

Summer 8-2-2012

## RNA CoMPASS: RNA Comprehensive Multi-Processor Analysis System for Sequencing

Guorong Xu  
guorong.xu@gmail.com

Follow this and additional works at: <https://scholarworks.uno.edu/td>

---

### Recommended Citation

Xu, Guorong, "RNA CoMPASS: RNA Comprehensive Multi-Processor Analysis System for Sequencing" (2012). *University of New Orleans Theses and Dissertations*. 1531.  
<https://scholarworks.uno.edu/td/1531>

This Dissertation is protected by copyright and/or related rights. It has been brought to you by ScholarWorks@UNO with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Dissertation has been accepted for inclusion in University of New Orleans Theses and Dissertations by an authorized administrator of ScholarWorks@UNO. For more information, please contact [scholarworks@uno.edu](mailto:scholarworks@uno.edu).

RNA CoMPASS: RNA Comprehensive Multi-Processor  
Analysis System for Sequencing

A Dissertation

Submitted to the Graduate Faculty of the  
University of New Orleans  
in partial fulfillment of the  
requirements for the degree of

Doctor of Philosophy  
in  
Engineering and Applied Sciences

by

Guorong Xu

August, 2012

© Copyright 2012, Guorong Xu

## Acknowledgments

First and foremost, I want to thank my advisor Dr. Christopher Taylor. His endless support and wisdom helped me finish this dissertation. His enthusiasm for Bioinformatics was contagious—and I definitely caught it. His depth of knowledge and very precise academic guidance brought me to develop a web-based GUI distributed computational pipeline, which provides all-in-one functionality including human transcriptome quantification, other typical endogenous RNA-Sequencing analysis and additionally the investigation of exogenous sequences.

I would like to thank Dr. Erik Flemington. The major experiment data and biology knowledge were provided by his lab. I really appreciate his consistent support.

I express my deep gratitude to all the professors in my dissertation committee for their precious suggestions about my Ph.D. research and valuable comments on my dissertation.

I wish to thank our bioinformatics group members: Mohamad Qayoom and Joseph Coco and Carl Baribault and Dr. Zhu Dongxiao's lab. Each individual provided insights that guided and challenged my thinking, substantially improved the dissertation.

I am grateful to Dr. Mahdi Abdelguerfi who cared much about my research and my family during the period of my Ph.D. study.

Lastly, I would like to thank my family members, especially my wife Yan Gao, for supporting and encouraging me to pursue this degree. Without my wife's encouragement, I would not have finished this work.

# Table of Contents

<b>List of Figures.....</b>	<b>vii</b>
<b>List of Tables .....</b>	<b>x</b>
<b>Abstract.....</b>	<b>xi</b>
<b>Chapter 1 Background and Introduction.....</b>	<b>1</b>
1.1 Microarray technology .....	1
1.2 Next-generation sequencing technology .....	2
1.3 RNA-seq technology .....	3
1.4 Sequence alignment.....	4
1.5 Junction mapping .....	5
1.6 Sequence searching .....	6
1.7 Taxonomical analysis .....	7
1.8 <i>De novo</i> assembly.....	7
1.9 Introduction to transcriptome .....	8
1.10 Analysis of exogenous sequence.....	9
1.11 Analysis of endogenous sequence.....	10
1.12 Motivation .....	12
1.13 Overview .....	12
<b>Chapter 2 Architecture of RNA CoMPASS .....</b>	<b>15</b>
2.1 Introduction .....	15
2.2 Java Parallel Processing Framework (JPPF) .....	16
2.3 Grid system managed by Portable Batch System (PBS) submission.....	16
2.4 Layer structure of RNA CoMPASS .....	17
<b>Chapter 3 Methods and Tools.....</b>	<b>20</b>
3.1 Introduction .....	20
3.2 Data flow through RNA CoMPASS .....	20
3.3 Initial Serial Pipeline.....	22
3.4 Methods .....	23
3.4.1 Exons reads alignment using Novoalign.....	23
3.4.2 Junctions reads alignment using TopHat .....	24
3.4.3 Sequence searching using BLAST.....	25
3.4.4 Taxonomical analysis using MEGAN .....	26
3.4.5 <i>De novo</i> assembly using ABySS.....	26

3.4.6	Gene expression calculation .....	27
3.4.7	Transcript expression calculation .....	28
3.4.8	Detection of differentially expressed genes and isoforms .....	32
3.4.9	Generation of reads coverage file for visualization .....	33
3.5	Existing bioinformatics tools used in RNA CoMPASS.....	33
3.5.1	Novoalign.....	34
3.5.2	Bowtie .....	35
3.5.3	TopHