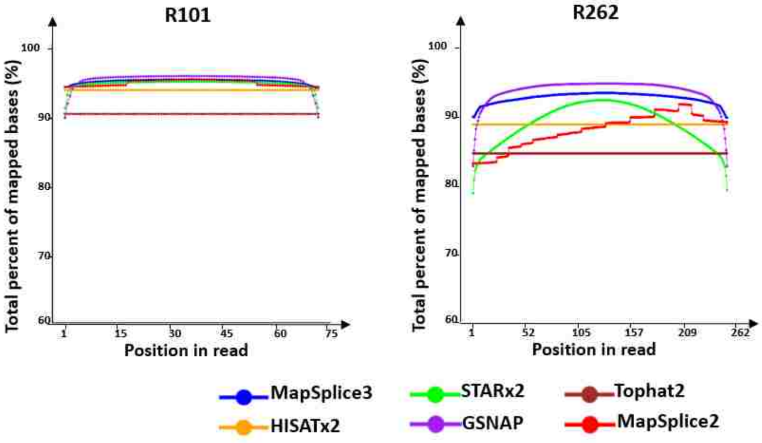


Figure 2.7: Distribution of base mapping rates along read sequence in real data. Total percentages of mapped bases for each base along the read sequence are counted and plotted for each aligner.

with higher mapping rate in 100bp and 300bp reads but lower mapping rate in 200bp reads. Further investigation shows that STARx2 adopts different mapping strategies for short and long reads for the trade-off of performance. Its result on 200bp may be improved if the long read alignment strategy is applied. The trend of the comparison on read and base mapping rates in high error simulated data (H100, H200, H300) are consistent with low error data with MapSplice3 and GSNAP being the most very sensitive in mapping. High error rates did reduce the alignment yields of HISATx2 and Tophat2 significantly to below 40% while others still maintain a mapping rate over 95%. GSNAP fares best in the mapping rates in two real datasets R101 and R262, followed closely by MapSplice3 and STARx2.

We also collect the base mapping rate, which is the proportion of total read bases that are mapped. The base mapping rate should be exactly the same as read mapping rate if every base of a read is mapped. However, only HISAT and Tophat2 strictly enforce that rule. Other aligners, such as MapSplice, GSNAP, and STAR, will report partial alignments of a read if deemed confident without requiring all bases mapped, resulting in a slightly

المنارة للاستشارات
www.manaraa.com

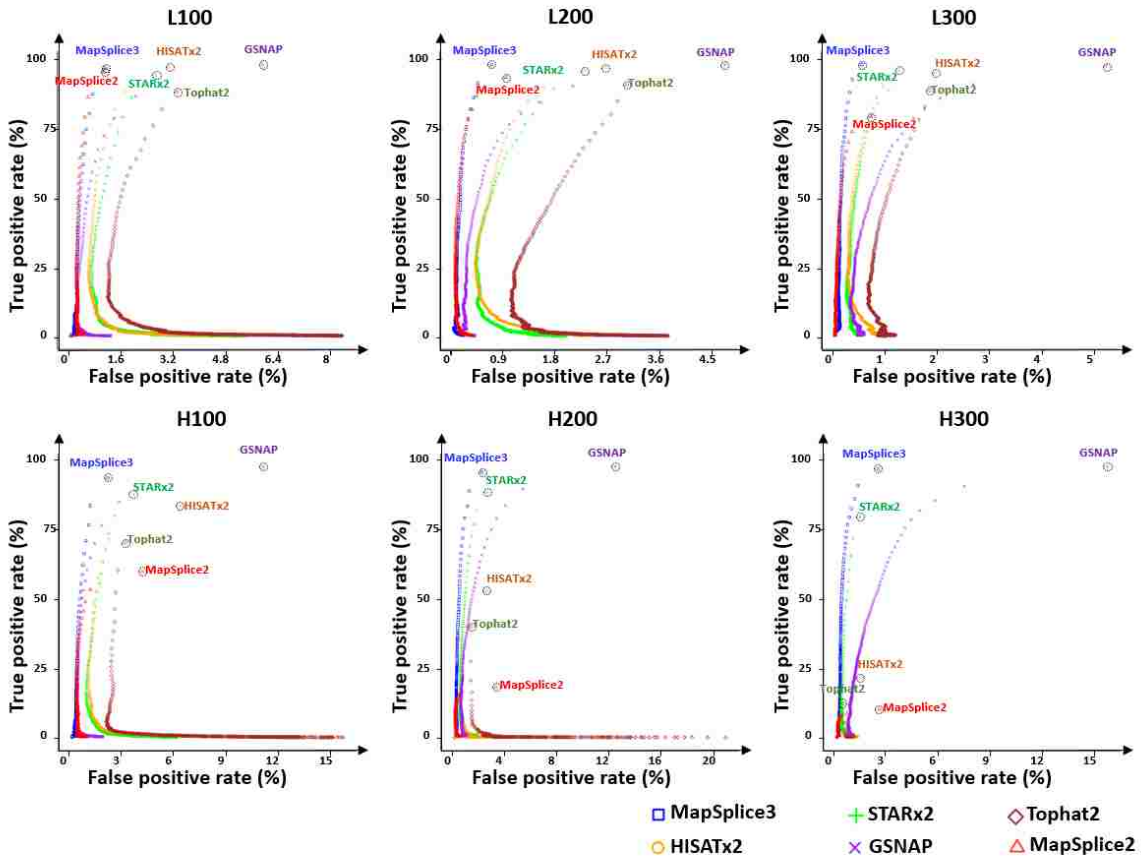


Figure 2.8: True positive rate (TPR) vs. false positive rate (FPR) (ROC-curve) for discovered splice junctions on simulated datasets. TPR is definded the same as sensitivity, which is the fraction of successfully detected true junctions among all true junctions. While FPR is defined as the fraction of falsely detected splice junctions among the total splice junctions.

lower base mapping rate than read mapping rate. The trimming of part of an alignment that cannot be mapped is called soft-clipping. The difference between base mapping rate and the read mapping rate indicates the extent of soft-clipping. For example, With L100 data, GSNAP shows higher read mapping rate than MapSplice3 but lower base mapping rate, demonstrating that more bases are soft-clipped in GSNAP than in MapSplice3.

To further understand that extent of soft-clipping, we plot the base mapping rates along each read position from the beginning to the end of the reads as shown in Figure 2.6 and

www.manaraa.com

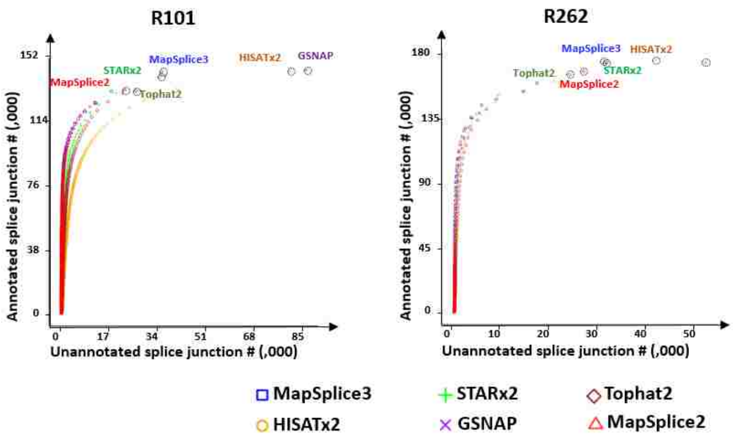


Figure 2.9: The number of annotated splice junctions vs. the number of unannotated splice junctions detected by each aligner on the real datasets.

Figure 2.7. In general, the higher the curve, the higher the mapping rate. MapSplice3 and GSNAP are consistently in the top pack outperforming other aligners in terms of base mapping rate across all datasets. The difference in the shape of the curves represents the difference in soft-clipping. HISAT and TOPHAT show a straight line along all read positions meaning no soft-clipping is conducted at any base. In contrast, other aligners show curves with different degree of drops towards the ends of the reads. These drops correspond to lower base mapping rate at these locations as a result of soft-clipping. GSNAP and STARx2 exhibit sharper drops than MapSplice3 in most of the datasets, with the degree of drops increases as the reads get longer. This is true especially for STARx2. This suggests that STAR and GSNAP fail to map many short sequences at the ends of reads, therefore, more aggressive soft-clipping has to be applied than that in MapSplice3. On the other hand, it also demonstrates that the context-based local alignment in MapSplice3 successfully salvages the alignment of short sequences which may be trimmed otherwise.

Completely forbidding soft-clipping as enforced by Tophat2 and HISATx2 does not

www.manaraa.com

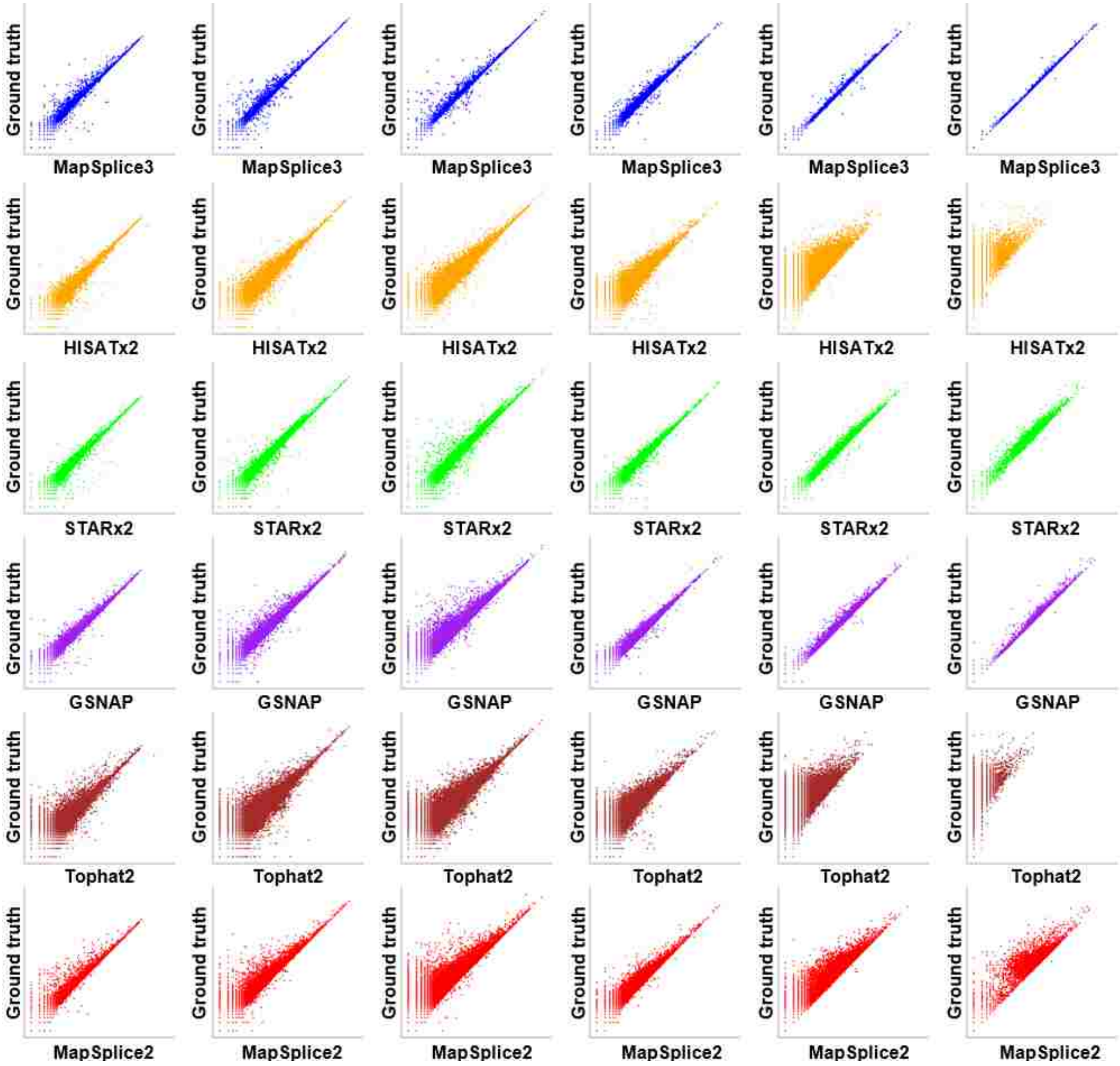


Figure 2.10: Correlation of splice junction supporting read numbers between aligners' results and ground truth in all the six simulated data sets. Each point in the scatter plot represents supporting numbers for one splice junction in aligners' results (X axis) and ground truth (Y axis).

improve the base mapping rate. On the contrary, it significantly reduces the read mapping rate since it does not report an alignment even if 99% of a read can be confidently mapped. The situation worsens as reads get longer or becomes noisier. The low mapping rate may potentially hamper the downstream gene expression analysis by causing low abundance or biases in gene expression.

المنارة للاستشارات
www.manaraa.com

We next examine specifically how well the aligner discovers spliced reads. To do this, we divided alignments into three categories: alignments that do not span any splice junction (unspliced), alignments that span only one splice junction (single-spliced) and alignments that span more than one splice junctions (multi-spliced). We then compare the distribution of alignments in each category against the ground truth. The results are shown in Figure 2.4 and Figure 2.5. As the read length gets longer from 100bp to 300bp, it is expected that the number of splice junction including one or more splice junctions will increase. This is shown in the stacked bar corresponding to the ground truth. However, different aligners show significant differences in the ability of mapping single-spliced and multi-spliced reads. The distribution of the three classes of alignments reported by MapSplice3 resembles most to ground truth, while other aligners reported relatively lower portions of single-spliced and multi-spliced junctions especially on the long read datasets with high error rates. Additionally, while GSNAP has the highest mapping rate overall, it reported the least amount of multi-spliced reads in all datasets.

High accuracy in splice junction discovery and coverage

An important utility of RNA-seq data is the detection of splice junctions which plays pivotal roles in detecting new genes or isoforms as well as quantifying alternative isoforms. Splice junctions can be derived directly from the spliced alignments.

To examine the performance of splice junction discovery, we compute the sensitivity, specificity and F-1 measure. The sensitivity of the splice junction discovery is the proportion of correct junctions identified through alignments out of all ground truth junctions. The specificity represents the proportion of false junctions out of all junctions identified

www.manaraa.com

in the alignments. F-1 measure measures which are the harmonic mean of sensitivity and specificity, are also calculated and shown to combine the two metrics. The result are shown in Table 2.7 GSNAP demonstrates the highest sensitivity with splice junction discovery in most of the datasets, closely followed by MapSplice3. However, its high sensitivity also comes with the worst specificity, meaning that it reports the most of false positive splice junctions in almost all datasets. In contrast, MapSplice3 significantly outperforms other aligners in the specificity in all datasets, achieving both high sensitivity and high specificity in all datasets, as indicated by the F-1 measure.

For real data, detected splice junctions are compared with gene annotations, only those match splice junctions in annotation within 5bps are considered as annotated ones. In R101, compared with MapSplice3, only GSNAP and HISATx2 reported larger numbers of splice junctions which have been annotated in Ensembl. However, the numbers of unannotated splice junctions for GSNAP and HISATx2 almost doubles that of MapSplice3. In R262, as mentioned several times before for the advantage of MapSplice3 on mapping longer reads, MapSplice3 discovered the largest number of annotated junctions with the highest precision.

One common procedure to filter out spurious splice junctions is to apply a threshold on the minimum number of reads supporting a splice junction. A splice junction will be filtered if its total supporting reads fall below the threshold. Given a list of splice junctions, we may calculate sensitivity and specificity each time we raise the support threshold, allowing us to examine the tradeoff between sensitivity and specificity through ROC curves. Figure 2.8 includes the ROC curves for each aligner. The ROC curves illustrate the true/-false positive rates of splice junction detection compared with ground truth given different

المنارة للاستشارات
www.manaraa.com

thresholds of splice junction supporting read numbers in simulated datasets. Each point in the ROC curves was computed with the detection (discrimination) threshold given by the number of reads mapped across each junction, i.e. for each aligner only junctions supported by at least N reads were selected for each point along the ROC curves, with N varied from 1 (the lowest threshold) to 1000 (the highest threshold). With the curve closer to the left border, the aligner is considered to have higher accuracy in splice junction detection. And with the curve closer to the top border, the aligner is considered to be more sensitivity in mapping splice junctions. The curve corresponding to MapSplice3 consistently sits closest to the top left corner of the figure, showing its superior sensitivity and precision in splice junction discovery in varying read length and in both low and high error datasets.

Besides the identity of splice junctions, the accuracy of splice junction coverage, which is the number of spliced reads spanning a splice junction, is also important for downstream analysis. For example, spliced reads are critical in parsing out both identify and quantity of alternative isoforms. The lack of coverage of spliced reads may lead to lower estimated abundance. To evaluate the accuracy of splice junction coverage, we drew scatter plots between ground truth coverage and coverage derived from the ground truth for all correctly discovered junctions. Figure 2.10 includes the coverage scatter plots for each aligner on each simulated dataset. The closer the points to the diagonal the more accurate the coverage. Points above the diagonal represent deficiencies in read coverage when some spliced reads are failed to be mapped. Points below the diagonal represent more read coverage than the ground truth which indicates false spliced alignments are reported. The spread of the points represents that none of the aligners are perfect in obtaining all the spliced alignments. However, in comparison, the points shown in the scatterplots of MapSplice3

المنارة للاستشارات
www.manaraa.com

are much more closely packed towards the diagonals than other aligners, demonstrating the highest overall correlation to the ground truth. Unlike other aligners, where the spread of the points becomes looser as the reads get longer, MapSplice3 shows an opposite trend with a tighter spread, showing its capability in recovering spliced long reads.

Accurate Alignments Alleviate Bias in Allelic Ratio

When mapping reads to the reference genome, reads carrying alternative bases at SNP coordinates would have an extra mismatch compared to those with reference nucleotide. In some cases, this extra mismatch would prohibit aligners from mapping the read correctly, sometimes leading to a partial alignment or completely incorrect alignment. Consequently, the percentage of the alternative alleles at a SNP location may be biased. To assess each of the aligners in their capabilities in reducing the allelic bias on a genomic scale, we counted the read coverage of reference and alternative nucleotides at each SNP position and calculated the Pearson Correlation relative to the ground truth from simulated datasets. In general, when working with low error datasets, all the aligners achieved relatively high correlations (higher than 90%) for either reference or alternative nucleotides, except for reference allele coverage reported by Tophat2. However, when error rate becomes higher, along with less mapped reads and bases, HISATx2, STARx2, Tophat2, and GSNAP all exhibited low correlations in at least one dataset. Only MapSplice3 and GSNAP maintained high correlations (higher than 90%) with ground truth in all the three datasets for both of the types of alleles, which is consistent with the high mapping rates observed from their performance in high error datasets.

المنارة للاستشارات

www.manaraa.com

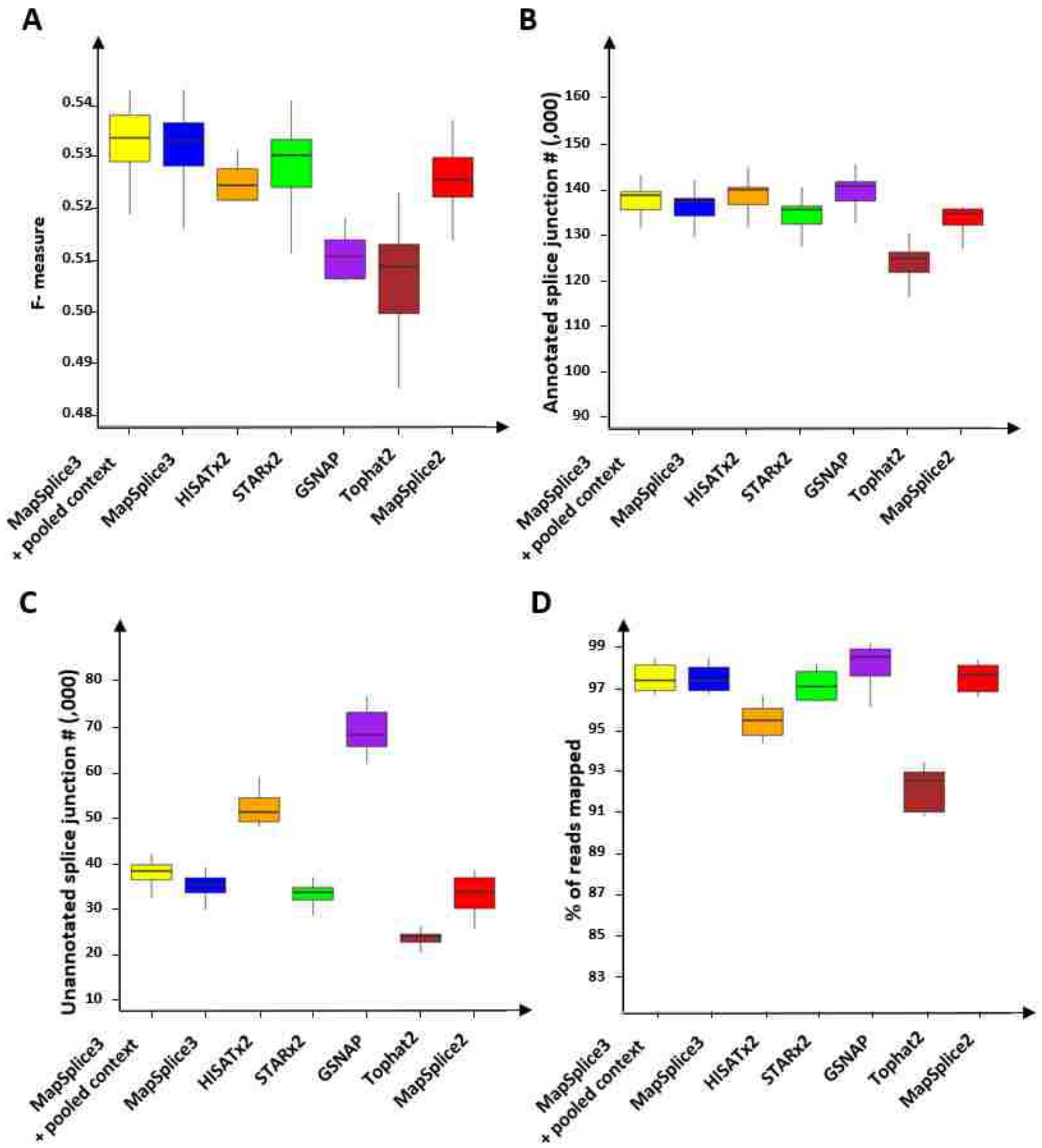


Figure 2.11: Summary of splice junction discovery and read mapping rates for aligners with 10 datasets from Geuvadis RNA sequencing project. (A) (B) Number of annotated splice junctions detected by each aligner on the 10 datasets. (C) Number of unannotated splice junctions detected by each aligner on the 10 datasets. (D) Read mapping rates for each aligner on the 10 datasets.

Performance improvement achieved with context

We first investigated the effectiveness of the context-aware alignment. Alignments were run on the simulated datasets under three configurations: Phase I only alignment without

المنارة للاستشارات
www.manaraa.com

any context, 2-Phase alignment with self-learned context and 2-Phase alignment with perfect context. The perfect context was obtained directly from the ground truth data. The summary of MapSplice3 performance under different configurations were reported in Table 2.6.

In general, leveraging context in terms of splice junctions and SNPs significantly improves the alignment performance over the one where context is not used. The improvements are across the board, including more accurate splice junction detection, higher read and base mapping rates as well as alleviated allelic biases at SNP locations. The biggest improvements lie in the relatively difficult cases when the reads are relatively short or the error rate of the reads are high. For example, for 100bp reads, MapSplice3 gains around 4% (L100) and 8% (H100) in splice junction prediction sensitivity, and around 2% and 8% in base mapping rate with self-learned context. In this case, context derived from a collection of reads may provide extra knowledge to complete the alignment of individual reads/bases that cannot be mapped otherwise. One important aspect that context helps solve is the misalignment of “short anchor ” specific to spliced reads. To detect splice junctions and map read sequences across them confidently, a sufficiently long (12bp) and error-free sequence anchoring on either side of the splice junction is always required. But in shorter and higher error rate reads, more anchor sequences are short or imperfect. Self-learned context will lower the requirement to align these sequences by guiding the alignments towards the specific context while completing the alignments. Another improvement is the reduction of mapping bias between the reference and the alternative allele at the SNP location, evidenced by the higher correlation of coverage between the alignment and the ground truth in both reference and alternate alleles.

www.manaraa.com

Alignments using perfect context performed the best but perfect context are not available in real applications. However, results using self-learned are very close to that using perfect context, suggesting that the self-learned context would provide high-quality alignment provided sufficient amount of reads. Another observation from this comparison is that even with perfect context including all the splice junctions and SNP locations with a dataset, the alignments are not perfect, which reflects the nature of potential ambiguity in read alignment besides possible improvement to be incorporated including other context like small indels.

Pooled context improves splice junction discovery sensitivity

We additionally performed an experiment on a larger scale with ten samples from Geuvadis RNA sequencing project. Besides running the other five tools and MapSplice3 under the regular mode, we also applied MapSplice3 with pooled context as described in Figure 2.3B. Splice junction discovery results and read mapping rates were reported in Figure 2.11. For splice junction discovery results, together with numbers of detected annotated and unannotated splice junctions, F-measure was also calculated and shown to combine the two metrics. Specifically, MapSplice3, as well as HISATx2 and GSNAP, detected the most annotated splice junctions. But at the same time, MapSplice3 reported much less unannotated splice junctions, regardless of using pooled context or separate context. And all the aligners, except for HISATx2 and Tophat2, exhibited strong mapping capability, with achieving relatively high read mapping rates higher than 95%. Compared to the regular mode of MapSplice3, with pooled context, its overall performance was improved with having more reads mapped, achieving higher F-measure, and detecting more annotated

المنارة للاستشارات
www.manaraa.com

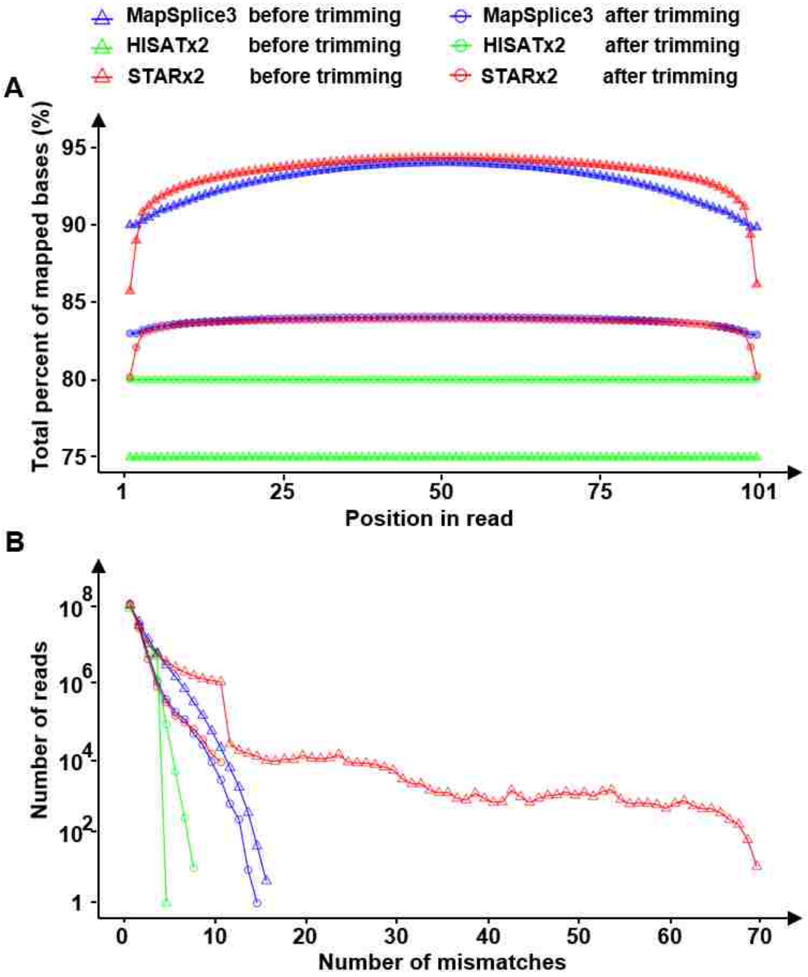


Figure 2.12: (A) Base mapping rates along the read sequence reported by each aligner in both untrimmed data and trimmed data. (B) Distribution of total mismatches in alignments.

splice junctions, though slightly more unannotated splice junctions were also reported.

Soft-clipping enables MapSplice3 on handling untrimmed data

The first step in processing raw reads is to trim the adapter sequences and low quality bases. One popular software to do this is Trimmomatic [Bolger et al., 2014], which removes the detected bases from adapter sequences as well as those with low qualities (Q score lower than 5). This is often called hard-clipping. While the trimmed data would be of high quality, sometimes, it may throw out too much data. For example, in the R101 data, about

المنارة للاستشارات
www.manaraa.com

87% of total reads survived the trimming process. In the meantime, a lot of paired reads become unpaired as one of the paired reads is contaminated with adapters. Removing a significant amount of reads may bias the library size, leading to inaccurate downstream analysis. In this experiment, we would like to investigate whether aligners may retain more reads by soft-clipping the adapter sequences during the alignment process. We apply the aligners MapSplice3, and STARx2 to R101 data both before and after adapter trimming. For comparison, we also apply HISATx2 to the same data to examine the results without soft-clipping.

Figure 2.12A presents the base yield results while applying each aligner to either dataset. In general, MapSplice3, and STARx2 exhibit significantly higher base yield with raw reads (10%) than with trimmed data. In contrast, HISATx2 generates 75% alignments with raw reads, even lower than trimmed reads. This is because HISATx2 will not align reads that are partially contaminated with adapters. Adapter trimming step is crucial here to remove this adapter sequences to facilitate mapping by HISATx2.

To evaluate the effectiveness of soft-clipping in removing adapter sequences, we try to search for adapter sequences in the alignment of raw reads by applying Trimmomatic again on alignments by MapSplice3 and STARx2. Only 0.0006% bases were identified as from adapters in MapSplice3's reported alignments. This suggests that soft-clipping is effective in automatically removing adapters.

To further examine the quality of the salvaged alignments when applying MapSplice3 and STARx2 to the raw reads, we plotted the distribution of total mismatches in alignments of these aligners in both trimmed and untrimmed data as shown in Figure 2.12B. Interestingly, every aligner has a very different upper bound of total mismatches. With

المنارة للاستشارات
www.manaraa.com

untrimmed data, STARx2 reports alignments with mismatches up to 70. There is 1.22% total alignment (around 4.2 million) with mismatches greater than 5. A closer look at these aligned reads suggests that many of the mismatches are in the low quality bases, meaning STARx2s soft-clipping is not effective in removing bad bases. Although the inclusion of such mismatches does not significantly decrease quality score of the alignment, the confidence of these alignments is low, making them questionable for downstream analysis. In contrast, MapSplice3 shows the most consistent trend of mismatch distribution of the alignments before and after trimming. HISATx2 exhibits fewer mismatches than STARx2 as its “end-to-end” alignment restriction naturally declines those alignments with adapters and low quality bases. However, the errors in the untrimmed data did increase slightly.

Conclusively, the soft-clipping mechanism adopted by MapSplice3 not only enables itself to report more alignments and mapped bases with untrimmed data (compared to HISATx2 and its results on trimmed data), but also avoids spurious alignments caused by adapters and sequencing errors efficiently (compared to STARx2). Additionally, the results of soft-clipping are consistent with the hard-clipping applied by trimmomic in removing adapters. However, it successfully salvages about 10% reads with confident alignments that would be removed due to insufficient length by trimming.

2.5.2 Fusion transcript prediction results

In addition to splice junction detection and spliced read alignment, MapSplice3 can also be used for fusion transcript discovery. To assess the fusion detection capacity of MapSplice3, we applied it to both synthetic and real datasets that were used in a previously published study [Liu et al., 2015] to compare a list of the State of the Art fusion detection software.

المنارة للاستشارات
www.manaraa.com

Table 2.3: Fusion detection results on synthetic datasets (5X, 20X, 50X). Symbol * marks the top 3 tools in terms of F_1 score.

	5X			20X			50X		
Tools	True	False	F_1	True	False	F_1	True	False	F_1
MapSplice3	66	0	0.61*	133	3	0.93*	135	3	0.94*
SOAPfuse	116	9	0.84*	136	18	0.89*	137	22	0.89*
FusionCatcher	96	13	0.74*	108	17	0.79*	108	19	0.78*
JAFFA	43	2	0.44	83	13	0.67	88	14	0.70
EricScript	67	9	0.59	104	12	0.78	107	13	0.78
chimerascan	63	10	0.57	102	22	0.74	104	25	0.75
PRADA	17	0	0.20	50	1	0.50	56	2	0.54
deFuse	16	1	0.19	69	7	0.61	52	5	0.50
FusionMap	31	0	0.34	73	7	0.63	86	13	0.69
Tophat-Fusion	29	7	0.31	46	11	0.44	53	14	0.49
MapSplice	21	3	0.24	40	7	0.41	42	10	0.42
BreakFusion	42	14	0.41	82	33	0.62	100	39	0.69
SnowShoes-FTD	2	1	0.03	3	1	0.04	3	1	0.04
FusionQ	47	7	0.46	53	11	0.50	64	18	0.55

Table 2.4: Fusion detection results on synthetic datasets (100X, 200X). Symbol * marks the top 3 tools in terms of F_1 score.

	100X			200X		
Tools	True	False	F_1	True	False	F_1
MapSplice3	135	5	0.93*	137	5	0.94*
SOAPfuse	139	26	0.88*	138	23	0.89*
FusionCatcher	108	20	0.78*	109	21	0.78*
JAFFA	88	16	0.69	88	16	0.69
EricScript	104	13	0.78*	106	13	0.78*
chimerascan	105	30	0.74	105	30	0.74
PRADA	57	2	0.55	57	3	0.54
deFuse	73	9	0.63	104	13	0.78*
FusionMap	92	27	0.68	98	50	0.66
Tophat-Fusion	53	14	0.49	56	15	0.51
MapSplice	51	8	0.49	51	9	0.49
BreakFusion	106	44	0.71	107	47	0.70
SnowShoes-FTD	3	1	0.04	3	1	0.04
FusionQ	81	18	0.65	45	8	0.44

They have generated five synthetic datasets with different fusion coverages (5X, 20X, 50X, 100X, 200X) were used. The reads in those data sets are all 100 base pairs long and simulated from the same set of 150 synthetic fusion transcripts. The real datasets are generated from 4 breast cancer cell lines (BT-474, SK-BR-3, KPL-4, and MCF-7) and

المنارة للاستشارات

www.manaraa.com

Table 2.5: Fusion detection results on real datasets. Symbol * marks the top three tools in terms of F_1 score.

Tools	Breast cancer Validated	Invalidated	F_1	Melanoma Validated	Invalidated	F_1
MapSplice3	16	19	0.173	7	9	0.084*
SOAPfuse	20	48	0.183*	10	98	0.078*
FusionCatcher	19	48	0.175	3	6	0.026
JAFFA	16	16	0.176*	2	2	0.026
EricScript	16	67	0.137	3	67	0.027
chimerascan	19	96	0.143	5	189	0.029
PRADA	15	22	0.160	3	4	0.038
deFuse	19	116	0.133	10	189	0.057
FusionMap	6	126	0.043	2	85	0.017
Tophat-Fusion	15	58	0.135	4	25	0.045
MapSplice	16	37	0.158	5	39	0.052
BreakFusion	15	1923	0.014	6	3092	0.004
SnowShoes-FTD	15	5	0.176*	4	1	0.052
FusionQ	4	453	0.013			
FusionHunter	13	10	0.150	4	4	0.051
ShortFuse	19	24	0.197*	7	30	0.075*

6 melanoma samples (M980409, M010403, M000216, M000921, M990802, and 501Mel). 27 and 11 previously experimentally validated fusions are used as the underlying truth for the evaluation.

We compare the fusion detection results by MapSplice3 to the fusion detection results by other software [Liu et al., 2015]. The results are included in Table 2.4 (synthetic datasets) and Table 2.5 (real datasets). Only SoapFuse are among the top three tools in all the 5 simulated datasets and 2 real datasets. And MapSplice3 outperforms other tools in all of the 5 simulated datasets and 1 real dataset. Besides, FusionCatcher and ShortFuse also show competitive capacity in simulated data and real data respectively.

المنارة للاستشارات
www.manaraa.com

2.5.3 Circular RNA detection results

To evaluate circular RNA detection capacity of our algorithm and compare it with other tools, we ran MapSplice3 and STAR on four real datasets (with accession numbers SRR1636985, SRR1636986, SRR1637089 and SRR1637090) from a previous study [Gao et al., 2015], and compared their results with that reported by CIRI in that study. These sequencing data sets are generated from HeLa cells based on ribominus RNA sequencing with or without RNase R treatment. Each dataset contains 13 to 43 million pairs of 101 bp reads. And 24 circular RNAs predicted by CIRI were experimentally validated in the same study. In our run, MapSplice3 successfully recovered all of them, while STAR detected 21. Also, the numbers of spanning reads identified by each tool for each back splicing events were collected and plotted in Figure 2.13. It reveals that MapSplice3 is a highly sensitive tool on circular rna detection as it not only successfully detected all validated back splicing events but also assigned relatively more reads spanning them.

2.5.4 Running time and memory requirements

Speed benchmarks were performed with the servers from Lipscomb High Performance Computing in the University of Kentucky. And the servers are equipped with Dual Intel Xeon CPUs E5-26708@2.60GHz and 64GB of 1600MHz RAM. The mapping speed and memory usage for each aligner are presented in Figure 2.14. In general, HISATx2 achieved the highest speed, followed by STARx2. MapSplice3 is not as fast as HISATx2

www.manaraa.com

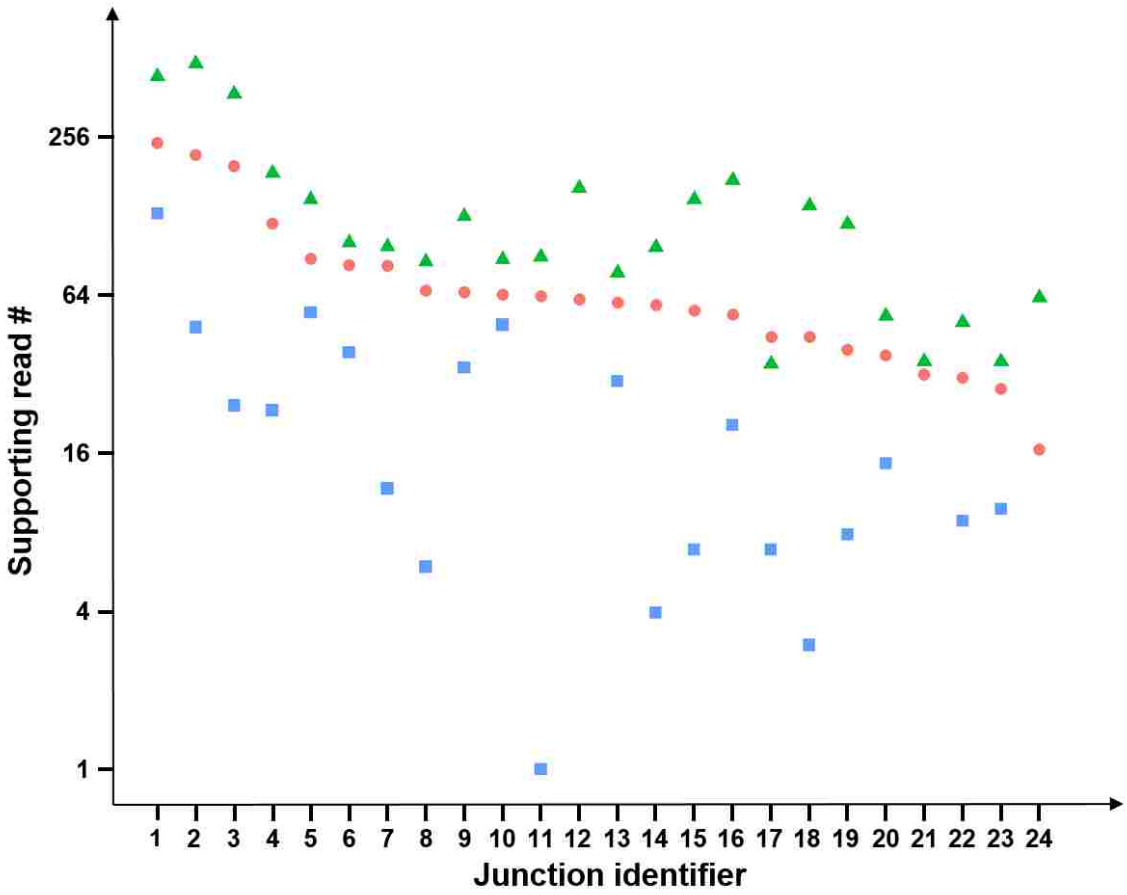
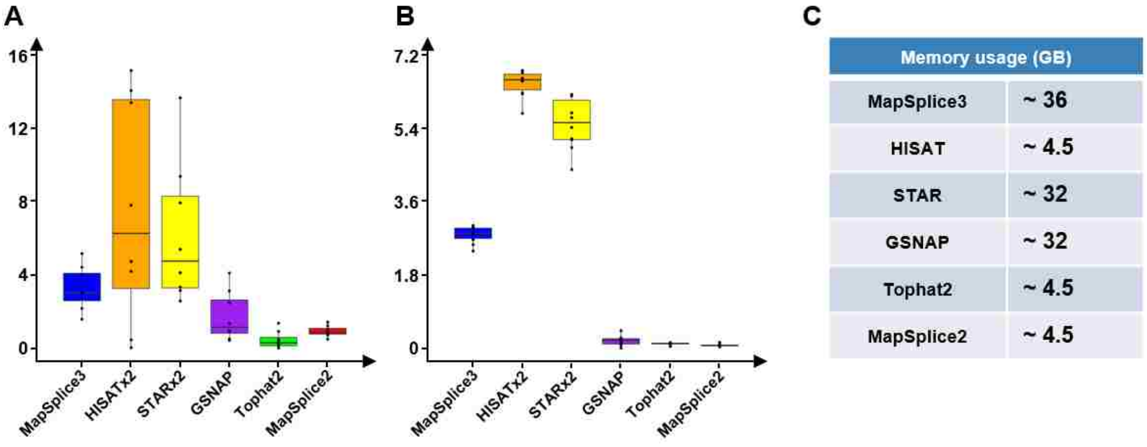


Figure 2.13: Number of supporting reads identified by different tools for each back-splice junction. Both MapSplice3 and CIRI discovered all the 24 experimentally validated back splice junctions, while STAR missed 3 of them (the 12th, 21th, and 24th in the figure).

and STARx2, but is in the same pack and greatly outperformed GSNAP, Tophat2, and MapSplice2. And interestingly, as shown in Figure 2.14A, aligners especially HISATx2 and STARx2 perform differently on datasets with different read lengths and error rates. MapSplice3 requires a relatively high memory of 36 GB for mapping reads to the human genome. The requirement would not increase with the size of data sets, as MapSplice3 only loads the suffix array index of the reference genome and a fixed size of reads into memory. And as the RAM price keeps dropping and high performance computing resources are commonly used in the large-scale genomic data research today, the memory requirements

المنارة للاستشارات
www.manaraa.com



Memory usage (GB)	
MapSplice3	~ 36
HISAT	~ 4.5
STAR	~ 32
GSNAP	~ 32
Tophat2	~ 4.5
MapSplice2	~ 4.5

Figure 2.14: Million bases mapped per second by each aligner on (A) simulated datasets, R101, R262, and (B) ten datasets from Geuvadis RNA sequencing project. (C) Memory requirements of spliced alignment software.

are not that expensive than before.

2.6 Conclusion

RNA-seq plays a more and more prominent role in profiling the transcriptome. In a typical RNA-seq analysis pipeline, the first and fundamental step is to correctly obtain the alignment of RNA-seq read onto the reference genome.

In spite of the many aligners that have been published, and those in development, this problem is far from being solved. Many significant improvements have been achieved compared to first-generation tools. The difficulty of correct alignment to a reference genome comprises several aspects. One such issue is the complexity of reference genome, especially where there are abundant repetitive sequences. In terms of size, long repetitive sequences can mislead whole reads to incorrect positions. Short repetitive sequences may cause the misalignment of parts of the read, which leads to false splice junctions reporting. The large number of genomic variants and possible sequencing errors in RNA-seq makes

www.manaraa.com

it challenging to identify unique and correct alignments for reads. Identical repeats cause ambiguity in read mapping. Repeats with sequence similarity can lead to incorrect, but unique placements of the whole or a part of a read, as aligners map reads with the least edit-distances. However, this is not always the correct mapping position. Another difficulty arises from evolving sequencing technologies. The length of reads generated continues to increase. Longer reads can significantly reduce the ambiguity of mapping caused by repetitive sequences in a reference genome. However, with longer read sequences, more biological features, such as single nucleotide variants, indels, and splice junctions, are covered. Additionally, longer reads are coincident with more sequencing errors. Currently, most aligners are designed and optimized for short reads (no longer than 100bp) and are ill-equipped to deal with longer reads. An unmet need is a tool that can cope with the difficulties of longer reads. MapSplice3 adopts several strategies to deal with these problems and, as such, bridges this gap. To demonstrate MapSplice3's ability to align RNA-seq reads with varying lengths and error profiles, we used six simulated datasets and three real data sets, and compared the alignment results of MapSplice3 with five popular tools. In these comparisons, different metrics were used to evaluate the aligners' performance. These output measures included read alignment yield, base yield, coverage of splice junctions, and the sensitivity and specificity of splice junction discovery. MapSplice3 demonstrated superiority over other tools across these aspects and outperformed other tools on the most challenging datasets that have long reads and high error rates. Additionally, using datasets containing gene fusions and circular RNAs, we demonstrated that MapSplice3 could efficiently align reads from those irregular transcripts. In those experiments, compared to other tools, MapSplice3 exhibits competitive accuracy and sensitivity in detecting both

المنارة للاستشارات

www.manaraa.com

Table 2.6: Summary of splice junction discovery performance, read / base mapping rates, and read coverage correlation at SNPs in simulated datasets reported by MapSplice3 with three different configurations.

		Mapping rate				Junction				Allel Corr	
Data	Tools	Total Read	Unique Read	Correct Read	Total Base	Sensi	Speci	F_1	Cov. Corr	Refer	Alter
L100	None	99.18	98.24	97.37	97.70	92.63	99.25	95.28	.9682	.9723	.9863
	Learned	99.88	98.82	98.24	99.75	96.35	98.94	97.63	.9976	.9767	.9982
	Perfect	99.88	98.99	98.44	99.77	98.30	99.08	98.69	.9978	.9806	.9995
L200	None	99.75	99.37	98.71	98.88	97.03	99.36	98.18	.9834	.9700	.9970
	Learned	99.94	99.50	99.23	99.84	98.20	99.34	98.77	.9986	.9744	.9997
	Perfect	99.94	99.56	99.29	99.86	98.84	99.40	99.12	.9986	.9775	.9999
L300	None	99.82	99.69	99.17	99.13	97.89	99.45	98.66	.9863	.9973	.9839
	Learned	99.97	99.80	99.59	99.87	98.58	99.45	99.01	.9988	.9990	.9853
	Perfect	99.97	99.87	99.64	99.88	98.97	99.53	99.25	.9988	.9974	.9866
H100	None	94.16	93.34	92.37	87.89	84.04	98.97	90.90	.9495	.9365	.9447
	Learned	98.36	97.43	96.14	95.65	91.82	97.91	94.77	.9961	.9518	.9709
	Perfect	99.12	98.68	97.54	96.95	95.94	98.14	97.03	.9959	.9810	.9995
H200	None	99.38	98.35	97.52	93.73	94.03	98.47	96.20	.9907	.9804	.9745
	Learned	99.84	98.83	98.04	97.00	95.88	97.82	96.84	.9987	.9892	.9895
	Perfect	99.88	98.83	98.04	97.71	97.21	98.07	97.64	.9988	.9946	.9995
H300	None	99.69	99.55	98.68	94.60	95.44	97.95	96.68	.9950	.9880	.9820
	Learned	99.76	99.59	98.96	96.66	96.57	97.36	96.96	.9990	.9943	.9881
	Perfect	99.76	99.60	98.99	97.15	97.45	97.69	97.57	.9990	.9706	.9964

gene fusion and circular RNAs.

www.manaraa.com

Table 2.7: Summary of splice junction discovery performance, read / base mapping rates, and read coverage correlation at SNPs in simulated datasets reported by MapSplice3 with three different configurations.

		Mapping rate				Junction				Allel Corr	
Data	Tools	Total Read	Unique Read	Correct Read	Total Base	Sensi	Speci	F_1	Cov. Corr.	Refer	Alter
L100	MapSplice3	99.88	98.82	98.24	99.75	96.35	98.94	97.63	.9976	.9767	.9982
	HISATx2	98.26	96.14	95.98	98.26	96.80	96.93	96.86	.9933	.9429	.9894
	STARx2	99.83	98.03	97.89	99.57	93.83	97.34	95.55	.9958	.9296	.9874
	GSNAP	99.98	98.60	98.25	99.70	97.54	94.00	95.74	.9908	.9758	.9944
	Tophat2	95.02	91.53	90.79	95.02	87.84	96.67	92.04	.9819	.7420	.9427
	MapSplice2	99.64	98.94	98.18	99.48	94.92	98.98	96.91	.9942	.9836	.9789
L200	MapSplice3	99.94	99.50	99.23	99.84	98.20	99.34	98.77	.9986	.9744	.9997
	HISATx2	93.44	92.20	92.13	93.44	97.13	97.46	97.29	.9881	.9603	.9814
	STARx2	99.76	98.44	98.32	99.51	96.14	97.81	96.97	.9874	.9753	.9990
	GSNAP	99.93	99.39	98.64	99.40	98.28	95.51	96.88	.9900	.9646	.9983
	Tophat2	84.68	82.45	82.01	84.68	90.98	97.11	93.95	.9719	.8923	.9786
	MapSplice2	99.55	99.67	98.58	97.30	93.63	99.10	96.29	.9855	.9863	.9939
L300	MapSplice3	99.97	99.80	99.59	99.87	98.58	99.45	99.01	.9988	.9990	.9853
	HISATx2	84.31	83.52	83.47	84.31	95.55	97.93	96.73	.9795	.9255	.9710
	STARx2	99.40	99.29	98.98	98.35	96.62	98.68	97.64	.9906	.9831	.9726
	GSNAP	99.69	99.46	98.35	98.47	97.99	94.40	96.16	.9825	.9961	.9828
	Tophat2	69.79	68.59	68.45	69.79	89.27	98.05	93.45	.9582	.7679	.9570
	MapSplice2	99.33	99.82	98.66	92.47	79.80	99.26	88.47	.9577	.9465	.9483
H100	MapSplice3	98.36	97.43	96.14	95.65	91.82	97.91	94.77	.9961	.9518	.9709
	HISATx2	59.66	58.32	58.14	59.66	81.89	93.70	87.40	.9481	.9058	.8418
	STARx2	98.68	96.43	96.14	96.88	86.12	96.40	90.97	.9928	.8969	.9921
	GSNAP	99.14	97.77	97.33	97.82	95.75	88.80	92.14	.9892	.9425	.9962
	Tophat2	40.90	39.66	39.49	40.90	68.89	96.85	80.51	.9100	.6873	.8389
	MapSplice2	98.40	97.64	96.66	93.92	58.77	95.89	72.88	.9791	.8960	.9729
H200	MapSplice3	99.84	98.83	98.04	97.00	95.88	97.82	96.84	.9987	.9892	.9895
	HISATx2	13.49	13.25	13.21	13.49	53.10	97.50	68.75	.7781	.7807	.5734
	STARx2	94.78	89.00	88.65	88.04	88.61	97.45	92.82	.9908	.6509	.9227
	GSNAP	99.44	97.96	97.20	98.21	97.82	87.30	92.26	.9939	.9464	.9913
	Tophat2	6.17	6.04	6.02	6.17	40.03	98.69	56.96	.6945	.6741	.5304
	MapSplice2	98.02	98.66	97.3	81.35	18.14	96.72	30.55	.9226	.9352	.9671
H300	MapSplice3	99.76	99.59	98.96	96.66	96.57	97.36	96.96	.9990	.9943	.9881
	HISATx2	2.25	2.23	2.23	2.25	21.53	98.49	35.34	.5749	.7436	.4664
	STARx2	42.64	42.6	42.49	35.83	79.37	98.54	87.92	.9701	.8483	.8350
	GSNAP	98.48	98.17	96.6	96.79	97.38	82.84	89.52	.9885	.9627	.9603
	Tophat2	0.76	0.76	0.75	0.76	12.15	99.61	21.66	.8605	.6831	.4291
	MapSplice2	98.46	99.53	97.9	73.31	9.99	97.31	18.12	.8521	.9082	.8590

المنارة للاستشارات
www.manaraa.com

Table 2.8: Summary of splice junction discovery performance and read / base mapping rates in real datasets reported by MapSplice3 and five other aligners.

		Mapping rate		Junction	
Data	Tools	Read	Base	Annotated	Unannotated
R101	MapSplice3	96.56	95.99	187,468	172,955
	HISATx2	94.59	91.57	188,214	393,982
	STARx2	95.94	95.60	183,702	170,395
	GSNAP	96.85	95.92	188,738	421,810
	Tophat2	90.84	90.84	173,218	128,410
	MapSplice2	96.49	95.71	173,994	109,108
R262	MapSplice3	95.95	94.21	135,623	29,778
	HISATx2	90.04	90.04	136,259	39,889
	STARx2	94.42	90.16	134,782	30,047
	GSNAP	97.35	95.29	135,151	49,707
	Tophat2	85.48	85.48	128,658	22,824
	MapSplice2	97.66	89.14	130,499	25,564

المنارة للاستشارات
www.manaraa.com

Chapter 3 Individualized RNA-seq read alignment

3.1 Introduction

In the past decade, high throughput sequencing (HTS) has been established as one of the major technologies to investigate the genome, epigenome, and transcriptome from tissue samples or even a single cell. Besides its unprecedented resolution due to deep sequencing, one of the great advantages of HTS is its unbiased nature in sampling genomic content from any given subject, which is critical for the discovery of previously uncharacterized genomic features. Recognizing these opportunities, large studies such as TCGA (http://cancergenome.nih.gov/) and the 1000 genomes project [The 1000 Genomes Project Consortium, 2015] have generated both genomic data in the form of whole genome or whole exome sequencing and transcriptomic data in the form of mRNA sequencing (RNA-seq). Recently, an approach called G&T-seq has been developed and is capable of performing parallel sequencing of both the genome and transcriptome from a single cell. These datasets are often generated to study the relationships between genetic variation and the transcriptome among individuals or even between specific cells. In the meantime, the availability of these multi-omics data presents an opportunity for integrative analysis, where information from different datasets can be borrowed to enhance the performance of the commonly used single-omics analysis.

المنارة للاستشارات
www.manaraa.com

As we mentioned above, sequencing is a sampling process that is subject-specific and unbiased. Unfortunately, a level of bias can be reintroduced if RNA sequencing data are aligned to a reference genome to understand the origin of each read. Although the current state-of-the-art spliced aligners already perform quite well in the discovery of previously unannotated splice junctions and even some structure variations, they almost always utilize the reference genome of a species as the template for read alignment [Trapnell et al., 2009, Kim et al., 2013, Dobin et al., 2013, Kim et al., 2015, Wang et al., 2010, Wu and Nacu, 2010]. The consequence of the reference-based approach is that the alignment will favor sequences with highest identities to the reference genome. While this assumption ensures the accurate alignment of a vast majority of the sequence reads, it can introduce bias against a special category of reads that deviate slightly from the reference but have the potential to be biologically significant in a specific subject. Besides noise and errors, this set of reads is often attributed to single nucleotide variations (SNVs), small indels as well as splice variants as a result of splice site mutations. The inability to accurately align this set of reads is often referred to as reference bias. Its effect in generating high false positive in genotype calls as well as allele frequency estimations has been noted in several studies involving whole genome and exon sequencing [Brandt et al., 2015, Meynert et al., 2014]. With RNA-seq, this reference bias can lead to a deficiency in characterizing transcripts carrying alternate alleles instead of the reference, compromising the identification of allele-specific transcripts that may be critical to characterizing various biological phenomena such as cis-regulatory variation and nonsense-mediated decay [Castel et al., 2015].

Additionally, SNV can change the transcriptome through splice site mutation, where a non-canonical splice site can be mutated into a canonical splice site, thus enabling the

للاستشارات المنارة
www.manaraa.com

expression of alternative transcript variants. Such alterations in the transcriptome have the potential to either cause disease directly or contribute to the susceptibility or severity of disease [Ward and Cooper, 2010, Tazi et al., 2009, Zhang et al., 2013]. However, most of the RNA-seq aligners rely on the canonical or semi-canonical flanking bases in the reference genome to perform confident spliced alignments. As such, variant transcripts in a specific individual caused by splice site mutations at non-canonical splice sites can be substantially penalized unless this parameter is completely disabled in the mapping software program.

A straightforward approach to solve bias introduced by the reference genome would be to use personalized genomes, where the specific nucleotide sequence of each subject approximated by substituting SNPs at the corresponding reference coordinates. This strategy was adopted by rPGA [Stein et al., 2015] where, in addition to the reference genome, the reads are also mapped to the subject's two haplotype genomes. However, to perform alignments against a personalized reference genome, the first step would be to build an index for it. It takes hours of CPU time for the available spliced aligners to index a human reference genome. Besides, indexing files consume five to tens of gigabytes in storage for each genome. Thus, such an approach triples the amount of disk space and running time required when mapping to a single genome. This does not even count the time and extra post-processing steps necessary to merge individual haplotype's alignment results into one consensus alignment. Taken together, the computational requirements raised by mapping to subject genomes would substantially limit its efficiency when aligning datasets involving hundreds or even thousands of individuals.

In this dissertation, we propose a new approach for individualized RNA-seq alignment,

المنارة للاستشارات
www.manaraa.com

designated iMapSplice. It makes use of personal genomic information and performs an unbiased alignment towards a genome index carrying both the reference and any alternative bases. The approach is light-weight and does not require an index to be rebuilt for each individual. Importantly, it breaks the computational emphasis or dependency on the position of canonical splice site dinucleotide motifs in the reference genome and enables iMapSplice to discover personal splice junctions created through splice site mutations. We report comparative analyses using a number of simulated and real human datasets. The results demonstrate improvements in general read mapping, accurate alignment yields, and both the sensitivity and accuracy of splice junction discovery. At the same time iMapSplice dramatically reduces the biases in allelic ratio and discovers many personal splice junctions.

3.2 iMapSplice algorithm

iMapSplice efficiently utilizes data provided on genomic DNA single nucleotide variants during different steps in the MapSplice algorithm to recover read alignments that either harbor SNV (SNPs, small indels) or contain spliced alignments flanked by mutated splice sites. This section provides an overview of how the method works to address challenges faced by mapping reads only to a reference genome.

The left panel in Figure 3.1 illustrates an example of how an RNA-seq read carrying a SNP may fail to align correctly to a reference genome. The RNA-seq read in this example carries a SNP as well as a sequencing error. One of the general strategies used in the current fast aligners is iterative maximal prefix match [Dobin et al., 2013, Kim et al., 2015]. When searching for a prefix carrying the SNP against the reference, the correct mapping location

المنارة للاستشارات
www.manaraa.com

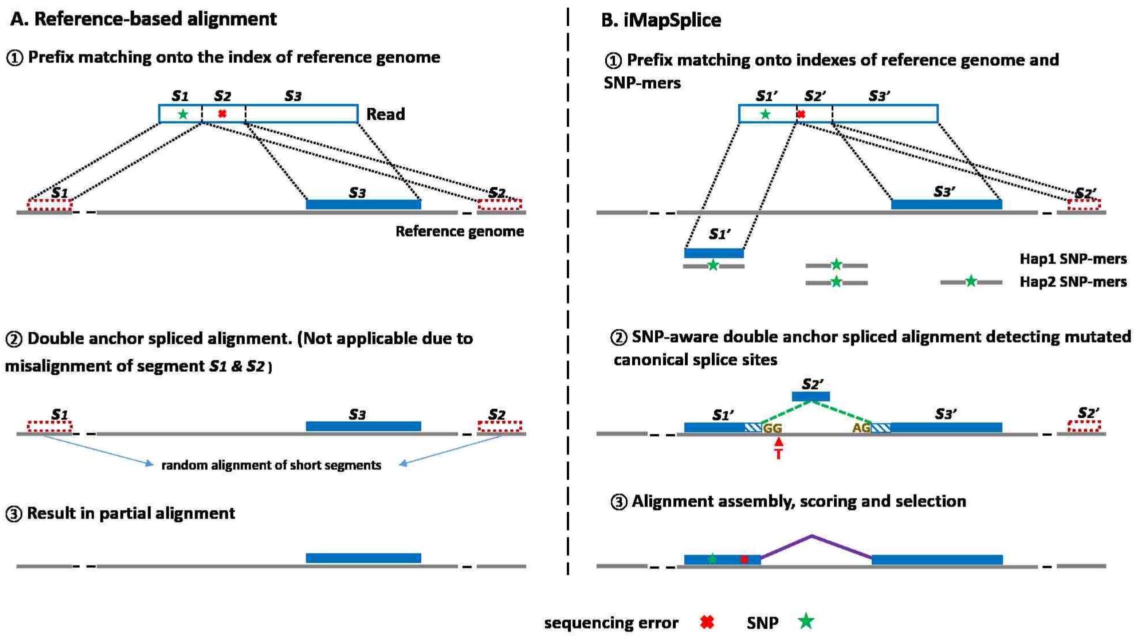

A. Reference-based alignment
① Prefix matching onto the index of reference genome
S1
S2
S3
Read
S1
S3
S2
Reference genome
② Double anchor spliced alignment. (Not applicable due to misalignment of segment S1 & S2)
S1
S3
S2
random alignment of short segments
③ Result in partial alignment
B. iMapSplice
① Prefix matching onto indexes of reference genome and SNP-mers
S1′
S2′
S3′
S3′
S2′
S1′
Hap1 SNP-mers
Hap2 SNP-mers
② SNP-aware double anchor spliced alignment detecting mutated canonical splice sites
S2′
S1′
S3′
S2′
GG
AG
T
③ Alignment assembly, scoring and selection
sequencing error ✖
SNP ★

Figure 3.1: (A) An example illustrating the challenge when mapping a RNA-seq read to the reference genome in the presence of SNPs. (B) An example illustrating how iMapSplice algorithm may resolve spliced alignment with SNPs as well as the basic steps of the alignment.

will be missed because no error is tolerated in this step. More often than not, short prefix alignments (<18bp) to random places (such as S_1 and S_2) are likely to be returned and have to be filtered out. Eventually, this may result in a partial alignment (S_3).

iMapSplice resolves this issue by including knowledge of the SNP in each alignment step, as shown in the right panel of Figure 3.1. The first step of iMapSplice searches for the exonic mapping of read segments through an approach called semi-maximal prefix match. Different from a maximal prefix search, in which only the mapping location of the longest match is returned, a semi-maximal prefix search returns all mapping positions with a match longer than a certain threshold (set as 30bp by default). In this step, it simultaneously maps reads to both reference genome as well as the exonic regions affected by SNPs, namely SNP-mers. A SNP-mer corresponds to a segment of genomic sequence carrying the vari-

www.manaraa.com

ant nucleotide, localizing it to the middle of the sequence.

Next, read mapping segments that are adjacent to each other on the genome will be merged. For two or more segments that are next to each other in the reads, but are separated on the reference genome, a spliced alignment is performed to bridge the two segments. We call this a double anchor spliced alignment. In this step, splice aligners including MapSplice have a high dependency on the presence of canonical dinucleotide splice donor and splice acceptor motifs (GT-AG, GC-AG, AT-AC) to avoid false positives. However, if the mapping software only considers the reference genome sequence, mutations in an individual that either change normal canonical splice site dinucleotide motifs or generate new ones can result in read misalignments. The dependency on canonical splice sites, therefore, greatly prohibits or discourages aligners from detecting novel but functional splice junctions that result from mutations which generate new canonical splice site pairings. Though some aligners are capable of reporting noncanonical splice junctions, high penalties are given to the alignment, which lowers mapping confidence and may lead to preference towards an alternative misalignment when multiple mapping options are available for the same read. To solve this problem, iMapSplice utilizes the information provided by nucleotide variants in the target region (from a hash table of SNPs) to create a list of candidate canonical splice sites to help in the determination of correct splice site pairings and improve read mapping accuracy. For example, in the second step shown in Figure 3.1B, with the known SNP (G $>$ T at the donor site), iMapSplice identifies the novel canonical donor splice site (GG $>$ GT) and completes the spliced alignment of segment $s_{2'}$. In conclusion, the SNP-aware double anchor alignment will utilize the SNP-information to

www.manaraa.com

identify personal spliced alignments that otherwise would be either missed or identified as non-canonical splice junctions.

The last step of iMapSplice completes segment assembly, candidate alignment scoring, and selection. In this step, aligned segments are assembled and different candidate alignments are generated with different potential combinations of segments. Candidate alignments are then scored based on the total number of mismatches, spliced alignment, and mapped length. The one with the highest score is selected as the primary alignment for each read. One of the most important metrics when scoring is the number of mismatches for each alignment. iMapSplice removes the mismatches that can be attributed to SNPs.

3.2.1 SNP-mer generation

SNP-mers can be derived in two ways corresponding to two variants of iMapSplice: iMapSplice-phased and iMapSplice-unphased. iMapSplice-phased applies when the genotype data are available (such as in the 1000 genomes project). In this case, SNP-mers of a fixed length k_{phased} (201bp by default) from the two genomic haplotypes are extracted. The reference nucleotides are then replaced with alternate nucleotides at all of the SNP positions. However, this approach cannot be generalized, since the genotype data are not available in many studies, where iMapSplice-unphased is applied.

In iMapSplice-unphased, a combination of variable length SNP-mers is used. Let k_{max} (201bp by default) and k_{min} (31bp by default) denote the maximum and the minimum length of a SNP-mer, respectively. In extracting the SNP-mer for each SNP, there are three scenarios according to the distance d between each SNP and its nearest neighbor SNP. When $d > k_{max}$, a SNP-mer of length k_{max} is extracted; when $k_{max} \geqslant d \geqslant k_{min}$, a SNP-mer

www.manaraa.com

of length d is extracted. In the two cases above, except for the specific SNP, all of the other bases are exactly the same as in the reference genome since there are no other SNPs within the genomic window where the corresponding SNP-mer covers. However, when $d < k_{min}$, there is more than one SNP within the k_{min} window size. This indicates a SNP-rich region. In such cases, there are 2^{n-1} SNP-mer possibilities, where n denotes the number of SNPs within that window. iMapSplice-unphased will randomly select m of them. The selection of m value effects the performance of iMapSplice-unphased on mapping reads covering this SNP. A small m may miss the perfect SNP-mer which matches read sequences without any error. However, a large m could confuse the aligner as too many possibilities are provided and also increase the running time. However, such SNP-rich regions are rare, and thus have little impact on the overall performance. Furthermore, the selection of SNP-mer length related parameters (k_{phased} in iMapSplice-phased, k_{max} and k_{min} in iMapSplice-unphased) also affects how well iMapSplice works. A SNP-mer has to be long enough to allow a partial read segment to be confidently mapped. However, a SNP-mer that is too long may not be necessary, as they will repeat the exact sequence from the reference genome.

Enhanced suffix array based indices of SNP-mers are built to facilitate the exonic alignment step (prefix match). Approximately 120,000 exonic SNPs were called for each individual human genome according to the 1000 genomes project study. Sequence extraction and index building can be completed in seconds, resulting in minimum overhead relative to alignment. iMapSplice performs an iterative semi-maximal prefix match of read sequences against both on the reference genome and SNP-mers. Segments mapped to SNP-mers will be converted to the reference genome coordinates and will be combined with other segments mapped to the reference genome.

المنارة للاستشارات
www.manaraa.com

3.3 Experimental results

In this section, we report the performance of iMapSplice regarding its capability for unbiased alignment of reads harboring SNPs and for the discovery of splice junctions with mutated splice sites.

Datasets and Setup for the Experiments

Performance was assessed using two types of data: one based on simulated RNA-seq reads and the other using real human datasets from the 1000 genome project.

Simulated datasets: Simulated RNA-seq reads were generated by BEERS [Grant et al., 2011] with two different mutation and error profiles. The low error reads were generated assuming a substitution frequency of 0.001, indel frequency of 0.0005 and base error frequency of 0.005. Corresponding rates in the high error reads were increased fivefold, 0.005, 0.0025, and 0.025 respectively. For both error categories, we generated two RNA-seq datasets with different read lengths, 50bp and 100bp. Each dataset contained 10 million pairs of paired-end reads with the same insert length of 200 bp. Note that the simulated data did not contain mutated splice sites and thus it cannot be used to assess the discovery of personal splice junctions as a result of splice site mutations.

Real datasets: Seventy-four RNA-seq datasets and their corresponding genotypes were downloaded from the Geuvadis RNA sequencing project [Lappalainen et al., 2013] and the 1000 Genomes Browser [The 1000 Genomes Project Consortium, 2015]. Numbers of reads in the RNA-seq datasets ranged from 44.6 million to 75.8 million. Approximately 2.5 million SNPs were detected for each individual, with roughly 83,000 of them localized

www.manaraa.com

Table 3.1: The set of tools and their index configuration used in performance comparison.

Category	Name	Version	Index
iMapSplice	iMapSplice-phased	1.0 Beta	reference genome (hg19) + individual phased SNPs
	iMapSplice-unphased	1.0 Beta	reference genome (hg19) + individual unphased SNPs
Ref-based	MapSplice	3.0 Beta	reference genome (hg19)
	HISAT2	2.0.5	reference genome (hg19)
	STAR	2.5,1	reference genome (hg19)
Mask-based	MapSplice MASK	3.0 Beta	reference genome (hg19) + SNP loci masked with Ns
	HISAT2 MASK	2.0.5	reference genome (hg19) + SNP loci masked with Ns
	STAR MASK	2.5.1	reference genome (hg19) + SNP loci masked with Ns
SNP-aware	HISAT2 SNP	2.0.5	reference genome (hg19) + individual unphased SNPs
	HISAT2 POP	2.0.5	reference genome (hg19) + population SNPs
	rPGA	2.0.0	reference genome (hg19) + two haplotype personal genomes

to the exonic regions according to human Gencode annotation (release 19) [Harrow et al., 2012]. All of the RNA-seq reads in the real datasets are paired-end reads and 75 base pairs in length.

To assess the performance of iMapSplice, we compared it to a number of publicly available RNA-seq aligners. All of the methods, version information, and the indices used are listed in Table 3.1. Default settings were used for all parameters.

3.3.1 Improvement in general read mapping

We first assessed improvements of read alignment performance that resulted from incorporating the knowledge of SNPs using the four simulated datasets. In this experiment, we compared iMapSplice against all the “Ref-based” tools and HISAT2 SNP. The “Mask-based” tools and rPGA are not proposed for general read mapping, and HISAT2 POP is not applicable since the SNP data were not consistent with the population profile. We col-

المنارة للاستشارات
www.manaraa.com

Table 3.2: The comparison of five baseline tools in terms of the number of accurate unique alignments out of 20 million synthetic reads in each of the four datasets with different read length and error profiles.

Methods	Low error 50bp	Low error 100bp	High error 50bp	High error 100bp
iMapSplice	18,931,227	19,168,761	14,934,484	16,918,697
MapSplice	18,839,058	19,121,804	13,485,776	16,378,977
STAR	18,31,6354	18,446,469	13,053,934	14,832,945
HISAT2	17,771,947	18,813,343	9,032,193	11,167,588
HISAT2 SNP	17,965,316	18,810,270	10,190,994	12,501,692

lected the numbers of accurate unique alignments reported by each method. As shown in Table 3.2, iMapSplice achieved the highest number of accurate unique alignments in each dataset. Incorporating SNPs into the index also improved the alignment performance for HISAT2 in three out of the four datasets. Note that the improved alignment percentage varied since it is a function of the percent of SNP affected reads, which is much less in low error data than in high error data. For the subset of simulated reads that achieved an improved accurate unique alignment with iMapSplice, we analyzed their alignment status with MapSplice that utilizes the typical method of mapping to reference genome (Table 3.3). In general, for short reads in both the low and high error rate categories, the majority of the improved reads were not mapped at all using the reference genome. Mapping longer reads significantly improved the alignment rate, which is reasonable since the short reads would have been more vulnerable to SNPs especially when partial alignments are allowed. For the longer reads, iMapSplice improved the accuracy by being better able to select correct unique alignment from multiple alignments, as well as completing partial alignments.

المنارة للاستشارات
www.manaraa.com

Table 3.3: Alignment categories that are improved by iMapSplice over MapSplice.

		Low error 50bp	Low error 100bp	High error 50bp	High error 100bp
Total improved alignment		98,055	53,381	1,516,038	672,871
MapSplice	Unmapped	47,439	1,522	947,351	165,273
	Muli-Incorrect	2,494	2,686	9,186	3,261
	Unique-Incorrect	10,929	8,629	143,409	42,341
	Multi-Partial	129	54	10,142	2,827
	Unique-Partial	19,148	13,125	235,418	394,494
	Multi	17,916	27,365	170,532	64,675

The uniquely aligned reads that are accurately detected by iMapSplice but not by MapSplice are organized into six categories. Unmapped: reads that were not be mapped at all; Multi-Incorrect: reads that displayed multiple incorrect alignments (none of the bases in read sequences were correctly mapped); Unique-Incorrect: reads were mapped uniquely but incorrectly; Multi-Partial: reads with multiple alignments, none of which were perfect, but at least one of them contained some bases in the read that were correctly mapped; Unique-Partial: reads with a single alignment and with some bases correctly mapped; Multi: reads with multiple alignments and one of them matched ground truth.

General splice junction sensitivity and specificity.

Splice junctions detected by the aligners were compared with ground truth. Detected splice junctions were categorized as correct if they matched ground truth exactly at both the splice donor and acceptor sites. We compared sensitivity and specificity of discovery in evaluating aligner performance of splice junctions with at least two supporting reads. Results are reported in Table 3.4. Sensitivity is the percentage of detected correct splice junctions among all the true junctions. Specificity is the fraction of detected correct splice junctions within all the detected junctions. In general and as expected, differences among the aligners were less when using the low error and long read simulated datasets. iMapSplice achieved the highest sensitivity in all four datasets. For specificity, iMapSplice also performed well, achieving the highest percentage in three out of four datasets and second highest in the other. For the two variants of HISAT2, the method making use of SNPs (HISAT2 SNP) exhibited higher sensitivity and specificity compared to read mapping to the standard reference genome (HISAT2) in almost all the datasets (except for the sensitivity in low error

www.manaraa.com

Table 3.4: Sensitivity, specificity, and F_1 score for splice junction discovery on simulated datasets.

	Low error 50bp			Low error 100bp			High error 50bp			High error 100bp		
Tools	Sensi	Speci	F_1	Sensi	Speci	F_1	Sensi	Speci	F_1	Sensi	Speci	F_1
iMapSplice	**90.44**	**98.90**	**94.48**	**97.28**	**98.91**	**98.09**	**77.49**	97.20	**86.23**	**93.71**	**97.67**	**95.65**
MapSplice	89.67	98.89	94.05	97.17	98.87	98.01	69.92	**97.39**	81.40	92.16	97.50	94.75
STAR	90.35	97.50	93.79	96.41	97.18	96.79	73.37	95.52	82.99	88.57	95.73	92.01
HISAT2	89.99	97.88	93.77	96.40	97.52	96.96	61.35	96.96	75.15	74.99	95.74	84.10
HISAT2 SNP	90.13	98.18	93.98	96.35	97.70	97.02	65.36	97.02	78.10	79.09	95.52	86.53

100bp data and the specificity in high error 100bp data).

3.3.2 Reduction of biases in allelic ratio

As pointed out by Brandt et al. [2015], mapping reads to a reference genome introduces an overestimation of reference allele frequency and underestimation of non-reference allele frequency. Reads carrying alternative bases at SNP coordinates have an extra mismatch compared to those with the reference nucleotide. In some cases, this extra mismatch would prohibit the aligner from mapping the read correctly, sometimes leading to a completely incorrect alignment or the end of the read being soft-clipped. One typical approach to alleviate the bias is to mask all the SNP positions with "N" before mapping the reads. Though allelic bias can be eliminated in this way, it also impairs the aligners' mapping ability as the reads are one more mismatch away from the reference genome. Recently developed HISAT2 provides another strategy through incorporating both variants and the reference genome into one graph and aligning read sequences to the graph paths. The strategy utilized by iMapSplice to alleviate the reference allelic ratio bias is to introduce the possibility of read mapping onto SNP-mers where alternative alleles are present.

Figure 3.2,3.3,3.4 shows the aggregated distribution of reference allelic ratio at all SNP positions in randomly selected human RNA-seq datasets from five individuals: NA12812,

المنارة للاستشارات
www.manaraa.com

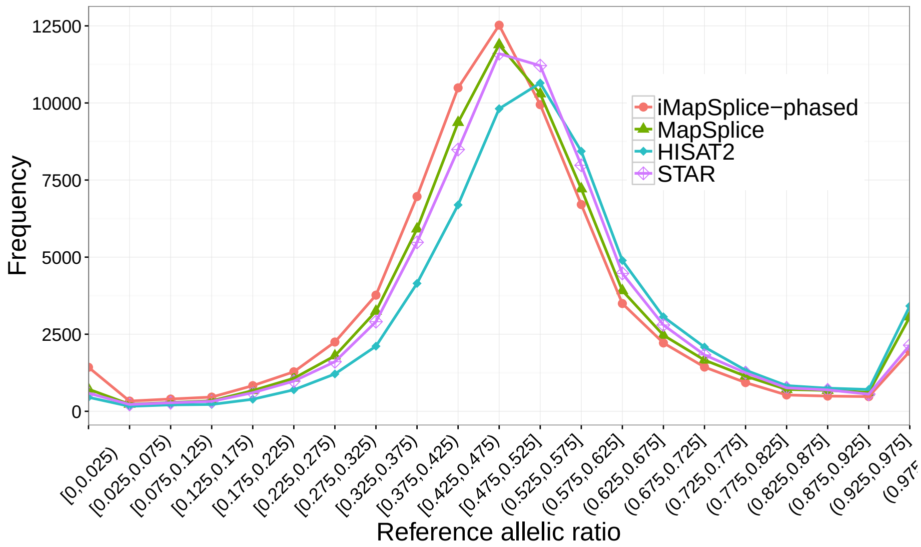


Figure 3.2: Aggregated reference allelic ratio distribution of iMapSplice and “reference only” methods (MapSplice, HISAT2, and STAR) on RNA-seq datasets from individuals NA12812, NA12749, NA07056, NA06994, and NA12275.

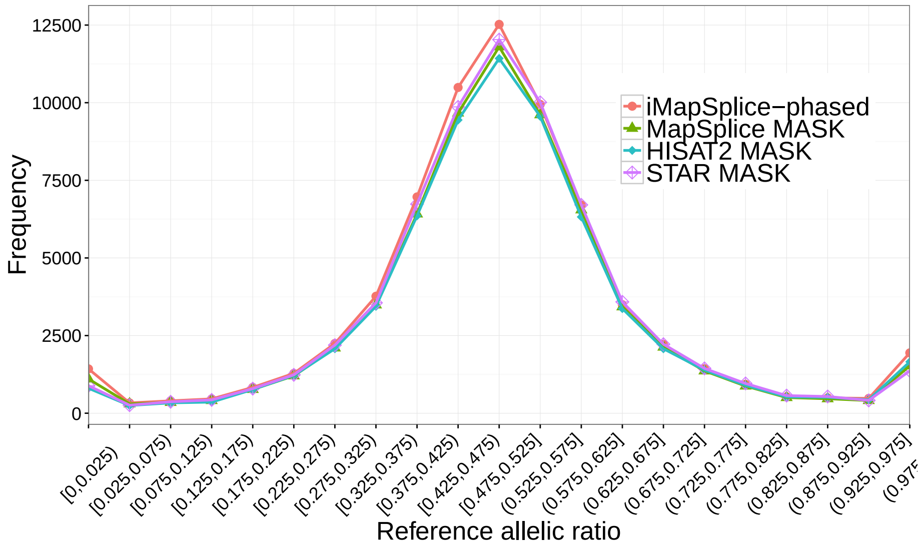


Figure 3.3: Aggregated reference allelic ratio distribution of iMapSplice and “mask-based” methods (MapSplice MASK, HISAT2 MASK, and STAR MASK) on RNA-seq datasets from individuals NA12812, NA12749, NA07056, NA06994, and NA12275.

NA12749, NA07056, NA06994, and NA12275. Compared to the other two category tools, the ”reference-based” programs (in Fig. 3.2) show significant bias towards the reference allele (reference allelic ratio > 0.5). Obviously, the choice of mapping strategy greatly

www.manaraa.com

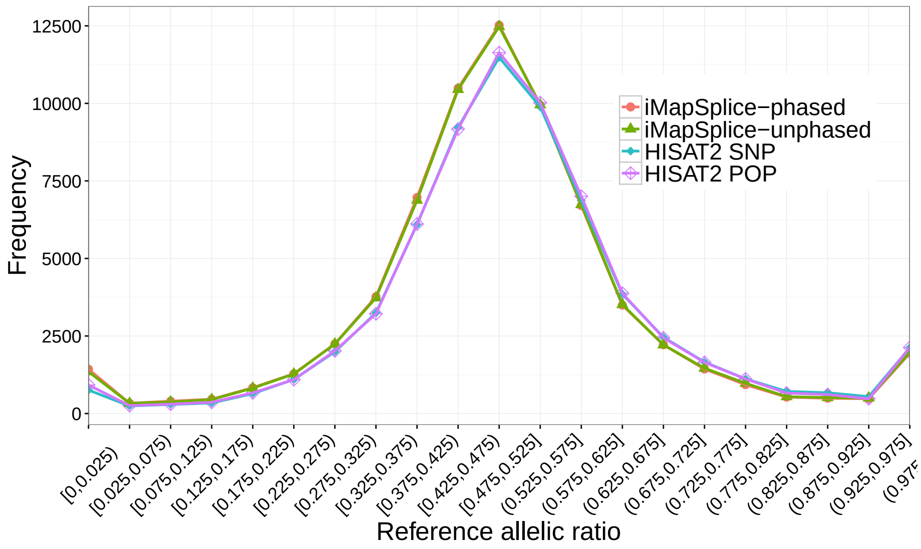


Figure 3.4: Aggregated reference allelic ratio distribution of iMapSplice and "SNP-aware" methods (HISAT2 SNP and HISAT2 POP) on RNA-seq datasets from individuals NA12812, NA12749, NA07056, NA06994, and NA12275.

affects the allelic bias independent of the aligner algorithms themselves. In Figure 3.3, both iMapSplice-phased and "mask-based" methods exhibit symmetric distributions with respect to the reference allelic ratio of 0.5. At the same time, however, iMapSplice-phased delivers the largest number of SNP positions with at least ten supporting reads (see Table 3.5). This is very likely due to the extra mismatches introduced by masking the SNP positions. As reported in Figure 3.4, the two variants of iMapSplice performs similarly, and show clear advantages over the two"SNP-aware" variants of HISAT2. These relationships are confirmed in the detailed numerical statistics reported in Table 3.5. Mean and skewness are used to characterize the bias for each distribution. iMapSplice and MapSplice Mask achieve the best performances in terms of mean and skewness respectively.

المنارة للاستشارات
www.manaraa.com

Table 3.5: Summarized comparison in terms of mean and skewness for reference allelic ratio distributions, and number of SNPs covered by at least ten reads.

Category	Tools	Mean	Skewness	SNP#(read count>10)
iMapSplice	iMapSplice-phased	.501	.171	68,899
	iMapSplice-unphased	.503	.195	68,750
Ref-based	MapSplice	.530	.460	67,290
	HISAT2	.563	.411	62,273
	STAR	.534	.327	66,768
Mask-based	MapSplice MASK	.502	.155	64.469
	HISAT2 MASK	.507	.297	63,199
	STAR MASK	.504	.196	66,154
SNP-aware	HISAT2 SNP	.522	.349	65,316
	HISAT2 POP	.519	.279	65,744

3.3.3 Discovery of personal splice junctions

We applied iMapSplice to datasets from 74 individuals. The numbers of detected novel canonical splice junctions created by splice site mutations are listed in Table 3.6. In total, iMapSplice reported 1,847 novel splice junctions with at least two supporting reads associated with nucleotide changes (mutations) that created a new canonical splice donor and acceptor pair. Among them, 1445, 589, 280, 123, and 26 appeared in at least 2, 5, 10, 20, and 50 individuals, respectively. Additionally, we compared our results against those recently reported in another study [Stein et al., 2015]. As shown in Table 3.6, iMapSplice achieved higher sensitivity, especially with junctions found in fewer individuals or supported by fewer reads. Even for those shared by a large number of individuals and reads, iMapSplice displayed improved performance. The eight personal splice junctions experimentally validated previously were all successfully detected by iMapSplice.

Effects of splice site mutations on expression iMapSplice enabled the discovery of two general types of splice site mutations: (i) the gain of a canonical splice site (GT-AG, GC-AG, AT-AC) and (ii) the loss of a canonical splice site. We examined how these mutations

www.manaraa.com

Table 3.6: Detected personal splice junctions changing from noncanonical to canonical with increasing numbers of supporting individuals and reads.

Read coverage threshold		Reported by iMapSplice				
		⩾2	⩾5	⩾10	⩾20	⩾50
Number of	⩾1	1,847	896	492	310	177
individuals	⩾2	1,445	782	470	299	175
with splice	⩾5	589	589	418	278	164
site mutated	⩾10	280	280	280	222	133
junction	⩾20	123	123	123	123	102
threshold	⩾50	26	26	26	26	26

(a)

Read coverage threshold		Previously reported by rPGA [Stein et al., 2015]				
		⩾2	⩾5	⩾10	⩾20	⩾50
Number of	⩾1	380	313	237	167	86
individuals	⩾2	308	280	219	156	82
with splice	⩾5	208	208	192	141	74
site mutated	⩾10	138	138	138	120	68
junction	⩾20	69	69	69	69	57
threshold	⩾50	9	9	9	9	9

(b)

affected steady-state expression (read counts) at the corresponding splice junctions (Table 3.7). In this experiment, we categorized all the splice junctions according to changes in average coverage between the individuals with and without splice site mutations.

Among both types of mutated splice junctions, although many had low coverage, iMapSplice detected hundreds of personal splice junctions that exhibited significant expression changes in association with the gain or lose of canonical splice site motifs.

For the mutated splice junctions in type (i), (Table 3.7A), previously non-functional splice sites increased to 2-10 reads at 201 sites and more than 10 reads at 81 sites. In contrast, loss of a canonical splice donor/acceptor pair (ii) significantly inhibited splicing expression (Table 3.7B).

Mutations in specific splice sites from genes *C14orf159*, *ANXA6*, and *TMEM216* are examples of the two types of mutations. The mutation (C $>$ T) in *C14orf159* (Figure

المنارة للاستشارات
www.manaraa.com

Table 3.7: Number of splice junctions falling into each category of average coverage changes between the individuals with and without mutations in splice sites (a). mutated from noncanonical to canonical; (b) mutated from canonical to noncanonical

		Average coverage on mutated junctions (canonical)			
		0	(0,2]	(2,10]	(10,+∞)
Average coverage on unmutated junctions (noncanonical)	0	0	4,043	**201**	**81**
	(0,2]	30	8	2	**2**
	(2,10]	0	0	1	0
	(10,+∞)	0	0	0	0

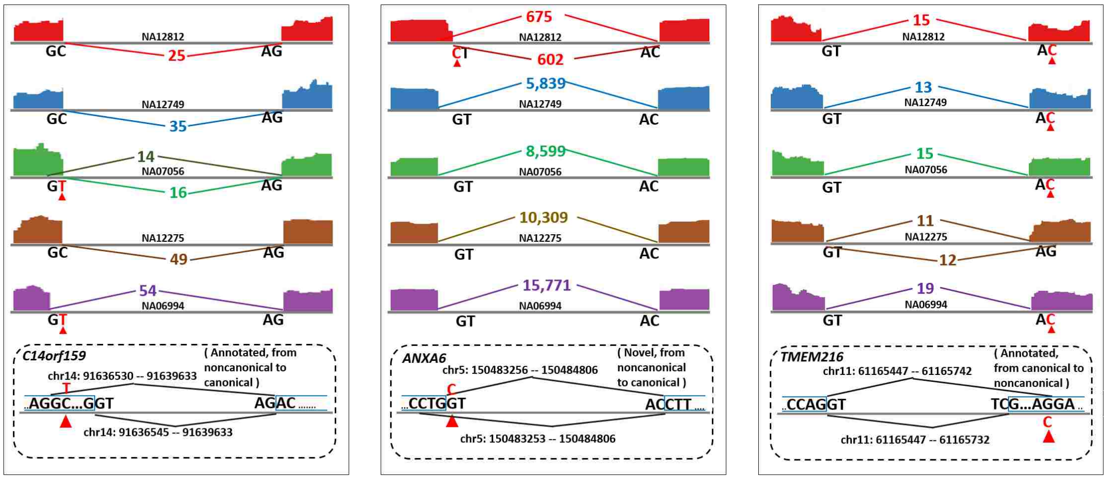
(a)

		Average coverage on mutated junctions (noncanonical)			
		0	(0,2]	(2,10]	(10,+∞)
Average coverage on unmutated junctions (canonical)	0	0	155	0	0
	(0,2]	872	37	2	1
	(2,10]	**10**	4	0	1
	(10,+∞)	**7**	**3**	0	1

(b)

3.5(a)) created a canonical donor site (GC $>$ GT) and led to a novel canonical splice junction in two individuals NA07056 and NA06994. The same splice junction did not show up in RNAseq data from the other three individuals without this mutation. The mutation (G $>$ C) in *ANXA6* (Figure 3.5(b)) also created a canonical splice site (acceptor site, GT $>$ CT, reverse strand). However, it is different from the one in *C14orf159* that converted an annotated semicanonical splice site to a canonical splice site. It created a completely novel splice junction (previously unannotated). The mutation (G $>$ C) in *TMEM216* (Figure 3.5(c)) affected expression at the splice junction in the opposite way. This mutation corrupted the canonical acceptor site (AG $>$ AC). Four individuals (NA12812, NA12749, NA07056, and NA06994) carrying this nucleotide variant lost the canonical splice junction that appeared in the nonmutated individual (NA12275) with the reference allele.

www.manaraa.com



(a) (b) (c)

Figure 3.5: Eamples of mutations and their corresponding impact on expression from the creation or loss of canonical splice junctions. Bases in black are the reference nucleotides, and those in red are alternate bases at SNP positions. The numbers in the figure show supporting read counts for the corresponding splice junctions.

3.3.4 Super fast and space-saving indexing strategy

As mentioned in the Methods section, iMapSplice adopts a very efficient indexing strategy. Compared to a direct approach of rebuilding the index for each personalized genome, this strategy saves a great amount of time and space thereby lowering the risk of computational bottlenecks with large-scale applications. iMapSplice takes less than 1 minute to build an index and the indexing files require only around 0.32 Gigabytes in storage. In contrast, as adopted by [Stein et al., 2015], to rebuild the index of an entire personalized genome for each haplotype, rPGA takes approximately 86.3 minutes of CPU time creating indexing files that are as large as 26.33 Gigabytes. Table 3.8 lists the indexing file storage usage and the runtime of indexing and mapping for iMapSplice on the 74 RNA-seq datasets as well as the estimated results for the other two individual SNP incorporated methods HISAT2 SNP and rPGA. iMapSplice achieved significant advantages in terms of both storage usage

www.manaraa.com

Table 3.8: Indexing file size and running time to align the 74 RNA-seq dataset reads. Note: indexing file storage usage and run time for HISAT2 SNP and rPGA are estimated according to their performance on a small subset of the datasets.

Tools	indexing file storage usage (GB)	time (hour)		
		indexing	mapping	total
iMapSplice	90.2	0.3	38.6	38.9
HISAT2 SNP	408	142	14	156
rPGA	5,845	319	44	363

Experiments were run on clusters with nodes equipped with Dual Intel Xeon CPUs E5-26708@2.60GHz and 64GB of 1600MHz RAM.

and runtime.

3.4 Conclusion

RNA-seq is a widely adopted technique used in transcriptional profiling for a wide range of applications including differential expression analyses, novel isoform prediction, genomic variants calling, RNA editing, and so on. In most of these applications, especially those that rely on a reference sequence, a critical step is to correctly map each RNA-seq read onto the corresponding specific nucleotide coordinates in the reference genome.

However, polymorphic variants such as SNPs may potentially cause the incorrect or incomplete alignment of reads, prohibit the discovery of personal splice junction, and skew expression coverage in the affected regions. As a result, downstream analyses including transcript reconstruction, alternative splicing analysis, and quantitative measurements of transcript expression are compromised. Although statistically, they may only affect a small proportion in each category on the whole genome, their functional importance cannot be overlooked as evidenced by existing research [Munger et al., 2014].

Our evaluation demonstrates that iMapSplice significantly improves the accuracy of RNA-seq read alignment by taking into account both the reference genomic sequence and

www.manaraa.com

personal SNP variants. The software performs an unbiased mapping of reads carrying either the reference or alternative base sequence. Comparative results show that the reference allelic ratio distribution derived from the alignment results using iMapSplice exhibits the closest mean relative to 0.5 and the skewness value is also among the smallest ones. Those observations demonstrate that iMapSplice significantly alleviates the reference allelic ratio bias which is a common deficiency for sequencing read aligners. Additionally, SNP variants in an individual can generate novel canonical splice junctions or alternatively introduce a base change that alters a canonical splice donor to splice acceptor pairing (GT-AG, GC-AG, AT-AC). Resulting changes in splice site utilization can be functionally significant and are important to detect. iMapSplice enhances the detection of personal splice sites by considering both reference and individual alternate alleles in determining the optimal alignment for each RNA-seq read.

Performance-wise, iMapSplice is a lightweight approach with minimum overhead in both storage and running time compared to other alignment methods that are also capable of considering individuals SNPs during the mapping process. iMapSplice can be readily applied to the datasets collected in large consortium, such as TCGA, ICGC as well as the 1000 genomes project by taking either the original reads or the alignment file as input, making it possible to uncover functionally important personalized transcript variants as a result of either mutated splice site or allele specific transcript. We expect to continue to improve iMapSplice in the near future to incorporate other structure variations such as small indels. The alignment strategy will be fairly similar and the extension should be straightforward.

As sequencing technologies continue to advance and it becomes more common to ob-

المنارة للاستشارات
www.manaraa.com

tain genome sequencing (or SNPs) and RNA-seq data in parallel, we think iMapSplice has the potential to be a widely used computational tool not only for obtaining more reliable read alignments, but also to connect genomic mutations with functionally important variation in splice site utilization. Both of them are indispensable for the characterization of personalized transcriptome and the realization of precision medicine.

المنارة للاستشارات
www.manaraa.com

Chapter 4 Scalable query over large scale sequencing datasets

4.1 Introduction

As introduced in Chapter 1, The vast amount of sequencing data shared through public databases provides invaluable resources for researchers to test hypotheses by reusing existing datasets. However, the computational solution to index and query large-scale sequencing data remains an unmet need.

Very recently, Sequence Bloom Tree (SBT) [Solomon and Kingsford, 2016] and its improvements SBT-AS [Sun et al., 2017] and Split-SBT [Solomon and Kingsford, 2017] were developed to retrieve sequencing experiments based on sequence content. Given a query transcript, SBT returns the set of experiments whose sequencing reads contain a significant proportion of k-mers present in the query. The sequence query with SBT can be described as a process trying to narrow down the search from unions of sequencing samples to smaller subsets and eventually to individual samples.

In this dissertation, we propose a drastically different approach to tackle this problem. Our approach split the sequence query into multiple *k*-mer queries of their appearance information across all samples. The results from the *k*-mer query are then taken together to determine the query results. This is achieved by a novel indexing structure, namely SeqOthello, which employs a compact two-level hierarchical indexing structure of *k*-mers. Each node is implemented as a compact hashing classifier called Othello [Yu et al., 2017]. Given the fact that *k*-mers in the same transcript often co-occur, thus sharing similar fre-

المنارة للاستشارات
www.manaraa.com

quencies across multiple samples, the entire *k*-mer set is divided into frequency groups forming the bottom layer of SeqOthello. The root node of the hierarchy provides classification of a *k*-mer to its frequency group, the search within which further provides the occurrence information of *k*-mer across all samples. We have conducted comprehensive experiments on SeqOthello to compare its accuracy and scalability with those of SBT. Comparison of SeqOthello against SBT demonstrated its superior performance in both query speed, disk and memory usage. All the SeqOthello implementation variants were around two orders of magnitude faster than SBT on a large query (36,076 sequences) without loss of accuracy and compromises in memory usage. In the meantime, SeqOthello achieves a compression ratio of 165:1 (913:1 with compressed *k*-mer), compared to that of 25:1 for SBT.

4.2 Sequencing data *k*-mer content

k-mers: A genomic sequence is a string composed of elements from the alphabet set $\sigma = \langle \mathrm{G}, \mathrm{T}, \mathrm{A}, \mathrm{C} \rangle$. A *k*-mer is genomic sequence of length k. For any sequence of length L, there exist a maximum of $L-k+1$ possible *k*-mers. A genomic sequence can be represented by the set of *k*-mers present in the sequence, for which reason *k*-mers are considered the building blocks or DNA “words” of genomic sequences. By iterating through *k*-mers of each read in a sequencing data, one can collect the total set of *k*-mers present in a sample as well as the number of reads containing each *k*-mer. Jellyfish [Marçais and Kingsford, 2011] is one of the efficient programs in *k*-mer enumeration.

المنارة للاستشارات
www.manaraa.com

4.2.1 Compressed *k*-mer

Compressed *k*-mers: The total number of unique *k*-mers can be reduced to a compressed set by an approach called homopolymer compression (HC) transformation strategy [Au et al., 2012]. The compressed version of a *k*-mer is achieved by replacing consecutive duplicate bases with a single base. For example, a raw *k*-mer ”AAAAATTTCCGAAAAATTTCC” can be transformed to ”ATCGATC”. Employing this approach, 1,099 billion possible unique raw exact *k*-mers of length 20 is transformed to 7 billion compressed *k*-mers, achieving a compression ratio 157:1. It is important to note that two sequences carrying the same set of exact *k*-mers will always have the same set of compressed *k*-mers. Two sequences carrying the same set of compressed *k*-mers may not always but is highly likely in having the same set of compressed *k*-mers.

Sequencing information of a RNA-seq sample can be reduced to the complete set of *k*-mers present in its reads. The query of a transcript against this sample can be determined by whether all *k*-mers in the query (or a large proportion to tolerate noise) are present in the sample. To determine whether a sequence exists in a number of samples, one would need the information of the occurrence of its *k*-mers in each sample. It will be desirable if we have access to an occurrence map that contains such information.

The motivation behind the proposed algorithm, SeqOthello, is to provide a fast and efficient mapping from *k*-mers to their corresponding *occurrence map*s. The specific implementation is driven by the inherent similarity of occurrence among adjacent *k*-mers within a biological sequence, where these *k*-mers are highly likely to be sequenced in one read and/or co-sequenced in other reads of the same sequence to maintain similar cover-

www.manaraa.com

age. Such similarity extends to the occurrence map across many samples. Thus, if *k*-mers with similar occurrence can be indexed together, it will ultimately improve the locality of a search. How to identify such similarity without knowledge of annotation is a challenging question. Without involving clustering [Simpson et al., 2009], which is beyond the scope of the current work, we take a simple but a bit crude approach, where we divide the frequency spectrum $(0, 1]$ of the *k*-mer occurrence into *buckets*, with each of them corresponding to a frequency range. The *k*-mers are allocated to corresponding buckets based on their occurrence frequencies across samples. Thus, they are naturally divided into multiple disjoint sets.

Figure 4.1 shows the two-layer hierarchy of SeqOthello. The frequency buckets storing mapping from *k*-mer to occurrence map form the bottom layer. And they are indexed by an Othello [Yu et al., 2017] in the top layer of the hierarchy. A sequence query starts from the root to identify buckets containing all its *k*-mers, followed by the search of *k*-mer occurrence map in each of the identified buckets. The occurrence map of these *k*-mers will then be synthesized to determine the samples containing the query sequence.

4.2.2 Mapping between *k*-mer to occurrence map within a frequency bucket

Given a set of sequencing samples, the frequency of a *k*-mer is defined as the fraction of samples where it is present as captured by sequencing reads. The distribution of *k*-mer frequency is in a ”U” shape (Fig 4.2). This makes perfect biological sense. The low frequency *k*-mers unique to one or two samples represents individual difference and noise, while the high frequency *k*-mers may come from the group of house keeping genes which are required to be present in almost all samples, both of which constitute a significant fraction of

www.manaraa.com

total unique k-mers. The frequency between thus becomes more typical to k-mers representing special conditions, such as tissue specificity. We divide the frequency domain $(0,1]$ into b disjoint frequency ranges $\{r_1, r_2, \ldots, r_b\}$. The k-mers whose frequency fall in r_i will constitute the frequency bucket B_i. Considers for frequency split are discussed following implementation details of frequency bucket. The bottom layer of SeqOthello contains the set of frequency buckets $\{B_1, B_2, \ldots, B_b\}$. Each frequency bucket is implemented using a novel hashing data structure Othello [Yu et al., 2017] in order to maintain a mapping from the k-mers to their occurrence maps. The occurrence map of a k-mer encodes the k-mer's presence or absence information in all samples. There are essentially two ways to store the information. One is to store the list of sample IDs containing the k-mer; the other is to store a bitmap where the presence and absence are encoded as 1 and 0 respectively. To optimize the memory usage, we adopt a hybrid approach where the first approach is chosen for low information density (such as low presence or absence), where the latter is used for the rest.

Mapping from k-mer to frequency bucket

The top layer of SeqOthello hierarchy performs the mapping from a k-mer to its frequency bucket. The structure is determined by the total set of k-mers present in at least one sample, i.e., $\cup\{B_1, B_2, \ldots, B_b\}$, and the total number of the frequency buckets b.

A typical implementation of such mapping is to use a $\lceil \log_2 b \rceil$-Othello structure. The memory overhead for storing such Othello is at most $4\lceil \log_2 b \rceil n$ bits, where n is the number of unique k-mers among all buckets. In order to assist memory alignment and improve query efficiency, we recommend the value of $\lceil \log_2 b \rceil$ to be some small powers of 2 (namely 2, 4, 8, or 16).

www.manaraa.com

4.2.3 Sequence query on SeqOthello

Given a sequence of length w and a threshold value θ, $0 < \theta \leq 1$, SeqOthello reports whether this sequence is present or not in each of the m samples.

The query process on SeqOthello works as follows. First, SeqOthello computes the occurrence bitmap of all $w-k+1$ k-mers. In order to utilize the locality provided by k-mer frequency buckets, SeqOthello first execute the Othello query on the root layer to classify these $w-k+1$ k-mers into frequency buckets. Then for each frequency bucket, if there is one or more k-mer mapped into this bucket, SeqOthello queries the second layer structure to get the occurrence bitmap of these k-mers. Such procedure will generate all $w-k+1$ occurrence bitmaps for each of the k-mers. Then SeqOthello assembles the bitmaps into the final result: the sum of the i-th bit of all the $w-k+1$ bitmaps is the total number of k-mers occurs in the i-th sample. When this value is greater than $\theta(w-k+1)$, SeqOthello reports that the sequence is present in the i-th sample.

In the above procedure, SeqOthello only accesses the root node during the first step and all the second layer mapping structures are accessed one after another. Hence, the memory space occupied by a node (either the root layer Othello or a second layer mapping structure) can be released once all the queries assigned to this node finish. This property is particularly desirable for memory-limited devices. Note that such procedure can also be implemented for batch queries. For multiple sequences, SeqOthello executes all root layer queries for all the sequences together and then execute the second layer query for each frequency bucket one by one.

We have implemented three variations of SeqOthello based on different sets of k-mers

المنارة للاستشارات
www.manaraa.com

extracted from sequencing data.

- SeqOthello-Exact. This is the default mode, where SeqOthello indexes all the *k*-mers present in the sequencing samples. “Exact” is to differentiate this mode from “Compressed”.
- SeqOthello-Compressed. SeqOthello is built on the compressed set of *k*-mers derived from the whole set of exact *k*-mers using homopolymer compression (HC) transformation strategy.
- SeqOthello-UniversalCompressed. In this case, SeqOthello is built on the entire universe of compressed *k*-mers, including both compressed *k*-mers that are present in the sequencing data and those that are not. This variation allows us to check the issue of alien *k*-mers by comparing with the other two approaches. It also provides us the upper bound on the size of SeqOthello-Compressed by including all compressed *k*-mers.

4.3 Transcript query over 148 equine RNA-seq datasets

In this section, we report the results of querying the previously annotated transcript sequences over 148 equine RNA-seq datasets [Adam, 2016].

The three implementation variations of SeqOthello are compared with Sequence Bloom Tree (SBT), version beta v0.3.5. We are aware of two very recently proposed methods, Split Sequence Bloom Tree (SSBT) [Solomon and Kingsford, 2017] and AllSome Sequence Bloom Tree (SBT-ALSO) [Sun et al., 2017]. Both methods are incremental work to im-

www.manaraa.com

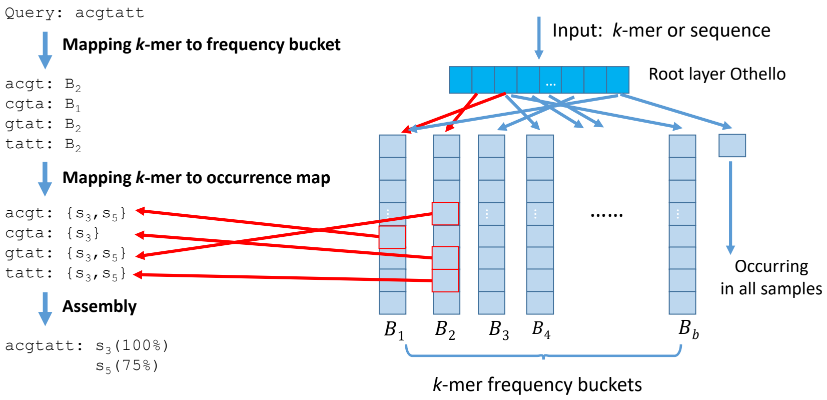


Figure 4.1: Left: A toy example illustrating steps of sequence query over SeqOthello structure. Right: Two-layer hierarchical structure of SeqOthello (note that *k*-mer occurrence map is not shown). For any given query, SeqOthello first enumerates its *k*-mers and maps them to corresponding frequency buckets using the first layer Othello. Secondly, the associated frequency buckets in the second layer will be loaded into memory to extract the occurrence map for each *k*-mer falling in its bucket. In the last step, occurrence maps of all *k*-mers will be assembled to identify samples with the presence of *k*-mers more than a certain threshold θ.

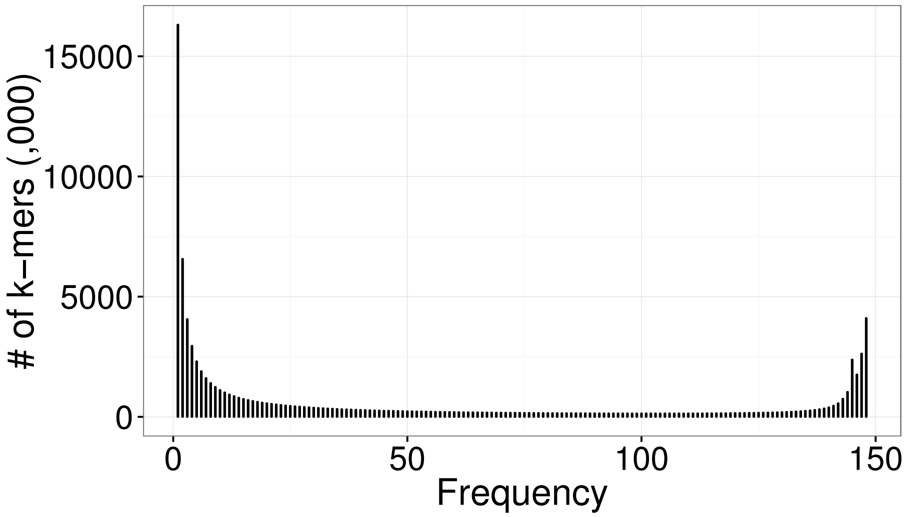


Figure 4.2: *k*-mer frequency distribution in 148 equine RNA-seq datasets used in the experiment section.

prove the performance of SBT by minimizing the number of nodes visited during a query. According to their manuscripts, which are in pre-prints, SSBT showed a five-fold improvement in search speed and disk usage, while SBT-ALSO achieved a two-fold improvement in construction speed and a three to seven-fold advantage on query speed. SeqOthello

www.manaraa.com

adopts a very different strategy with much more significant overall improvement than these two methods.

Performance of SeqOthello is generally determined by the following parameters: (1) θ, the minimum fraction of *k*-mers identified in a sample in a query sequence in order to call it as present; (2) *k*, the length of *k*-mer; (3) *mincov*, the minimum number of reads in a sample containing the *k*-mer in order to call it present in the sample. (2) and (3) both determine the total number of unique *k*-mers collected to represent each sample during the index construction. By default, 20-mers are used and *mincov* is set to be 3, unless otherwise specified. In our experiment, the frequency buckets in SeqOthello are assigned using a very simple method, where each frequency bucket contains *k*-mers with exactly the same frequency count, totaling 148 buckets.

The dataset used in our experiment consists of RNA-seq data from 148 equine samples in order to study tissue specific gene/transcript expression under 20 varying conditions, including equine cartilaginous tissues and induced chondrocytes [Adam, 2016]. The entire datasets consist of 6.1 billion 100bp paired-end reads, totaling about 1.28 TB disk space.

To evaluate the accuracy of our approach, we performed a batch query of all 36,076 transcript sequences, obtained from Ensembl equine database (release 87). The query result of each sequence is compared against the ground truth of the presence of the sequence in each sample. RSEM [Li and Dewey, 2011b] was performed to estimate the transcript expression in the unit of TPM (Transcripts per Million) in all these samples. The positive result of a query includes the set of samples where the input transcript is expressed higher than a given TPM value (1, 10, and 100 are used for our experiments) and the negative set includes the samples where the TPM of the transcript is lower than 1. The accuracy

المنارة للاستشارات
www.manaraa.com

is assessed by ROC curves, the correlation between the true positive rate (sensitivity) and false positive rate (1-specificity).

In the first experiment, we look at how the accuracy of different SeqOthello variations compare with SBT. Figure 4.3 and Figure 4.4 show the ROC curves corresponding to two different TPM cutoffs, 1 and 100 respectively. For each point in the ROC curves, the true positive rate and false positive rate are computed for a θ (varied from 0.5 to 1 with a step size of 0.02). Note that θ is the minimum proportion of k-mers present in a sample in order to call the existence of the query sequence in the sample. Larger θ requires more proportion of k-mers present in a sample and less tolerates noises present in individual difference. This naturally leads to small values of both true positive rates and false positive rates.

SeqOthello-Exact shows slightly better performance than SBT across both TPM cutoffs, with ROC curves sitting on the left side of SBT. But the difference is minimal. Note that we do expect their performance to be similar because they use the same underlaying k-mer and similar metric θ to determine transcript presence. The only factor makes them different is probably the effect of alien k-mers, which will not be the same between bloom filter and Othello.

SeqOthello-Compressed, in comparison, offers quite impressive accuracy, its ROC curve almost overlaps with SeqOthello-Exact when the TPM cutoff is 1 (Figure 4.3), but it has higher false positives with TPM = 100 (Figure 4.4). The major difference between the two methods is that the same sensitivity and specificity of SeqOthello-Compressed are obtained at a much higher θ (0.9-0.96) than that of SeqOthello-Exact (θ=0.8). This means that it requires more fraction of compressed k-mers than exact k-mers in order to distin-

www.manaraa.com

guish the presence and absence of a query sequence. That indeed makes sense because we are trying to use a much smaller set of compressed *k*-mers to represent the information of exact *k*-mers. One compressed *k*-mer may correspond to multiple exact *k*-mers. Thus they are more likely to show up in a sequence as random noise. A more stringent θ is required to weed out false positive query result.

Next we evaluate the effect of alien *k*-mers, the *k*-mers that are not present in any sample, to the query. Othello [Yu et al., 2017] is a probabilistic hashing classifier and will not give 100% accurate query result for alien *k*-mers. The SeqOthello-UniversalCompressed considers the complete set of compressed 20-mers. Therefore, no *k*-mers in the compressed format are alien to them. The ROC curves of SeqOthello-Compressed and SeqOthello-UniversalCompressed almost completely overlap with each other in both figures, with slightly better specificity in the latter approach. This demonstrate that alien *k*-mers does not pose a major issue in the accuracy of sequence query. The problem is mitigated by the requirement of many *k*-mers in a sequence.

To further evaluate the capability of the model in eliminating real false positives, we query 60,730 mouse transcript sequences from mouse Ensemble gene annotation release M12 towards SeqOthello indexing equine sequencing reads. Using the default setting, there are only 6 of the total 60,730 queries towards SeqOthello-Exact returned occurrence in at least one sample. We then used the BLAT tool (https://genome.ucsc.edu/cgi-bin/hgBlat?command=start) to search those query sequences against the equine genome (cab2). All of them were successfully aligned with a minimum score of 54, showing that they are likely to be homologous genes. This exercise demonstrated that the *k*-mer based sequence query methods are highly reliable.

www.manaraa.com

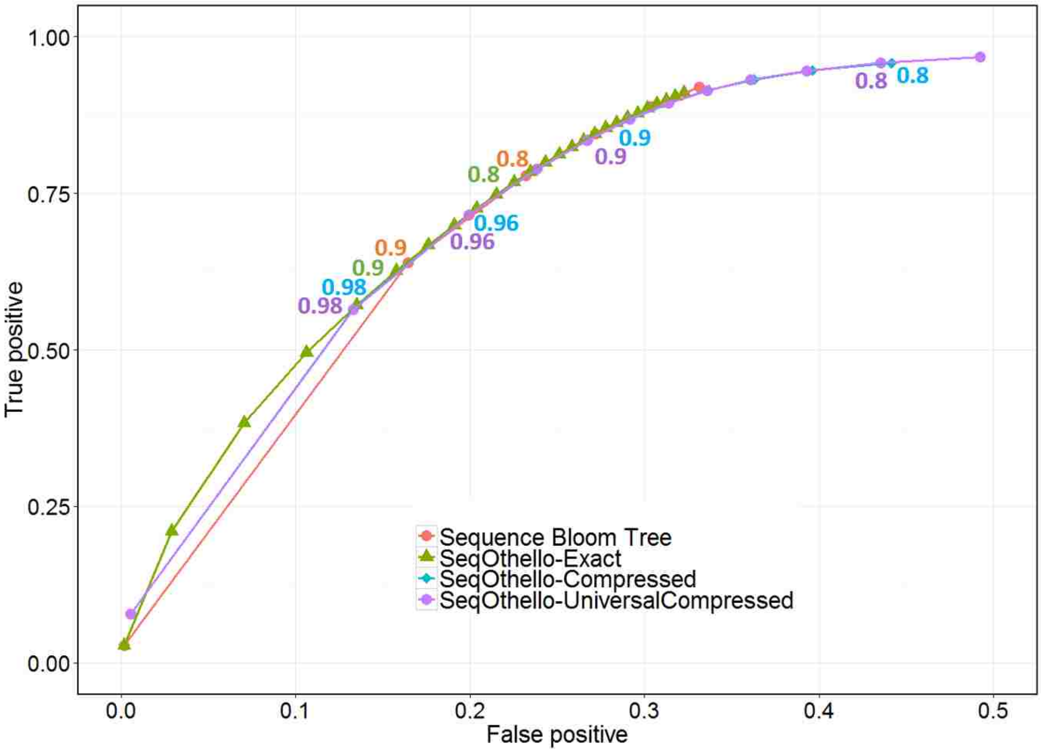


Figure 4.3: Comparison of transcript sequence query results between SBT and SeqOthello implementation variants. Transcripts with TPM≥1 were considered as present in a sample. Transcripts with TPM<1 were considered as absent in a sample.

Query accuracy as a function of transcript expression, *mincov*, and *k*

We next conducted a number of experiments to understand the behavior of SeqOthello in different parameter settings. In this experiment, SeqOthello-Exact is used. We first look at the accuracy of the algorithm in detecting transcripts with different expressions. Three TPM cut-offs, 1, 10, and 100 are used to determine the positive set of transcript presence. Figure 4.5 shows that query sensitivity increases significantly with higher transcript expression.

To investigate whether the *mincov* affects the query accuracy. *mincov* is the minimum number of reads in a sample required to support a *k*-mer in order to call its presence. Note that *k*-mers with support smaller than *mincov* will be removed from consideration

www.manaraa.com

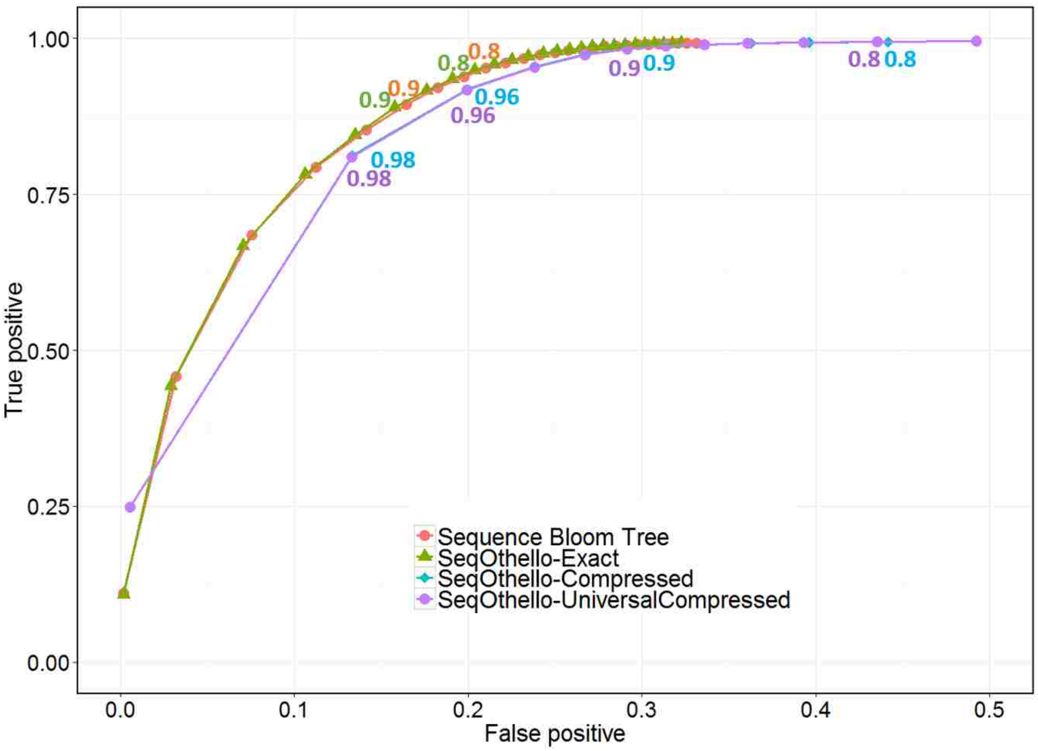


Figure 4.4: Comparison of transcript sequence query results between SBT and SeqOthello implementation variants. Transcripts with TPM$\geq$100 were considered as present in a sample. Transcripts with TPM$<$1 were considered as absent in a sample.

in order to remove noise. Figure 4.6 presents the ROC-curves when different *mincov* are applied to construct SeqOthello. It shows that higher *mincov* significantly minimizes false positive rates while almost maintaining the same sensitivity, suggesting that $mincov = 10$ or even $mincov = 20$ can be more desirable in real applications if false positive is more of a concern. Figure 4.7 showed the effects of *k*-mer length on the performance. While longer *k*-mers have shown advantages in other sequencing application, such as metagenomics, the improvement offered by 31-mer over 20-mer in SeqOthello are not quite significant in querying performance, suggesting 20-mer might be the most cost-efficient way to build the system, as it is much smaller in size.

All the experiments were performed using the servers from Lipscomb High-Performance

المنارة للاستشارات
www.manaraa.com

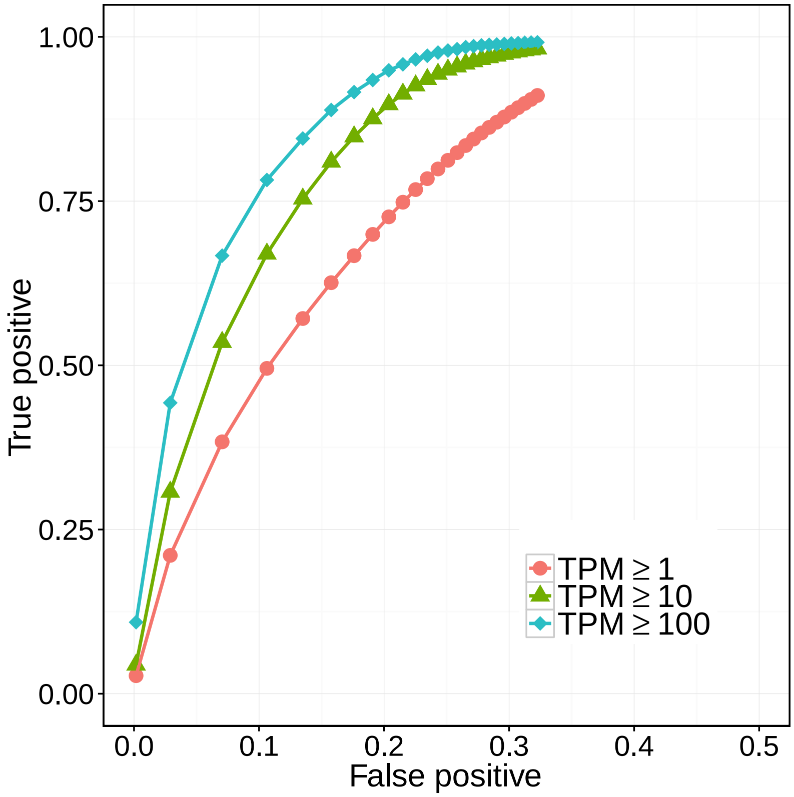


Figure 4.5: Accuracy in terms of ROC-curves for transcript sequence query results reported by SeqOthello-Exact as a function of expression levels.

Computing at the University of Kentucky. The servers are equipped with Dell R820, Quad Intel E5-4640 8-core (Sandy Bridge) @ 2.4 GHz and 512 GB/node of 1600 Mhz RAM. Sixteen threads were used during the index construction process. The query program was executed using single thread for both SBT and the three SeqOthello implementation variants. The disk usages, peak memory requirements, and construction/query run time for each tool are presented in Table 4.1. To compare different tools in a fair manner, all the results shown in the table were obtained from the run with the same settings for *k*-mer length (k=20) and θ (θ=0.8). All other parameters follow the default settings.

The original dataset requires about 1.3T disk space to store the raw fastq files. SeqOthello-Exact requires only 7 GB to store the entire index, achieving a compression ratio of 182:1,

www.manaraa.com

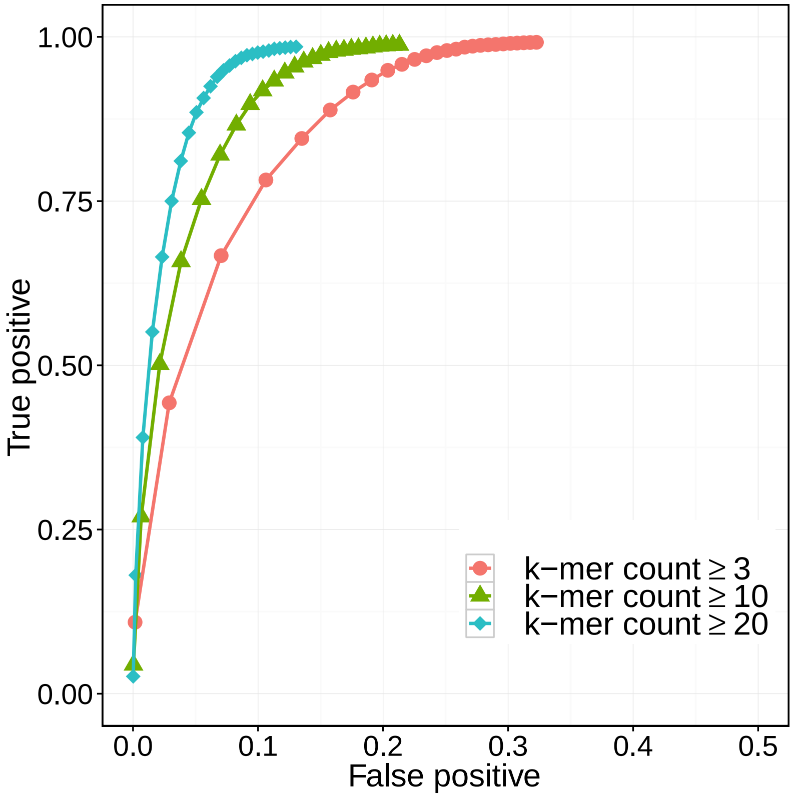


Figure 4.6: Accuracy in terms of ROC-curves for transcript sequence query results reported by SeqOthello-Exact as a function of *mincov*, min number of reads supporting a *k*-mer.

which is 7 times better than SBT. The SeqOthello-Compressed offers an even greater advantage with a compression ratio of 913:1. All methods requires significant amount of time to construct, SeqOthello in general requires under 4 hours, about half of the time needed by SBT. With current implementation of Othello, the peak memory of the construction is about 16 GB, the highest among all steps and much higher than SBT.

In terms of query performance, SeqOthello-Exact delivers 6 times speedup over SBT in single sequence query, 40 times in small batch query (1000 sequences) and over 100 speedup in large batch query of more than 36,076 sequences. All of these are achieved using less than 2 GB memory. The top layer of SeqOthello-Exact uses about 1 GB memory and each of the second layer bucket uses less than 512 MB memory; for SeqOthello-

المنارة للاستشارات
www.manaraa.com

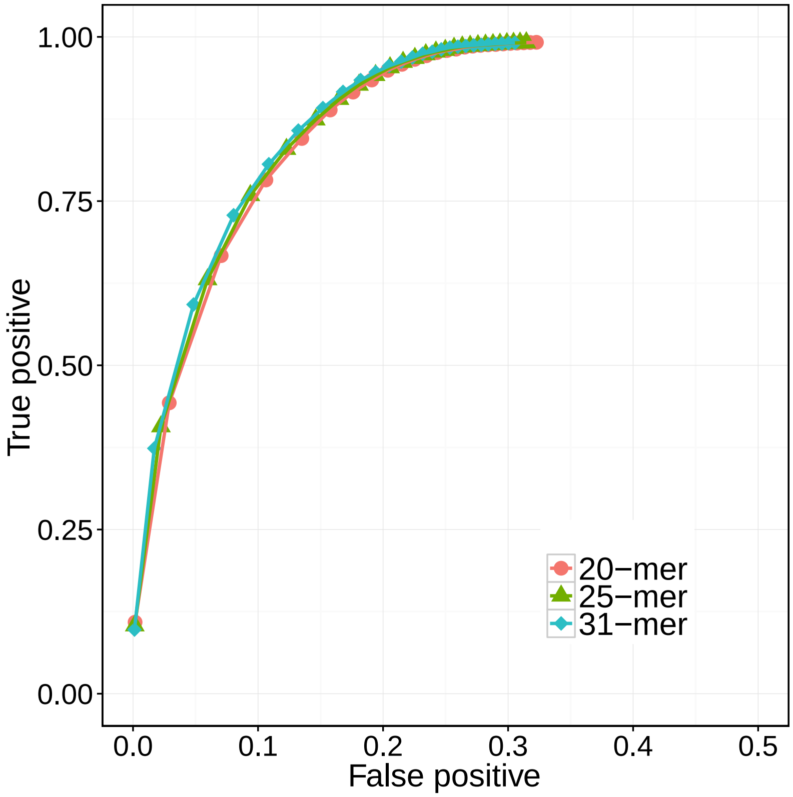


Figure 4.7: Accuracy in terms of ROC-curves for transcript sequence query results reported by SeqOthello-Exact as a funciton of *k*-mer length.

Compressed, the top layer Othello uses about 256 MB memory, and each of the second layer data structure uses less than 48 MB memory; for SeqOthello-UniversalCompressed, the top layer is an large array, which uses about 6.5 GB memory but the largest bucket in the second layer uses less than 48 MB memory. SeqOthello can be easily executed on today's standard PCs with 8GB RAM. SeqOthello-Compressed offers even greater advantages in its performance, since its total number of *k*-mers is much smaller than other variants. In comparison, the peak memory of SBT varies with the size of batch query. It requires only 0.2 GB memory for small query but it needs more than 9 GB memory for querying 36,076 transcripts.

Results on SeqOthello-UniversalCompressed provided us some insight on the upper

المنارة للاستشارات
www.manaraa.com

Table 4.1: Disk space, peak memory, and CPU time required by each tool for index construction and sequence query.

			SeqOthello		
		SBT	Exact	Compressed	UniversalCompressed
Disk usage (GB)		51	7	1.4	7.7
Compression ratio		25:1	182:1	913:1	165:1
Construction	Run time(min)	432	219	250	230
	Peak Memory(GB)	0.4	10.2	2.7	10.6
1 query	Run time(min)	1.2	0.2	0.1	0.2
	Peak Memory(GB)	0.2	1.8	0.4	7.7
1,000 queries	Run time(min)	8.2	0.2	0.1	0.3
	Peak Memory(GB)	0.4	1.8	0.4	7.7
36,076 queries	Run time(min)	199.6	1.7	2.1	2.4
	Peak Memory(GB)	8.9	1.8	0.4	7.7

bound of disk size, memory and query performance when the set of unique *k*-mers are much bigger than what we have in current data.

4.4 Gene fusion transcript query against TCGA Pan-Cancer RNA-seq datasets

The Cancer Genome Atlas (TCGA) (cancergenome.nih.gov) contains RNA-seq data of 10,113 tumor samples obtained from 9,215 cancer patients. The database allows researchers to detect and characterize novel transcriptomic alterations across 29 different cancer types in the GDC Legacy Archive (cancergenome.nih.gov). We have constructed a SeqOthello index, storing the occurrences of 1.47 billion 21-mers across all tumor samples. The preparation of k-mers averages 4 minutes per sample while the construction of SeqOthello on all samples took less than 9 hours. The index occupies only 76.6 GB of space, thus is portable for querying at different locations.

We use the SeqOthello index to conduct a survey of all gene-fusion events curated by TCGA Fusion Gene Database as of December 2017 [Yoshihara et al., 2015]. The database documented of 11,658 unique tier-1 fusion events from TCGA detected by PRADA [Torres-

المنارة للاستشارات

www.manaraa.com

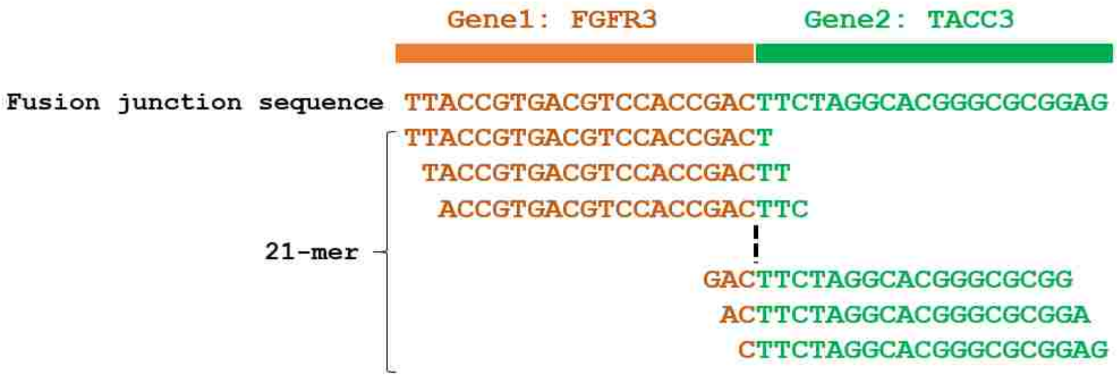


Figure 4.8: An illustration of fusion junction sequence constructed for fusion query using SeqOthello. Each fusion junction sequence consists of 20 bases from donor exon in one gene and 20 bases from acceptor exon in the other gene. Each 21-mer within this 40-base sequence spans the fusion junction. The query of a fusion sequence using SeqOthello may return a maximum of 20 k-mer hits for each RNA-seq Experiments indexed by SeqOthello.

García et al., 2014]. This represents 10,994 gene fusion pairs as multiple junctions might exist for one fusion pair. For each fusion junction, we construct a fusion sequence that will be used to query SeqOthello for its presence. The sequence consists of 20 bases from the donor exon and 20 bases from the acceptor exon, thereby guaranteeing that any 21-mer from the sequence will span the fusion junction (Figure 4.8).

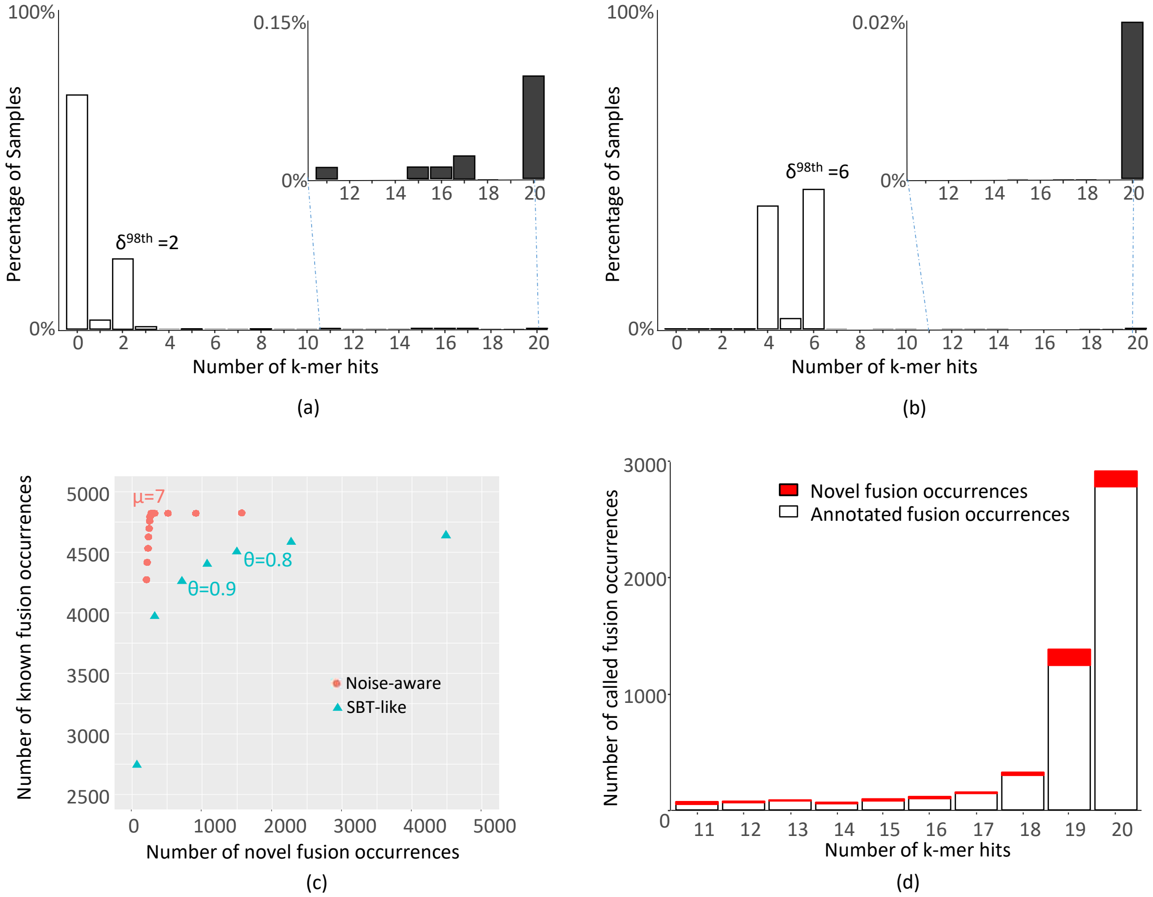
A SeqOthello query of a fusion sequence returns the number of k-mer hits in each sample. A simple method to determine the fusion occurrence in each sample can be done in SBT-like approach, where a minimum fraction of k-mer hits, θ, is required to call the presence. However, this technique yields lackluster sensitivity and specificity. Lowering θ permits fusion detection with fewer spanning reads, but may increase false-positive calls if the fusion junction sequence contains repetitive *k*-mers that are abundant in many samples. Instead of using a fixed threshold for all fusion calls, we develop a noise-aware approach. This approach first evaluates the background noise of the query result due to repetitive *k*-

www.manaraa.com

mers that are abundant in large fraction of samples, which can be detected leveraging the distribution of *k*-mer hits across TCGA tumor samples queried through SeqOthello. Two examples with different levels of repetitive *k*-mers are shown in Figure 4.9(a)(b). Assume true fusion occurs in less than 2% of TCGA samples as the highest occurrence rate reported so far is 0.953% out of all TCGA tumor samples by TMPRSS2-ERG27 (14.657% occurrence rate in prostate tumor samples). For each fusion, we estimate the level of background noise, σ, as the number of *k*-mer hits at the 98th percentile of the samples in the distribution of *k*-mer hits. We require additional number of k-mers, μ, beyond the background noise as evidence of expression to conclude the fusion occurrence in a sample. We compared the noise-aware approach with the θ-based SBT-like approach in recovering known fusion occurrences and in detecting unknown fusion occurrences. As shown in Figure 4.9(c), the noise-aware approach recovers more known fusions than the SBT-like approach without generating too many putative fusions that are likely to be false. Fusion occurrences called at $\mu = 7$ is used for further analysis as it renders the best sensitivity while being most conservative in generating candidates of novel occurrences. We then compared the distributions of actual *k*-mer hits of known fusion occurrences and novel occurrences in all the called fusion occurrences. The consistency between known and novel occurrences across the entire spectrum of *k*-mer hits further supports the validity of the noise-aware approach (Figure 4.9).

Under this method, we detect 92.7% of tier-1 fusion occurrences in TCGA Fusion Gene Database with at least 10 spanning reads reported by PRADA. Additionally, we identify 270 novel occurrences of fusion events across 17 tumor subtypes that are not identified by PRADA. We selected two fusion pairs with occurrences most inconsistent with current

المنارة للاستشارات
www.manaraa.com



(a) (b)

(c) (d)

Figure 4.9: An illustration of fusion calling criteria using SeqOthello's fusion junction sequence query results against TCGA pan-cancer RNA-seq data. (a) and (b) are examples of *k*-mer hit distribution as a result of fusion junction sequence query using SeqOthello. The presence of a small set of *k*-mers in large fraction of samples indicates background noise as a result of these *k*-mers being repetitive. For each fusion, we use σ^{98th}, the *k*-mer hit at 98th percentile as an estimation of background noise. (a) Histogram of *k*-mer hits by querying junction sequence spanning chr21:42880008-chr21:39956869 connecting gene pair TMPRSS2-ERG. The background noise is estimated at σ^{98th}=2. (b) Histogram of *k*-mer hits querying junction sequence spanning chr5:134688636-chr5:179991489 connecting gene pair H2AFY-CNOT6. The background noise is estimated at σ^{98th}=6. (c) The comparison of performance in recovering database-known fusion occurrences and detecting novel occurrences between noise-aware approach and SBT-like approach using θ-based containment query. Here μ is minimum number of *k*-mer hits required beyond the fusion-specific noise level used in noise-aware approach; θ is the minimum fraction of *k*-mer hits required to call the presence of a query as used in SBT containment query. (d) The distribution of the actual *k*-mer hits of all called fusion occurrences called with noise-aware approach.

المنارة للاستشارات
www.manaraa.com

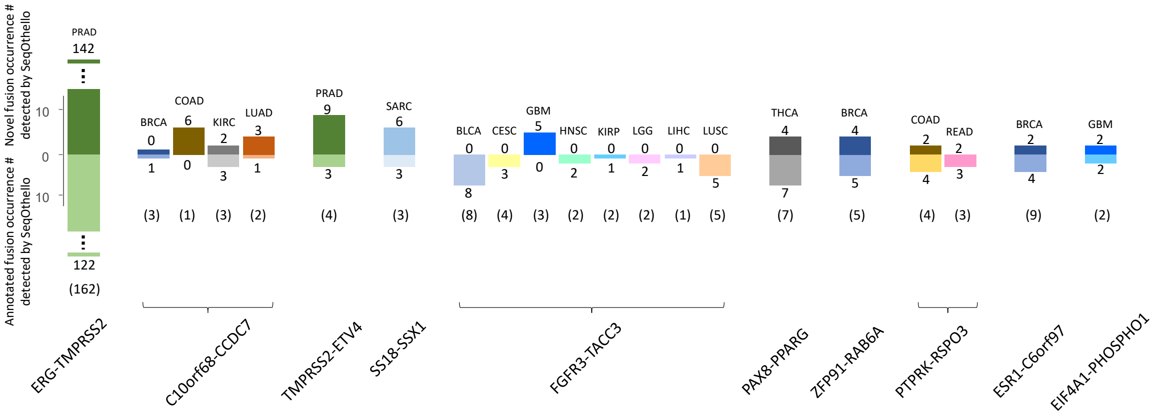


Figure 4.10: Top ten recurrent gene fusion detection results across 29 tumor types. Bar plots show occurrence number of top ten recurrent gene fusions detected by SeqOthello over different tumor types. Occurrences of each fusion on each tumor type are classified into novel occurrences (not in TCGA fusion database) and annotated occurrences (already curated by TCGA Gene Fusion Database).

curation for further validations: FGFR3-TACC3 in GBM samples (5 novel, 3 undetected) and ESR1-C6orf97 in BRCA samples (2 novel, 5 undetected). We were able to confirm all 7 novel fusion occurrences by the identification of at least 10 fusion spanning reads supporting each. For all undetected fusions, insufficient spanning reads were confirmed, which are consistent with low read support recorded in the database.

Figure 4.10 depicts the 10 novel, recurring fusions with greatest numbers of occurrences suggested by SeqOthello. Quite a few have doubled or even tripled the original recurring rates. Interestingly, all novel occurrences agree with the original fusion cancer-type classifications, rendering the chance of random occurrence negligible. This result corroborates their cancer specificity and supports the high precision of SeqOthellos query results. One example of this consistency is TMPRSS2-ERG, a clinical marker for prostate cancer. SeqOthello extracted 122 pre-identified occurrences of TMPRSS2-ERG and 142 novel occurrences, all from prostate cancer samples.

المنارة للاستشارات
www.manaraa.com

4.5 Conclusion

The tremendous growth of sequencing data has revolutionized and will continue to advance genomic research in an unprecedented speed. The problem on how to efficiently query against large-scale sequencing datasets as a whole emerges as a major challenge and its research is still in its infancy. Fortunately, major breakthrough, such as SBT, started to happen in the past two years.

We proposed and evaluated a novel indexing structure, SeqOthello, to support sequence query against large-scale transcriptome sequencing data.

Our experiment on 148 equine RNA-seq datasets demonstrated that SeqOthello achieved a compression ratio 182:1 of the raw sequencing data and were orders of magnitude faster than SBT, answering 36,076 queries under 2 minutes. It is notable that by using compressed *k*-mers, our approach achieved over 900:1 compression of the original data with quite comparable sensitivity and specificity.

Additionally, we constructed a SeqOthello index on the TCGA Pan-Cancer RNA-seq datasets, the latter totaling 54 TB in compressed fastq format. The SeqOthello index uses only 76.6 GB disk space, achieving a compression ratio of 700:1. Querying the index to assess the prevalence of 11,658 documented fusion events requires only five minutes on a standard desktop computer with 32 GB memory.

www.manaraa.com

Chapter 5 Efficient taxonomic classification of metagenomics read

5.1 Introduction

In Chapter 1, two categories of existing taxnomic classification methods are introduced. Our taxonomic classifier, MetaOthello, falls into the second category, where Kraken and Clark are representatives. Approaches in this category utilize indexing structures for *k*-mer matching. For example, Kraken indexes its lexicographically sorted *k*-mer database using a minimizer offset array, while Clark uses a hash table to store the mapping between a *k*-mer and its classification information. Both Kraken and Clark require computers with large memory to support the construction of their indexing structure (at least 170 GB RAM) and *k*-mer querying during classification (at least 70 GB RAM). Although there are variations of both algorithms with smaller memory footprints, they often afford significantly lower accuracy and much slower execution speed compared to the full version. For this reason, the ever-increasing amount of sequencing and reference genome data call for tools with better scalability in both memory and computation.

In this dissertation, we present a new algorithm, dubbed MetaOthello, for taxonomic classification of metagenomics sequencing reads. It employs a novel data structure, *l*-Othello, to support ultra-fast *k*-mer classification, achieving at least an order-of-magnitude improvement in speed over the state-of-the-art methods, Kraken and Clark, and three times faster than another recently published protein alignment-based tool Kaiju. In the meantime, MetaOthello also substantially reduces the memory footprint, typically requiring only one

المنارة للاستشارات
www.manaraa.com

third of the aforementioned methods. This modest memory requirement allows our algorithm to run on typical lab servers with 32 GB RAM, rendering it more accessible to biological researchers than those with memory requirements achievable only by supercomputers.

5.2 MetaOthello algorithm

5.2.1 K-mer taxon signatures

A k-mer is a length k subsequence of genomic sequences; for any sequence of length L, there exist a maximum of $L-k+1$ possible k-mers. Metagenomic reference material consists of one or more complete reference genomes belonging to an organism. Increasingly sophisticated sequencing techniques have permitted discovery of distinct reference genomes for a single species of organism, thereby capturing genomic variations that are often important to the functionality of the microbial species. The number of genomes (whether draft or complete) available as metagenomic reference material increases with each new discovery. If we consider each dataset as a collection of k-mers, a given taxon can be described by the set of k-mers present in the reference sequences belonging to its taxonomic subtree. The problem of classifying a metagenomic read thus simplifies to the identification of the taxon that best matches the set of k-mers associated with the target read. When k is sufficiently large (*e.g.,* $k \geqslant 20$), the majority of k-mers are unique to the species carrying them. These species-specific k-mers may serve as signatures, directly implicating the appropriate taxonomic classification. However, a significant proportion of k-mers is present in multiple species, making them unique only to higher-ranking taxa. In

www.manaraa.com

this work, we formalize the taxonomic specificity of a *k*-mer as the signature of a taxon: A *k*-mer is considered to be a signature of a taxon if (1) the *k*-mer does not appear in any genomic references belonging to ancestors or siblings of the target taxon, but only to sequences belonging to the taxon's subtree, and (2) the *k*-mer is not a signature of any lower-ranked taxon in the subtree. Equivalently, the taxon evincing a *k*-mer signature is the lowest common ancestor (LCA) of all species in the taxonomy whose reference genomes contain that *k*-mer.

In this way, as illustrated in Fig.5.1A, the set of all *k*-mers present in the genomic references of a taxonomy can be divided into disjoint collections, each of which contains the set of signature *k*-mers belonging to a single node in the taxonomy tree. Formally, let S be the set of all *k*-mers present in genomic references annotated by the taxonomy and let $T = \{1, 2, \cdots, |T|\}$ be the taxa (nodes) present in the taxonomy. Then S can be divided into $|T|$ disjoint sets, $S = \{S_0, S_1, \cdots, S_t, \cdots, S_{(|T|-1)}\}$, where for any node $t \in T$, S_t corresponds to the set of *k*-mer signatures belonging to taxon t. Thus, there exists a mapping, $g : S \rightarrow T$, such that $g(s) = t$ if the *k*-mer, $s \in S$, is a signature of the taxon, $t \in T$. In MetaOthello, this mapping is supported by the hashing classifier Othello [Yu et al., 2017].

5.2.2 Taxonomic classification of sequencing reads

As illustrated in Fig.5.1B, given any sequencing read, our algorithm iterates over each *k*-mer from the beginning of the read and, for each *k*-mer, retrieves the taxon to which it is specific using Othello. Taxonomic classification of the read is determined by assembling the taxa for all *k*-mers in the read. The classification is straightforward when all *k*-mers indicate the same taxon, but this is not often the case. Disparate taxa are considered to be

www.manaraa.com

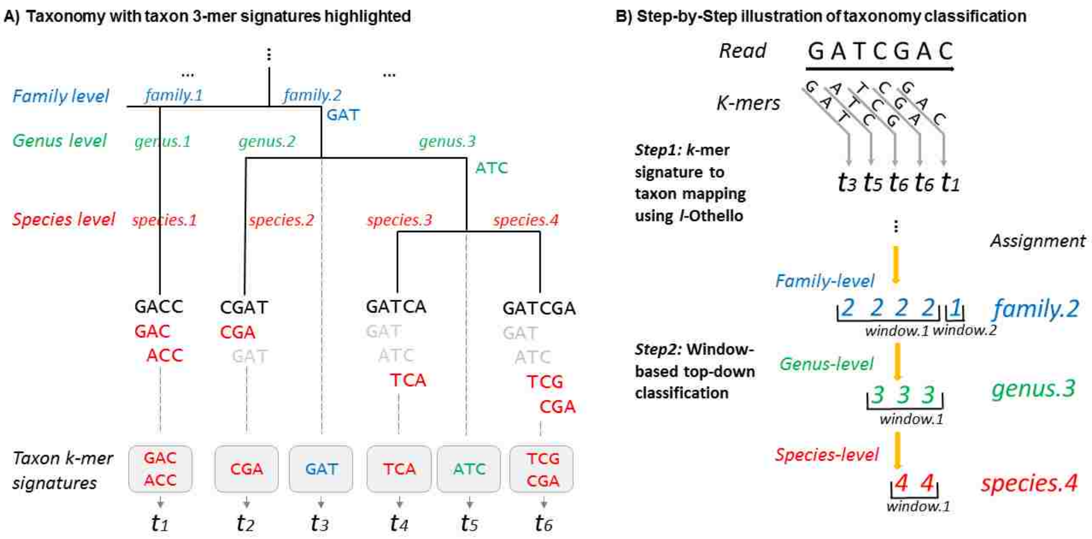


Figure 5.1: Illustration of MetaOthello algorithm. A: an example of taxonomy with reference sequences in the leaf nodes. 3-mers that are signatures to each node are highlighted in red colors with different shades. B: a two-step approach in read classification.

consistent if they belong to the same path in the taxonomy, meaning that one assignment is the higher rank of the other. When these taxa belong to different branches, they represent conflicting information. The issue is further complicated by the possibility of false taxonomic information returned from querying alien *k*-mers, where the *k*-mer in the read does not appear in any of the reference sequences.

To tackle this challenge, we have designed a window-based classification approach. A window is defined as a sequence of consecutive *k*-mers that are assigned to the same taxon of a given level. The window-based approach guards against false-positive assignments due to alien *k*-mers.

Additionally, each window corresponds to a maximum read subsequence that matches the reference sequences. Thus, the longer the window, the longer the subsequence match, and the less likely the match is random. In comparison, other algorithms such as Kraken and Clark count the total number of *k*-mer matches, regardless of their spatial distribution

المنارة للاستشارات
www.manaraa.com

across the read.

If multiple taxon windows are available, MetaOthello scores each of them using the summed squares of window sizes as in the following formula; the taxon with the maximum score will be selected:

$$Score(t) = \sum (w_i^t)^2$$

where w_i^t denotes the number of k-mers in the i_{th} window classified to taxon t.

A k-mer signature belonging to a taxon is also specific to its higher-ranking taxa, so at higher taxonomic ranks, there exist more k-mers to distinguish a taxon from its siblings. Thus, longer k-mer windows and more accurate classifications are expected at higher taxonomic ranks. Under this assumption, a “top-down” strategy is adopted during read classification. Given a read sequence, MetaOthello starts the classification at the top rank and continues the classification down the ranks until there does not exist a sufficiently large k-mer window supporting the level. Based on the k-mer distribution in each taxon, MetaOthello establishes a threshold on minimum window-size when the classification on that taxon requires. Theorem 1 shows that the minimum window size threshold can be precomputed for each taxon prior to read classification. The minimum window size required for a taxon is determined by the probability of an alien k-mer query on l-Othello returning a taxon rooted in t and the acceptable false-positive rate. The larger the size of the taxon subtree, the higher the probability that a random alien k-mer may match to t and thus the longer the window required for reliable classification. Additionally, a larger window size will be required in order to lower the false-positive rate.

Theorem 1 *Given a user-defined false-positive rate* λ *and the total read number* M*, the*

المنارة للاستشارات

www.manaraa.com

minimum window-size threshold required for a taxon t *can be computed as* $\log_{p(t)} \frac{\lambda}{(1-\lambda)M}$, *where* p_t *denotes the probability that an alien k-mer query on l-Othello returns a value in the taxon subtree with root* t*.*

The proof is presented in our published journal article [Liu et al., 2018]. For example, when t is a genus-level node, supposing $l = 12$, then $p(t) \sim (1+7)2^{-l} = \frac{1}{256}$. Given 10 million reads and suppose $\lambda = 0.001$, then $\log_{p(t)} \frac{\lambda}{(1-\lambda)M} = 3.42$, and only windows larger than three will be taken into consideration when determining the read assignment.

5.3 Experimental results on simulated datasets

5.3.1 Classification accuracy and taxon-specific *k*-mer relative abundance

Accurate classification of a read to a taxon is largely dependent upon the presence of *k*-mer signatures. Thus the abundance of these signature *k*-mers (*i.e.,* the proportion of taxon-specific *k*-mers among all *k*-mers present in the reference sequences for the taxon) becomes an important indicator of the capability of our algorithm. Thus we first investigate the correlation between classification accuracy and the relative abundance of taxon-specific *k*-mers.

Classification accuracy is computed as the fraction of reads assigned correctly. Using a next-generation sequencing (NGS) read simulator called ART [Huang et al., 2012], we simulated 10,000 reads for each of 2,629 reference genomes in the NCBI RefSeq bacterial genome database, for a total of 26,290,000 reads. The database is available at `ftp://ftp.ncbi.nih.gov/genomes/archive/old_refseq/Bacteria/`. Each read is paired-end and of length 100 bp with a fragment size of 250 bp, generated using the de-

المنارة للاستشارات

www.manaraa.com

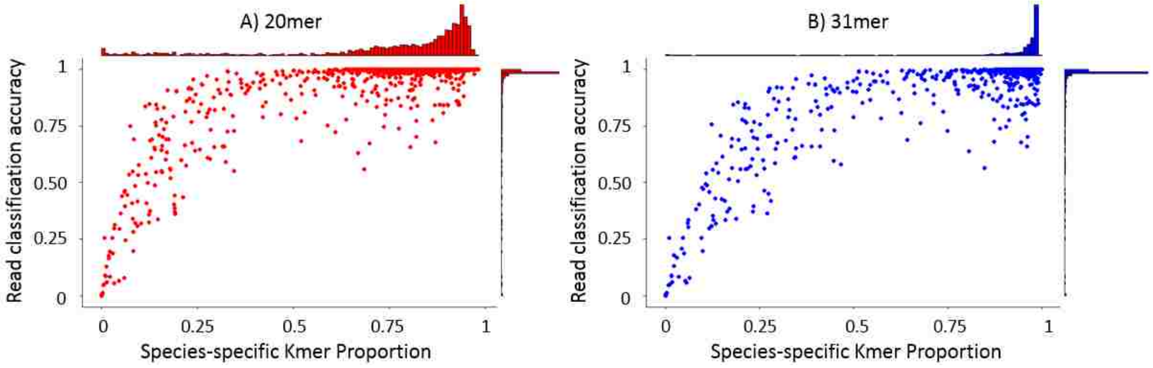


Figure 5.2: Correlation between species-specific *k*-mer signatures and classification accuracy when *k*=20 (A) and *k*=31 (B). In each panel, the central figure depicts the correlation between species-specific *k*-mer proportion and read-classification accuracy for all species; the top histogram shows the distribution of species as a function of species-specific *k*-mer proportion, and the right histogram shows the distribution of classification accuracy for all species.

fault error profile for the HiSeq platform. Figure 5.2 shows the read-classification accuracy for each species as a function of species-specific *k*-mer proportion, where read assignments were generated by the MetaOthello algorithm using 20-mers and 31-mers, respectively. The scatter plot for either *k*-mer size demonstrates that the vast majority of all species manifest more than 50% 20-mers that are species-specific, and almost all species have 75% species-specific 31-mers. Although in general more species-specific *k*-mers afford better classification accuracy, these results suggest that the presence of 50% or more species-specific *k*-mers affords suitably high classification accuracy, thereby demonstrating the utility of *k*-mer signatures in classifying metagenomic reads.

5.3.2 Comparison with the state-of-the-art tools

We assess the performance of MetaOthello in comparison to three of state-of-the-art tools: Kraken (version 0.10.5 beta), Clark (version 1.2.3), and Kaiju (version 1.4.4). Besides the newly published tool Kaiju, Kraken and Clark were chosen based on the recommen-

المنارة للاستشارات
www.manaraa.com

dation of a recent benchmarking paper [Lindgreen et al., 2016], which evaluated 14 tools using six datasets and subsequently declared Kraken and Clark the best performers over Genometa [Davenport et al., 2012], GOTTCHA [Freitas et al., 2015], LMAT [Ames et al., 2013], MEGAN [Huson et al., 2007, 2011], MG-RAST [Glass and Meyer, 2011], the One Codex webserver, taxator-tk [Dröge et al., 2015], MetaPhlAn [Segata et al., 2012], MetaPhyler [Liu et al., 2010], mOTU [Sunagawa et al., 2013], and QIIME [Caporaso et al., 2010]. The comparison was benchmarked against three publicly available datasets: HiSeq, MiSeq, and SimBA5. The same datasets have been used multiple times to evaluate a number of metagenomic classification tools, including Kraken in previous studies [Wood and Salzberg, 2014]. All tools were executed using the same reference database (NCBI RefSeq as of October 1st, 2016), and all other parameters follow the default settings.

Classification Accuracy

We first compare classification accuracy. Three different *k*-mer lengths (20-mer, 25-mer, and 31-mer) are used to assess the performance relationship with *k*-mer size for Kraken, Clark, and MetaOthello; Kaiju is not a *k*-mer-based algorithm. To facilitate the comparison and to mimic the sequencing data generated by current platforms, we discarded reads shorter than 36 bp and those whose taxon is not included in the reference taxonomy.

Reads were classified by each algorithm at three taxonomic levels: phylum, genus, and species. MetaOthello, Kraken, and Kaiju were able to classify reads at the three levels simultaneously, while Clark required three separate runs to conduct similar classifications. Thus for Clark, results from these runs were merged for the purpose of direct comparison. Precision and sensitivity were computed at each of the three classification levels. Precision

المنارة للاستشارات
www.manaraa.com

is defined as the ratio between correctly assigned reads and the total number of reads in an assignment; sensitivity is calculated as the fraction of total reads assigned correctly. F1 score (*i.e.,* the harmonic mean of precision and sensitivity) was also calculated to quantify the balance between these two metrics. Results of the comparison are shown in Table 5.1.

In general, longer *k*-mers enhance the precision of read classification but decrease sensitivity, as observed in MetaOthello, Kraken, and Clark. Within each dataset, the overall winner (in bold) was the one with highest F1-score when considering across all three *k*-mer sizes. In phylum-level classification, MetaOthello outperforms the other algorithms in all three datasets all using 20-mers. At both genus and species levels, MetaOthello exhibits the best performance on two out of the three datasets using either 25-mers or 20-mers. Kraken performs the best in the remaining comparisons, followed closely by MetaOthello in both cases. In general, Kaiju delivered much lower (20% to 30%) sensitivity compared to the other three tools, due to its lack of capability in classifying reads from non-protein coding regions.

Runtime and Memory

Speed benchmarks were performed using the servers from Lipscomb High-Performance Computing at the University of Kentucky. The servers are equipped with Dell R820, Quad Intel E5-4640 8-core (Sandy Bridge) @ 2.4 GHz and 512 GB/node of 1600 Mhz RAM. Each algorithm was executed using eight threads and *k*-mer lengths as specified previously; all other parameters follow the default settings. The speed for each tool is presented in Figure 5.3. In general, MetaOthello achieved the highest processing speed, clocking roughly 1 billion bases per minute. This figure represents an order-of-magnitude improve-

المنارة للاستشارات
www.manaraa.com

Table 5.1: Accuracy of read taxonomic classification in terms of precision, sensitivity, and F_1 score.

			Phylum	Genus	Species
			Prec / Sens / F1	Prec / Sens / F1	Prec / Sens / F1
HiSeq	MetaOthello	20mer	98.4 / 95.0 / **.967**	97.2 / 92.5 / .948	82.0 / 69.4 / .751
		25mer	99.4 / 92.2 / .957	99.1 / 91.2 / .950	84.2 / 69.1 / **.760**
		31mer	99.4 / 89.0 / .939	**99.3** / 88.2 / .934	85.7 / 68.0 / .758
	Kraken	20mer	97.8 / 94.8 / .963	96.1 / 92.0 / .940	80.2 / 69.2 / .743
		25mer	**99.7** / 92.3 / .959	99.1 / 91.4 / **.951**	83.7 / 69.4 / .759
		31mer	**99.7** / 88.3 / .937	**99.3** / 87.6 / .931	85.4 / 67.6 / .745
	Clark	20mer	97.7 / **95.5** / .966	95.1 / **92.6** / .939	76.4 / **69.5** / .728
		25mer	**99.7** / 92.1 / .958	99.1 / 91.2 / .950	83.5 / 69.2 / .757
		31mer	**99.7** / 88.8 / .940	**99.3** / 88.1 / .934	85.4 / 68.0 / .757
	Kaiju		99.4 / 68.7 / .812	98.6 / 65.1 / .785	**89.2** / 34.7 / .499
MiSeq	MetaOthello	20mer	99.2 / 97.5 / **.983**	96.2 / 92.2 / .942	91.8 / 78.6 / .846
		25mer	99.6 / 95.4 / .975	97.4 / 91.4 / **.943**	93.0 / 78.3 / **.850**
		31mer	99.6 / 92.9 / .961	98.0 / 89.7 / .937	93.8 / 77.2 / .847
	Kraken	20mer	99.0 / 97.5 / **.983**	95.8 / 92.2 / .939	91.0 / **78.9** / .845
		25mer	99.8 / 95.1 / .974	97.4 / 91.2 / .942	92.7 / 78.3 / .849
		31mer	**99.9** / 92.3 / .960	98.0 / 89.3 / .935	93.6 / 76.8 / .844
	Clark	20mer	98.8 / **97.8** / **.983**	94.4 / **92.5** / .934	86.9 / 78.8 / .826
		25mer	99.8 / 95.2 / .975	97.1 / 91.5 / .942	91.9 / 78.5 / .847
		31mer	**99.9** / 92.7 / .962	98.0 / 89.8 / .937	93.4 / 77.3 / .846
	Kaiju		99.5 / 75.7 / .860	**98.5** / 68.0 / .805	**95.2** / 40.6 / .570
simBA5	MetaOthello	20mer	**99.9** / **99.7** / **.998**	99.6 / 95.8 / **.977**	99.3 / 84.2 / .911
		25mer	**99.9** / 98.2 / .990	99.8 / 94.6 / .971	99.5 / 83.1 / .906
		31mer	99.5 / 92.2 / .957	99.5 / 88.7 / .938	99.4 / 77.9 / .873
	Kraken	20mer	99.8 / 99.5 / .996	99.4 / **95.9** / .976	98.8 / **84.6** / **.912**
		25mer	**99.9** / 98.5 / .992	99.8 / 95.0 / .974	99.5 / 83.8 / .909
		31mer	**99.9** / 94.2 / .970	**99.9** / 90.9 / .952	**99.7** / 80.0 / .887
	Clark	20mer	99.8 / 99.6 / .997	98.5 / 95.8 / .971	94.4 / 84.2 / .890
		25mer	**99.9** / 98.4 / .992	99.8 / 94.8 / .973	99.4 / 83.4 / .907
		31mer	99.9 / 93.5 / .966	**99.9** / 90.2 / .948	**99.7** / 79.2 / .883
	Kaiju		99.6 / 75.6 / .860	97.9 / 65.9 / .788	96.5 / 46.7 / .630

Table 5.2: The proportion of reads classified into the top-five genera by each algorithm using different k-mer lengths.

	MetaOthello			Kraken			Clark			Kaiju
k-mer length	20	25	31	20	25	31	20	25	31	
Streptococcus	14.3	13.2	12.1	14.6	13.3	12.2	14.3	13.2	12.0	10.1
Haemophilus	7.00	6.67	5.94	7.09	7.13	6.01	6.95	6.41	5.89	4.77
Prevotella	5.62	4.89	4.27	5.77	5.01	4.27	5.59	4.57	4.19	7.65
Veillonella	3.32	2.67	1.92	3.44	2.93	1.98	3.23	2.59	1.89	5.17
Neisseria	2.22	1.86	1.53	2.28	1.94	1.55	2.21	1.78	1.51	3.11

ment over Kraken and Clark, the two most-rapid state-of-the-art tools within the category of alignment-free classifiers. Impressively, the high speed does not entail a compromise in

www.manaraa.com

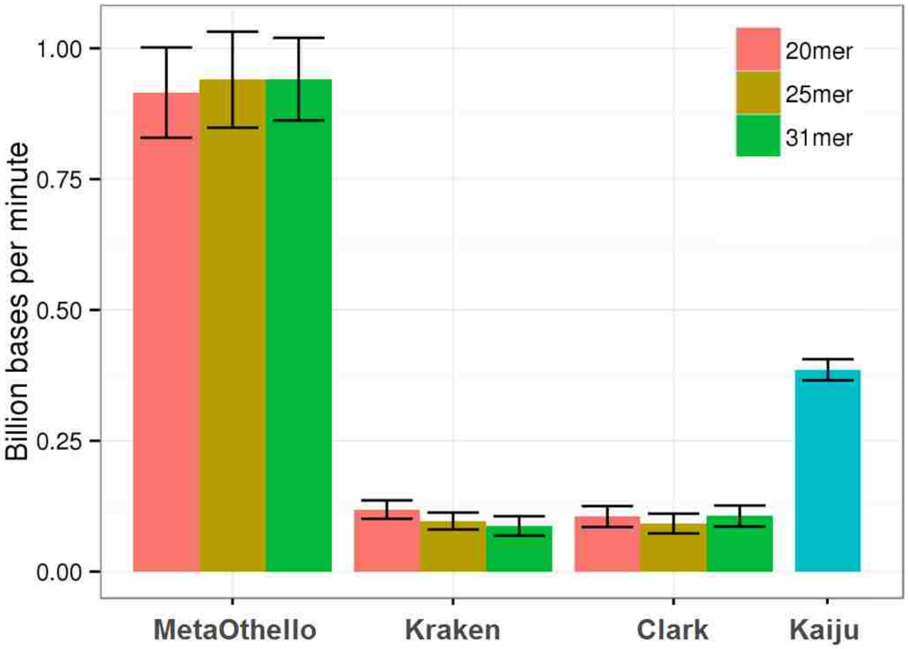


Figure 5.3: Billion bases processed per minute by each tool with three *k*-mer length settings using 8 threads.

the memory requirement. MetaOthello only consumes about one-third (peak memory 27 GB) the RAM required by Kraken and Clark (peak memory 73 GB).

The construction of the MetaOthello index from the NCBI RefSeq bacterial genome sequence database requires roughly 6 hours with peak memory usage up to 40 GB using 16 threads. In contrast, Kraken and Clark used 164 GB and 120 GB respectively for index construction but both finished under 4 hours with 16 threads.

In summary, MetaOthello achieves a significant speedup with much smaller memory footprint in comparison with Kraken and Clark while delivering competitive or even superior performance in classification accuracy. While Kaiju is relatively scalable, it suffers from low sensitivity in classification.

المنارة للاستشارات
www.manaraa.com

5.4 Metagenomic classification of real datasets

5.4.1 Human Microbiome Project data

To assess the performance of MetaOthello relative to Kraken, Clark, and Kaiju on real datasets, the three algorithms were run on sequencing data from three saliva samples (NCBI SRA accessions: SRS015055, SRS019120, and SRS014468) used in the Human Microbiome Project [Human Microbiome Project, 2012]. We ran the three *k*-mer-based algorithms at each of three different *k*-mer length settings (20-mer, 25-mer, and 31-mer) as with the simulated data. The three samples were analyzed separately, and the results were pooled together to assess the relative abundances of species. The top five most-abundant genera are presented in Table 5.2. The four tools reported the same five most-abundant genera: Streptococcus, Haemophilus, Prevotella, Veillonella, and Neisserlia, all of which are known to be associated with human saliva. Interestingly, although the absolute abundance (*i.e.,* the fraction of total reads assigned to a given genus) varies with *k*-mer size, the relative abundances remain stable except for Kaiju. The false-positive rate, however, cannot be assessed in this case.

5.4.2 Zika virus detection

The 2015-2016 Zika fever outbreak caused by Zika virus are spread over Americas, the Pacific, and Asia. Increasing metagenomic NGS data has been generated to detect the Zika virus and track their evolution [Cunha et al., 2016] or study the correlation between clinical presentation and infection [Sardi et al., 2016]. However, this analysis is greatly limited by the lack of reference sequence diversity in current genome database. For example, in

المنارة للاستشارات
www.manaraa.com

NCBI/RefSeq genome database as of Oct 30th, 2016, there is only one reference genome sequence, but 47 strain genome sequences have been identified [Sardi et al., 2016]. Hence, besides using the regular reference genome *k*-mer database, we also explored an alternative approach where we included *k*-mers directly from Zika virus sequencing data as part of the reference.

To assess MetaOthello's performance and how additional *k*-mers from sequencing data help on Zika read detection, we applied MetaOthello on three Zika virus genome sequencing datasets from NCBI SRA with accessions SRR3332512, SRR3332513, and SRR3332514. Those three datasets are sequenced from animal samples at day 4 post infection with Zika virus. Firstly, we ran MetaOthello with reference genome 20-mer only database, and successfully identified 74-82% reads in each dataset as Zika reads. Next, we added 20-mers from the sequencing reads of raw Zika virus sample (SRR3332511) into database and repeated the analysis. In order to avoid the effects of noises and contamination, we only incorporated 20-mers with a minimum frequency of 10,000 in this experiment. As shown in Table 5.3, we do observe an increase up to 12% in terms of sensitivity. In order to determine the correctness of those additionally classified Zika reads, we randomly selected 100 of them and aligned to ENA sequences using blastn through EBI webserver (https://www.ebi.ac.uk/Tools/sss/ncbiblast/nucleotide.html). All of them were successfully aligned to at least one Zika strain with a minimum identity of 95%. We believe this approach provides a viable solution to fill the gap between limited reference genome sequences in a regular database and the highly diverse genomic datasets in real-world applications.

المنارة للاستشارات
www.manaraa.com

Table 5.3: Data sizes and numbers of reads classified as Zika reads by MetaOthello with two alternative databases.

Data	Total read #	Detected Zika read # by MetaOthello	
		RefSeq genome *k*-mer database	RefSeq genome + sequencing data *k*-mer database
SRR3332512	591,839	442,850 (74.83%)	510,579 (86.27%)
SRR3332513	483,758	392,360 (81.11%)	430,574 (89.01%)
SRR3332514	629,483	495,612 (78.73%)	567,522 (90.16%)

5.5 Conclusion

In this chapter, we present MetaOthello, a novel metagenomic sequencing read classifier. MetaOthello leverages a novel probabilistic hashing structure, *l*-Othello, to conduct taxonomic classification using taxon-specific *k*-mer signatures. The algorithm delivers ultra-fast and memory-efficient solutions to *k*-mer-based taxonomic classification. Within the set of alignment-free approaches, MetaOthello achieves an order-of-magnitude improvement in classification speed relative to the fastest algorithms, Kraken and Clark, while reducing the RAM requirement from 70G to 27G. MetaOthello exhibits high sensitivity and precision competitive with Kraken and Clark, and in most cases achieves a better balance between the sensitivity and specificity (as quantified by F1 score). It is also three times faster than the protein alignment-based method Kaiju and delivers much higher classification accuracy.

المنارة للاستشارات
www.manaraa.com

Chapter 6 Conclusion

Recent years, we have witnessed how the Next Generation Sequencing (NGS) technology advances biomedical research. Each run of NGS will generate tens to hundreds of millions of sequencing reads. Accurate and scalable approaches for the analysis of those large-scale datasets remain as unmet needs. This dissertation presents four novel computational methods for NGS data analysis, three of them focus on RNA-seq data analysis, and the other one is designed for metagenomics sequencing data analysis.

RNA-seq plays a central role in profiling the transcriptome and is used to a vast range of applications such as differential expression analysis on both gene and transcript levels, novel isoform prediction, genomic variants calling, RNA editing detection and so on. In most of those applications, especially those reference-based ones, a critical and fundamental step is to correctly discover the alignment of each RNA-seq read on the reference genome. In Chapter 2, we introduce MapSplice3, a comprehensive reference-based spliced aligner. MapSplice3 adopts a two-phase alignment strategy. It identifies candidate mapping positions for each read sequence in the first phase and learns transcriptome context from the intermediate alignments. Context refers to transcriptome features including genome-wide splicing structure and SNPs. In the second phase, MapSplice3 performs context-aware alignment, which maps previously unmapped or partially mapped reads to the learned context and reference genome simultaneously. Our experiments show that the context-aware alignment improves both the performance of splice junction discovery and mapping completeness of read sequences. Additionally, MapSplice3 shows great sensitiv-

المنارة للاستشارات
www.manaraa.com

ity and accuracy in gene fusion and circular RNA detection.

Although MapSplice3 has already performed quite well in read alignment, utilizing a standard reference genome as the only template can lead to mapping bias which overestimates reference allelic ratio, as well as the failure to detect personal splice junctions created by splice site mutations. In Chapter 3, we describe iMapSplice, an individualized RNA-seq read aligner. By taking into account both the reference genome sequence and personal genomic variants, iMapSplice performs an unbiased mapping of reads carrying either the reference or alternative base. Besides general improvements in read alignment and splice junction discovery, iMapSplice significantly alleviates the allelic ratio bias, which is a common deficiency for sequencing read aligners. Additionally, considering genomic variants breaks the dependency on splice site dinucleotide motifs in the reference genome, and enables iMapSplice to detect more than 1,000 personal canonical splice junctions created through splice site mutations in 74 human datasets. Performance-wise, iMapSplice is a lightweight approach with minimum overhead in both storage and running time compared to other alignment methods that are also capable of considering individual variants during the mapping process.

Conventional approaches including MapSplice3 and iMapSplice provide great capacities for transcriptome profiling but still are not efficient for global surveys of given transcriptome features (such as gene fusions) against large-scale datasets. For example, to discover gene fusions in TCGA Pan-cancer RNA-seq data (10,113 datasets in total), even using the most-efficient existing fusion detection algorithm, it is estimated to require 785 days of computation. To tackle this challenge, we developed SeqOthello, an efficient sequencing data indexing structure. It supports super-fast sequence query against large-scale

المنارة للاستشارات
www.manaraa.com

transcriptome sequencing data. The data structure contains two-layer hierarchy, with the first layer mapping genomic sequence *k*-mers to their frequency buckets and the second layer mapping *k*-mer to their occurrence maps across all samples. The mapping at each node can be reduced into a many-to-one mapping problem between (hundreds of) millions of *k*-mers and up to millions of disjoint categories. These mappings are efficiently implemented using the data structure Othello. For the same application described above (gene fusion survey in TCGA data), it is able to return the presence information of 11,658 previously documented gene fusion events for each dataset in five minutes. At the same time, it achieves great compression ratio (700:1).

Besides the computational methods for RNA-seq data analysis, we describe a novel taxonomic classifier for metagenomics sequencing data in Chapter 5. The software, MetaOthello, builds upon taxon-specific *k*-mer signatures to support direct read assignment to any level in the taxonomy. It employees the same probabilistic hashing classifier Othello, to support efficient query of a taxon using its k-mer signatures. MetaOthello achieves an order-of-magnitude faster speed compared to the current most efficient programs and consumes only one-third of the RAM at the same time. Also, MetaOthello is capable of conducting a hierarchical top-down taxonomic classification and delivers performance competitive to, if not better than, other algorithms in both sensitivity and specificity as validated by benchmarking on a variety of datasets.

All software packages of the methods described in this dissertation are open-source, released, and freely available to the research community.

As the technology keeps evolving and the prices keep dropping, NGS will be more and more widely adopted and applied in biological and medical studies and applications. It is

www.manaraa.com

believed that the aggregation of NGS data will eventually provide a much more comprehensive picture of molecular biology. To analyze such large-scale data, besides enormous computational resources, accurate and scalable computational methods are in high demand. We expect that the methods presented in this dissertation are great fits for the analysis of NGS data, and will help scientists gain insights from their experiments.

Copyright© Xinan Liu, 2018.

www.manaraa.com

Bibliography

Daehwan Kim, Ben Langmead, and Steven L Salzberg. HISAT: a fast spliced aligner with low memory requirements. *Nature methods*, 12(4):357–60, 2015. ISSN 1548-7105. doi: 10.1038/nmeth.3317. URL `http://www.ncbi.nlm.nih.gov/pubmed/25751142{%}5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4655817`.

Manfred G.; Grabherr, Nir Brian J. Haas, Moran Yassour Joshua Z. Levin, Dawn A. Thompson, Ido Amit, Xian Adiconis, Lin Fan, Raktima Raychowdhury, Qiandong Zeng, Zehua Chen, Evan Mauceli, Nir Hacohen, Andreas Gnirke, Nicholas Rhind, Federica di Palma, Bruce W., Friedman, and Aviv Regev. Trinity: reconstructing a full-length transcriptome without a genome from RNA-Seq data. *Nature Biotechnology*, 29(7):644–652, 2013. ISSN 1546-1696. doi: 10.1038/nbt.1883.Trinity.

Cole Trapnell, Brian A. Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J. Van Baren, Steven L. Salzberg, Barbara J. Wold, and Lior Pachter. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5):511–515, 2010. ISSN 10870156. doi: 10.1038/nbt.1621.

Bo Li and Colin N. Dewey. RSEM: Accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, 12, 2011a. ISSN 14712105. doi: 10.1186/1471-2105-12-323.

www.manaraa.com

Adam Roberts, Cole Trapnell, Julie Donaghey, John L. Rinn, and Lior Pachter. Improving RNA-Seq expression estimates by correcting for fragment bias. *Genome Biology*, 12(3), 2011. ISSN 14747596. doi: 10.1186/gb-2011-12-3-r22.

Yin Hu, Yan Huang, Ying Du, Christian F. Orellana, Darshan Singh, Amy R. Johnson, Anaïs Monroy, Pei Fen Kuan, Scott M. Hammond, Liza Makowski, Scott H. Randell, Derek Y. Chiang, D. Neil Hayes, Corbin Jones, Yufeng Liu, Jan F. Prins, and Jinze Liu. DiffSplice: The genome-wide detection of differential splicing events with RNA-seq. *Nucleic Acids Research*, 41(2), 2013. ISSN 03051048. doi: 10.1093/nar/gks1026.

Ning Leng, John A. Dawson, James A. Thomson, Victor Ruotti, Anna I. Rissman, Bart M.G. Smits, Jill D. Haag, Michael N. Gould, Ron M. Stewart, and Christina Kendziorski. EBSeq: An empirical Bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics*, 29(8):1035–1043, 2013. ISSN 13674803. doi: 10.1093/bioinformatics/btt087.

M. I. Love, Simon Anders, and Wolfgang Huber. *Differential analysis of count data - the DESeq2 package*, volume 15. 2014. ISBN 0000000000. doi: 110.1186/s13059-014-0550-8. URL `http://biorxiv.org/lookup/doi/10.1101/002832{%}5Cnhttp://dx.doi.org/10.1186/s13059-014-0550-8`.

Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics (Oxford, England)*, 26(1):139–40, 2010. ISSN 1367-4811. doi: 10.1093/bioinformatics/

المنارة للاستشارات
www.manaraa.com

btp616. URL `http://www.ncbi.nlm.nih.gov/pubmed/19910308{%}5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2796818`.

Cole Trapnell, Lior Pachter, and Steven L. Salzberg. TopHat: Discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9):1105–1111, 2009. ISSN 13674803. doi: 10.1093/bioinformatics/btp120.

Daehwan Kim, Geo Pertea, Cole Trapnell, Harold Pimentel, Ryan Kelley, and Steven L Salzberg. TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biology*, 14(4):R36, 2013. ISSN 1465-6914. doi: 10.1186/gb-2013-14-4-r36. URL `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=pubmed{&}cmd=Retrieve{&}dopt=AbstractPlus{&}list{_}uids=23618408{%}5Cnhttp://genomebiology.com/content/pdf/gb-2013-14-4-r36.pdf`.

Kai Wang, Darshan Singh, Zheng Zeng, Stephen J. Coleman, Yan Huang, Gleb L. Savich, Xiaping He, Piotr Mieczkowski, Sara A. Grimm, Charles M. Perou, James N. MacLeod, Derek Y. Chiang, Jan F. Prins, and Jinze Liu. MapSplice: Accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Research*, 38(18), 2010. ISSN 03051048. doi: 10.1093/nar/gkq622.

Thomas D. Wu and Serban Nacu. Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics*, 26(7):873–881, 2010. ISSN 13674803. doi: 10.1093/bioinformatics/btq057.

Alexander Dobin, Carrie A. Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali

المنارة للاستشارات

www.manaraa.com

Jha, Philippe Batut, Mark Chaisson, and Thomas R. Gingeras. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, 2013. ISSN 13674803. doi: 10.1093/bioinformatics/bts635.

Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990. ISSN 00222836. doi: 10.1016/S0022-2836(05)80360-2.

W James Kent. BLAT The BLAST -Like Alignment Tool. *Genome Research*, 12:656–664, 2002. ISSN 1088-9051. doi: 10.1101/gr.229202.

Ben Langmead and Steven L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–359, 2012. ISSN 15487091. doi: 10.1038/nmeth.1923.

Jie Wu, Olga Anczuków, Adrian R. Krainer, Michael Q. Zhang, and Chaolin Zhang. OLego: Fast and sensitive mapping of spliced mRNA-Seq reads using small seeds. *Nucleic Acids Research*, 41(10):5149–5163, 2013. ISSN 03051048. doi: 10.1093/nar/gkt216.

Yang Liao, Gordon K. Smyth, and Wei Shi. The Subread aligner: Fast, accurate and scalable read mapping by seed-and-vote. *Nucleic Acids Research*, 41(10), 2013. ISSN 03051048. doi: 10.1093/nar/gkt214.

M Salson, T Lecroq, M Léonard, and L Mouchard. Burrows-Wheeler Transform. *Theoretical Computer Science accepted 2009*, pages 13–25, 2008. doi: 10.1007/978-3-642-27848-8. URL `http://www.stringology.org/event/2008/psc08p02{_}presentation.pdf`.

www.manaraa.com

Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009. ISSN 13674803. doi: 10.1093/bioinformatics/btp324.

Steve Hoffmann, Christian Otto, Stefan Kurtz, Cynthia M. Sharma, Philipp Khaitovich, Jörg Vogel, Peter F. Stadler, and Jörg Hackermüller. Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Computational Biology*, 5(9), 2009. ISSN 1553734X. doi: 10.1371/journal.pcbi.1000502.

Pär G. Engström, Tamara Steijger, Botond Sipos, Gregory R. Grant, André Kahles, Gunnar Rätsch, Nick Goldman, Tim J. Hubbard, Jennifer Harrow, Roderic Guigó, Paul Bertone, Tyler Alioto, Jonas Behr, Regina Bohnert, Davide Campagna, Carrie A. Davis, Alexander Dobin, Thomas R. Gingeras, Géraldine Jean, Peter Kosarev, Sheng Li, Jinze Liu, Christopher E. Mason, Vladimir Molodtsov, Zemin Ning, Hannes Ponstingl, Jan F. Prins, Paolo Ribeca, Igor Seledtsov, Victor Solovyev, Giorgio Valle, Nicola Vitulo, Kai Wang, Thomas D. Wu, and Georg Zeller. Systematic evaluation of spliced alignment programs for RNA-seq data. *Nature Methods*, 10(12):1185–1191, 2013. ISSN 15487091. doi: 10.1038/nmeth.2722.

Gregory R. Grant, Michael H. Farkas, Angel D. Pizarro, Nicholas F. Lahens, Jonathan Schug, Brian P. Brunk, Christian J. Stoeckert, John B. Hogenesch, and Eric A. Pierce. Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM). *Bioinformatics*, 27(18):2518–2528, 2011. ISSN 13674803. doi: 10.1093/bioinformatics/btr427.

المنارة للاستشارات
www.manaraa.com

Scott A. Tomlins, Bharathi Laxman, Sooryanarayana Varambally, Xuhong Cao, Jindan Yu, Beth E. Helgeson, Qi Cao, John R. Prensner, Mark A. Rubin, Rajal B. Shah, Rohit Mehra, and Arul M. Chinnaiyan. Role of the TMPRSS2-ERG Gene Fusion in Prostate Cancer. *Neoplasia*, 10(2):177–IN9, 2008. ISSN 14765586. doi: 10.1593/neo.07822. URL `http://linkinghub.elsevier.com/retrieve/pii/S1476558608800644`.

Daehwan Kim and Steven L. Salzberg. TopHat-Fusion: An algorithm for discovery of novel fusion transcripts. *Genome Biology*, 12(8), 2011. ISSN 14747596. doi: 10.1186/gb-2011-12-8-r72.

Andrew McPherson, Fereydoun Hormozdiari, Abdalnasser Zayed, Ryan Giuliany, Gavin Ha, Mark G F Sun, Malachi Griffith, Alireza Moussavi, Janine Senz, Nataliya Melnyk, Marina Pacheco, Marco A. Marra, Martin Hirst, Torsten O. Nielsen, S. Cenk Sahinalp, David Huntsman, and Sohrab P. Shah. Defuse: An algorithm for gene fusion discovery in tumor rna-seq data. *PLoS Computational Biology*, 7(5), 2011. ISSN 1553734X. doi: 10.1371/journal.pcbi.1001138.

Wenlong Jia, Kunlong Qiu, Minghui He, Pengfei Song, Quan Zhou, Feng Zhou, Yuan Yu, Dandan Zhu, Michael L. Nickerson, Shengqing Wan, Xiangke Liao, Xiaoqian Zhu, Shaoliang Peng, Yingrui Li, Jun Wang, and Guangwu Guo. SOAPfuse: An algorithm for identifying fusion transcripts from paired-end RNA-Seq data. *Genome Biology*, 14 (2), 2013. ISSN 1474760X. doi: 10.1186/gb-2013-14-2-r12.

Julia Salzman, Charles Gawad, Peter Lincoln Wang, Norman Lacayo, and Patrick O. Brown. Circular RNAs are the predominant transcript isoform from hundreds of hu-

المنارة للاستشارات
www.manaraa.com

man genes in diverse cell types. *PLoS ONE*, 7(2), 2012. ISSN 19326203. doi: 10.1371/journal.pone.0030733.

Yuan Gao, Jinfeng Wang, and Fangqing Zhao. CIRI: An efficient and unbiased algorithm for de novo circular RNA identification. *Genome Biology*, 16(1), 2015. ISSN 1474760X. doi: 10.1186/s13059-014-0571-3.

Linda Szabo, Robert Morey, Nathan J. Palpant, Peter L. Wang, Nastaran Afari, Chuan Jiang, Mana M. Parast, Charles E. Murry, Louise C. Laurent, and Julia Salzman. Erratum to: Statistically based splicing detection reveals neural enrichment and tissue-specific induction of circular RNA during human fetal development [Genome Biology, 16, (2016) (126)], DOI: 10.1186/s13059-015-0690-5, 2016. ISSN 1474760X.

Scott D. Kahn. On the future of genomic data, 2011. ISSN 00368075.

Christopher Wilks, Phani Gaddipati, Abhinav Nellore, and Ben Langmead. Snaptron: Querying splicing patterns across tens of thousands of RNA-seq samples. *Bioinformatics*, 34(1):114–116, 2018. ISSN 14602059. doi: 10.1093/bioinformatics/btx547.

Brad Solomon and Carl Kingsford. Fast search of thousands of short-read sequencing experiments. *Nature Biotechnology*, 34(3):300–302, 2016. ISSN 15461696. doi: 10.1038/nbt.3442.

Curtis Huttenhower and Human Microbiome Project Consortium. Structure, function and diversity of the healthy human microbiome. *Nature*, 486(7402): 207–14, 2012. ISSN 1476-4687. doi: 10.1038/nature11234. URL `http:`

المنارة للاستشارات
www.manaraa.com

`//www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3564958{&}tool=pmcentrez{&}rendertype=abstract.`

J. Craig Venter, Karin Remington, John F. Heidelberg, Aaron L. Halpern, Doug Rusch, Jonathan A. Eisen, Dongying Wu, Ian Paulsen, Karen E. Nelson, William Nelson, Derrick E. Fouts, Samuel Levy, Anthony H. Knap, Michael W. Lomas, Ken Nealson, Owen White, Jeremy Peterson, Jeff Hoffman, Rachel Parsons, Holly Baden-Tillson, Cynthia Pfannkoch, Yu Hui Rogers, and Hamilton O. Smith. Environmental Genome Shotgun Sequencing of the Sargasso Sea. *Science*, 304(5667):66–74, 2004. ISSN 00368075. doi: 10.1126/science.1093857.

Gene W. Tyson, Jarrod Chapman, Philip Hugenholtz, Eric E. Allen, Rachna J. Ram, Paul M. Richardson, Victor V. Solovyev, Edward M. Rubin, Daniel S. Rokhsar, and Jillian F. Banfield. Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature*, 428(6978):37–43, 2004. ISSN 00280836. doi: 10.1038/nature02340.

Daniel H. Huson, Alexander F. Auch, Ji Qi, and Stephan C. Schuster. MEGAN analysis of metagenomic data. *Genome Research*, 17(3):377–386, 2007. ISSN 10889051. doi: 10.1101/gr.5969107.

Arthur Brady and Steven L. Salzberg. Phymm and PhymmBL: Metagenomic phylogenetic classification with interpolated Markov models. *Nature Methods*, 6(9):673–676, 2009. ISSN 15487091. doi: 10.1038/nmeth.1358.

Gail L. Rosen, Erin R. Reichenberger, and Aaron M. Rosenfeld. NBC: The naïve Bayes

المنارة للاستشارات

www.manaraa.com

classification tool webserver for taxonomic classification of metagenomic reads. *Bioinformatics*, 27(1):127–129, 2011. ISSN 13674803. doi: 10.1093/bioinformatics/btq619.

Daehwan Kim, Li Song, Florian P. Breitwieser, and Steven L. Salzberg. Centrifuge: Rapid and sensitive classification of metagenomic sequences. *Genome Research*, 26(12):1721–1729, 2016. ISSN 15495469. doi: 10.1101/gr.210641.116.

Peter Menzel and Anders Krogh. Kaiju : Fast and sensitive taxonomic classification for metagenomics. *bioRxiv*, 7:1–9, 2015. ISSN 2041-1723. doi: 10.1101/031229. URL `http://dx.doi.org/10.1038/ncomms11257`.

Sasha K. Ames, David a. Hysom, Shea N. Gardner, G. Scott Lloyd, Maya B. Gokhale, and Jonathan E. Allen. Scalable metagenomic taxonomy classification using a reference genome database. *Bioinformatics (Oxford, England)*, 29(18):2253–2260, 2013. ISSN 13674803. doi: 10.1093/bioinformatics/btt389.

Derrick E. Wood and Steven L. Salzberg. Kraken: Ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15(3), 2014. ISSN 1474760X. doi: 10.1186/gb-2014-15-3-r46.

Ye Yu, Djamal Belazzougui, Chen Qian, and Qin Zhang. A concise forwarding information base for scalable and fast name lookups. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, pages 1–10. IEEE, oct 2017. ISBN 978-1-5090-6501-1. doi: 10.1109/ICNP.2017.8117530. URL `http://ieeexplore.ieee.org/document/8117530/`.

المنارة للاستشارات
www.manaraa.com

Anthony M. Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 2014. ISSN 14602059. doi: 10.1093/bioinformatics/btu170.

Silvia Liu, Wei Hsiang Tsai, Ying Ding, Rui Chen, Zhou Fang, Zhiguang Huo, Sunghwan Kim, Tianzhou Ma, Ting Yu Chang, Nolan Michael Priedigkeit, Adrian V. Lee, Jianhua Luo, Hsei Wei Wang, I. Fang Chung, and George C. Tseng. Comprehensive evaluation of fusion transcript detection algorithms and a meta-caller to combine top performing methods in paired-end RNA-seq data. *Nucleic Acids Research*, 44(5), 2015. ISSN 13624962. doi: 10.1093/nar/gkv1234.

The 1000 Genomes Project Consortium. A global reference for human genetic variation, 2015. ISSN 0028-0836. URL `http://www.nature.com/nature/journal/v526/n7571/full/nature15393.html{%}5Cnhttp://www.nature.com/nature/journal/v526/n7571/pdf/nature15393.pdf`.

Débora Y C Brandt, Vitor R C Aguiar, Bárbara D Bitarello, Kelly Nunes, Jérôme Goudet, and Diogo Meyer. Mapping Bias Overestimates Reference Allele Frequencies at the HLA Genes in the 1000 Genomes Project Phase I Data. *G3 (Bethesda, Md.)*, 5(5):931–41, 2015. ISSN 2160-1836. doi: 10.1534/g3.114.015784. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4426377{&}tool=pmcentrez{&}rendertype=abstract`.

Alison M Meynert, Morad Ansari, David R FitzPatrick, and Martin S Taylor. Variant detection sensitivity and biases in whole genome and exome sequenc-

المنارة للاستشارات
www.manaraa.com

ing. *BMC bioinformatics*, 15(1):247, 2014. ISSN 1471-2105. doi: 10.1186/1471-2105-15-247. URL `http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-15-247`.

Stephane E. Castel, Ami Levy-Moonshine, Pejman Mohammadi, Eric Banks, and Tuuli Lappalainen. Tools and best practices for data processing in allelic expression analysis. *Genome Biology*, 16(1):195, 2015. ISSN 1465-6906. doi: 10.1186/s13059-015-0762-6. URL `http://genomebiology.com/2015/16/1/195{%}5Cnhttp://genomebiology.com/2015/16/1/195/abstract{%}5Cnhttp://www.genomebiology.com/2015/16/1/195{%}5Cnhttp://www.genomebiology.com/content/pdf/s13059-015-0762-6.pdf`.

Amanda J. Ward and Thomas A. Cooper. The pathobiology of splicing, 2010. ISSN 00223417.

Jamal Tazi, Nadia Bakkour, and Stefan Stamm. Alternative splicing and disease, 2009. ISSN 09254439.

Fan Zhang, Mu Wang, Tran Michael, and Renee Drabier. Novel alternative splicing isoform biomarkers identification from high-throughput plasma proteomics profiling of breast cancer. *BMC systems biology*, 7 Suppl 5 (Suppl 5):S8, 2013. ISSN 1752-0509. doi: 10.1186/1752-0509-7-S5-S8. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4028860{&}tool=pmcentrez{&}rendertype=abstract`.

Shayna Stein, Zhi Xiang Lu, Emad Bahrami-Samani, Juw Won Park, and Yi Xing.

المنارة للاستشارات

www.manaraa.com

Discover hidden splicing variations by mapping personal transcriptomes to personal genomes. *Nucleic Acids Research*, 43(22):10612–10622, 2015. ISSN 13624962. doi: 10.1093/nar/gkv1099.

Tuuli Lappalainen, Michael Sammeth, Marc R Friedländer, Peter a C 't Hoen, Jean Monlong, Manuel a Rivas, Mar Gonzàlez-Porta, Natalja Kurbatova, Thasso Griebel, Pedro G Ferreira, Matthias Barann, Thomas Wieland, Liliana Greger, Maarten van Iterson, Jonas Almlöf, Paolo Ribeca, Irina Pulyakhina, Daniela Esser, Thomas Giger, Andrew Tikhonov, Marc Sultan, Gabrielle Bertier, Daniel G MacArthur, Monkol Lek, Esther Lizano, Henk P J Buermans, Ismael Padioleau, Thomas Schwarzmayr, Olof Karlberg, Halit Ongen, Helena Kilpinen, Sergi Beltran, Marta Gut, Katja Kahlem, Vyacheslav Amstislavskiy, Oliver Stegle, Matti Pirinen, Stephen B Montgomery, Peter Donnelly, Mark I McCarthy, Paul Flicek, Tim M Strom, Hans Lehrach, Stefan Schreiber, Ralf Sudbrak, Angel Carracedo, Stylianos E Antonarakis, Robert Häsler, Ann-Christine Syvänen, Gert-Jan van Ommen, Alvis Brazma, Thomas Meitinger, Philip Rosenstiel, Roderic Guigó, Ivo G Gut, Xavier Estivill, and Emmanouil T Dermitzakis. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, 501(7468):506–11, 2013. ISSN 1476-4687. doi: 10.1038/nature12531. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3918453{&}tool=pmcentrez{&}rendertype=abstract`.

Jennifer Harrow, Adam Frankish, Jose M. Gonzalez, Electra Tapanari, Mark Diekhans, Felix Kokocinski, Bronwen L. Aken, Daniel Barrell, Amonida Zadissa, Stephen Searle, If Barnes, Alexandra Bignell, Veronika Boychenko, Toby Hunt, Mike Kay, Gaurab

www.manaraa.com

Mukherjee, Jeena Rajan, Gloria Despacio-Reyes, Gary Saunders, Charles Steward, Rachel Harte, Michael Lin, Cédric Howald, Andrea Tanzer, Thomas Derrien, Jacqueline Chrast, Nathalie Walters, Suganthi Balasubramanian, Baikang Pei, Michael Tress, Jose Manuel Rodriguez, Iakes Ezkurdia, Jeltje Van Baren, Michael Brent, David Haussler, Manolis Kellis, Alfonso Valencia, Alexandre Reymond, Mark Gerstein, Roderic Guigó, and Tim J. Hubbard. GENCODE: The reference human genome annotation for the ENCODE project. *Genome Research*, 22(9):1760–1774, 2012. ISSN 10889051. doi: 10.1101/gr.135350.111.

Steven C. Munger, Narayanan Raghupathy, Kwangbom Choi, Allen K. Simons, Daniel M. Gatti, Douglas A. Hinerfeld, Karen L. Svenson, Mark P. Keller, Alan D. Attie, Matthew A. Hibbs, Joel H. Graber, Elissa J. Chesler, and Gary A. Churchill. RNA-Seq alignment to individualized genomes improves transcript abundance estimates in multiparent populations. *Genetics*, 198(1):59–73, 2014. ISSN 19432631. doi: 10.1534/genetics.114.165886.

Chen Sun, Robert S. Harris, Rayan Chikhi, and Paul Medvedev. Allsome sequence bloom trees. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10229 LNCS, pages 272–286, 2017. ISBN 9783319569697. doi: 10.1007/978-3-319-56970-3_17.

Brad Solomon and Carl Kingsford. Improved search of large transcriptomic sequencing databases using split sequence bloom trees. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioin-*

المنارة للاستشارات
www.manaraa.com

formatics), volume 10229 LNCS, pages 257–271, 2017. ISBN 9783319569697. doi: 10.1007/978-3-319-56970-3_16.

Guillaume Marçais and Carl Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770, 2011. ISSN 13674803. doi: 10.1093/bioinformatics/btr011.

Kin Fai Au, Jason G. Underwood, Lawrence Lee, and Wing Hung Wong. Improving PacBio Long Read Accuracy by Short Read Alignment. *PLoS ONE*, 7(10), 2012. ISSN 19326203. doi: 10.1371/journal.pone.0046679.

Jared T. Simpson, Kim Wong, Shaun D. Jackman, Jacqueline E. Schein, Steven J.M. Jones, and Inanç Birol. ABySS: A parallel assembler for short read sequence data. *Genome Research*, 19(6):1117–1123, 2009. ISSN 10889051. doi: 10.1101/gr.089532.108.

Emma Adam. DIFFERENTIAL GENE EXPRESSION IN EQUINE CARTILAGINOUS TISSUES AND INDUCED CHONDROCYTES. *Theses and Dissertations–Veterinary Science*, jan 2016. doi: http://dx.doi.org/10.13023/ETD.2016.343. URL `https://uknowledge.uky.edu/gluck{_}etds/25`.

Bo Li and Colin N Dewey. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC bioinformatics*, 12(1):323, 2011b. ISSN 1471-2105. doi: 10.1186/1471-2105-12-323. URL `http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-12-323`.

K. Yoshihara, Q. Wang, W. Torres-Garcia, S. Zheng, R. Vegesna, H. Kim, and R. G.W. Ver-

المنارة للاستشارات
www.manaraa.com

haak. The landscape and therapeutic relevance of cancer-associated transcript fusions. *Oncogene*, 34(37):4845–4854, 2015. ISSN 14765594. doi: 10.1038/onc.2014.406.

Wandaliz Torres-García, Siyuan Zheng, Andrey Sivachenko, Rahulsimham Vegesna, Qianghu Wang, Rong Yao, Michael F Berger, John N Weinstein, Gad Getz, and Roel G W Verhaak. PRADA: pipeline for RNA sequencing data analysis. *Bioinformatics (Oxford, England)*, 30(15):2224–6, 2014. ISSN 1367-4811. doi: 10.1093/bioinformatics/btu169. URL `http://bioinformatics.oxfordjournals.org/content/30/15/2224.long`.

Xinan Liu, Ye Yu, Jinze Liu, Jinpeng Liu, Corrine F. Elliott, and Chen Qian. A novel data structure to support ultra-fast taxonomic classification of metagenomic sequences with k-mer signatures. In *Bioinformatics*, volume 34, pages 171–178, 2018. doi: 10.1093/bioinformatics/btx432.

Weichun Huang, Leping Li, Jason R. Myers, and Gabor T. Marth. ART: A next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012. ISSN 13674803. doi: 10.1093/bioinformatics/btr708.

Stinus Lindgreen, Karen L. Adair, and Paul P. Gardner. An evaluation of the accuracy and speed of metagenome analysis tools. *Scientific Reports*, 6, 2016. ISSN 20452322. doi: 10.1038/srep19233.

Colin F. Davenport, Jens Neugebauer, Nils Beckmann, Benedikt Friedrich, Burim Kameri, Svea Kokott, Malte Paetow, Björn Siekmann, Matthias Wieding-Drewes, Markus Wienhöfer, Stefan Wolf, Burkhard Tümmler, Volker Ahlers, and Frauke Sprengel.

المنارة للاستشارات
www.manaraa.com

Genometa - A fast and accurate classifier for short metagenomic shotgun reads. *PLoS ONE*, 7(8), 2012. ISSN 19326203. doi: 10.1371/journal.pone.0041224.

Tracey A.llen K. Freitas, Po E. Li, Matthew B. Scholz, and Patrick S.G. Chain. Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. *Nucleic acids research*, 43(10):e69, 2015. ISSN 13624962. doi: 10.1093/nar/gkv180.

Daniel H Huson, Suparna Mitra, Hans-Joachim Ruscheweyh, Nico Weber, and Stephan C Schuster. Integrative analysis of environmental sequences using MEGAN4. *Genome research*, 21(9):1552–60, 2011. ISSN 1549-5469. doi: 10.1101/gr.120618.111. URL `http://www.ncbi.nlm.nih.gov/pubmed/21690186{%}5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3166839`.

Elizabeth M. Glass and Folker Meyer. The Metagenomics RAST Server: A Public Resource for the Automatic Phylogenetic and Functional Analysis of Metagenomes. In *Handbook of Molecular Microbial Ecology I: Metagenomics and Complementary Approaches*, pages 325–331. 2011. ISBN 9780470644799. doi: 10.1002/9781118010518.ch37.

J. Dröge, I. Gregor, and A. C. McHardy. Taxator-tk: Precise taxonomic assignment of metagenomes by fast approximation of evolutionary neighborhoods. *Bioinformatics*, 31 (6):817–824, 2015. ISSN 14602059. doi: 10.1093/bioinformatics/btu745.

Nicola Segata, Levi Waldron, Annalisa Ballarini, Vagheesh Narasimhan, Olivier Jousson, and Curtis Huttenhower. Metagenomic microbial community profiling using unique

المنارة للاستشارات
www.manaraa.com

clade-specific marker genes. *Nature Methods*, 9(8):811–814, 2012. ISSN 15487091. doi: 10.1038/nmeth.2066.

Bo Liu, Theodore Gibbons, Mohammad Ghodsi, and Mihai Pop. MetaPhyler: Taxonomic profiling for metagenomic sequences. In *Proceedings - 2010 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2010*, pages 95–100, 2010. ISBN 9781424483075. doi: 10.1109/BIBM.2010.5706544.

Shinichi Sunagawa, Daniel R. Mende, Georg Zeller, Fernando Izquierdo-Carrasco, Simon A. Berger, Jens Roat Kultima, Luis Pedro Coelho, Manimozhiyan Arumugam, Julien Tap, Henrik Bjørn Nielsen, Simon Rasmussen, Søren Brunak, Oluf Pedersen, Francisco Guarner, Willem M. De Vos, Jun Wang, Junhua Li, Joël Doré, S. Dusko Ehrlich, Alexandros Stamatakis, and Peer Bork. Metagenomic species profiling using universal phylogenetic marker genes. *Nature Methods*, 10(12):1196–1199, 2013. ISSN 15487091. doi: 10.1038/nmeth.2693.

J. Gregory Caporaso, Justin Kuczynski, Jesse Stombaugh, Kyle Bittinger, Frederic D. Bushman, Elizabeth K. Costello, Noah Fierer, Antonio Gonzalez Pa, Julia K. Goodrich, Jeffrey I. Gordon, Gavin A. Huttley, Scott T. Kelley, Dan Knights, Jeremy E. Koenig, Ruth E. Ley, Catherine A. Lozupone, Daniel McDonald, Brian D. Muegge, Meg Pirrung, Jens Reeder, Joel R. Sevinsky, Peter J. Turnbaugh, William A. Walters, Jeremy Widmann, Tanya Yatsunenko, Jesse Zaneveld, and Rob Knight. QIIME allows analysis of high-throughput community sequencing data, 2010. ISSN 15487091.

Consortium Human Microbiome Project. A framework for human microbiome research.

www.manaraa.com

Nature, 486(7402):215–221, 2012. ISSN 1476-4687. doi: 10.1038/nature11209. URL `http://www.ncbi.nlm.nih.gov/pubmed/22699610`.

Mariana Sequetin Cunha, Danillo Lucas Alves Esposito, Iray Maria Rocco, Adriana Yurika Maeda, Fernanda Gisele Silva Vasami, Juliana Silva Nogueira, Renato Pereira de Souza, Akemi Suzuki, Marcelo Addas-Carvalho, Maria De Lourdes Barjas-Castro, Mariângela Ribeiro Resende, Raquel Silveira Bello Stucchi, Ilka De Fátima Santana Ferreira Boin, Gizelda Katz, Rodrigo Nogueira Angerami, and Benedito Antonio Lopes da Fonseca. First Complete Genome Sequence of Zika Virus (Flaviviridae, Flavivirus) from an Autochthonous Transmission in Brazil. *Genome announcements*, 4(2):2015–2016, 2016. ISSN 2169-8287. doi: 10.1128/genomeA.00032-16. URL `http://www.ncbi.nlm.nih.gov/pubmed/26941134`.

Silvia I. Sardi, Sneha Somasekar, Samia N. Naccache, Antonio C. Bandeira, Laura B. Tauro, Gubio S. Campos, and Charles Y. Chiub. Coinfections of zika and chikungunya viruses in bahia, Brazil, identified by metagenomic next-generation sequencing. *Journal of Clinical Microbiology*, 54(9):2348–2353, 2016. ISSN 1098660X. doi: 10.1128/JCM.00877-16.

المنارة للاستشارات
www.manaraa.com

Vita

Xinan Liu

- Education
  - B.S. in Computer Science Aug. 2008 - July 2012

    Harbin Institute of Technology, Harbin, Heilongjiang, China
- Employment History
  - Research Scientist I, Bioinformatics, Mar. 2018 - Present

    Gilead Sciences, Inc., Foster City, CA, USA
  - Research Assistant, Aug. 2012 - May. 2017 & Aug. 2017 - Mar. 2018

    University of Kentucky, Lexington, KY, USA
  - Data Scientist Intern, May 2017 - Aug. 2017

    iCarbonX Co., Ltd., Shenzhen, Guangdong, China
- Publications
  - Xinan Liu, Ye Yu, Jinpeng Liu, Corrine F. Elliot, Chen Qian, and Jinze Liu. "A novel data structure to support ultra-fast taxonomic classification of metagenomics sequences with k-mer signatures." Bioinformatics, 2017.
  - Yi Zhang, Xinan Liu, James N. MacLeod, and Jinze Liu. "DeepSplice: Deep classification of novel splice junctions revealed by RNA-seq." Proceedings of

المنارة للاستشارات
www.manaraa.com

the IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2016.

  - Xinan Liu, James N. MacLeod, and Jinze Liu. “iMapSplie: a lightweight and personalized RNA-seq alignment approach to improve transcriptome profiling.” Under review as of April 4, 2018.
  - Ye Yu, Jinpeng Liu, Xinan Liu, Yi Zhang, Eamonn Magner, Erik Lehnert, Chen Qian, and Jinze Liu. “SeqOthello: Query over RNA-seq experiments at scale.” Under review as of April 4, 2018

- Conference Presentations
  - Xinan Liu, Ye Yu, James N. MacLeod, Chen Qian and Jinze Liu. “SeqOthello: a novel indexing structure to support accurate and scalable query over large scale sequencing read datasets.” The 25th Conference on Intelligent Systems for molecular Biology, Special Interest Group on High Throughput Sequencing Algorithms and Applications (ISMB HiTSeq), Prague, Czech Republic, July, 2017.
  - Xinan Liu, Ye Yu, Jinpeng Liu, Corrine F. Elliot, Chen Qian, and Jinze Liu. “MetaOthello: a novel data structure to support ultra-fast taxonomic classification of metagenomics sequences with k-mer signatures.” The 21st Annual International Conference on Research in Computational Molecular Biology, Satellite Workshop on Massively Parallel Sequencing (RECOMB-SEQ), Hong Kong, China, May, 2017.

المنارة للاستشارات
www.manaraa.com

  - Yi Zhang, Xinan Liu, James N. MacLeod, and Jinze Liu. “DeepSplice: Deep classification of novel splice junctions revealed by RNA-seq.” IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Shenzhen, China, December, 2016.
  - Xinan Liu, James N. MacLeod, and Jinze Liu. “iMapSplie: a lightweight and personalized RNA-seq alignment approach to improve transcriptome profiling.” The 24th Conference on Intelligent Systems for molecular Biology, Special Interest Group on High Throughput Sequencing Algorithms and Applications (ISMB HiTSeq), Orlando, USA, July, 2016.
  - Emma N. Adam, James N. MacLeod, Yi Zhang, Xinan Liu, and Jinze Liu. “Comparative Chondrogenic Potential of Equine Fetal Progenitor Cells and Adult Mesenchymal Stem Cells.” The Plant and Animal Genome XXIV Conference (PAG), San Diego, USA, January, 2016.

- Software
  - MapSplice3: a comprehensive reference-based RNA-seq read aligner.
    https://github.com/xa6xa6/MapSplice3
  - iMapSplice: an individualized RNA-seq read aligner.
    https://github.com/xa6xa6/iMapSplice
  - MetaOthello: a taxonomic classifier of metagenomics sequences.
    https://github.com/xa6xa6/metaOthello

www.manaraa.com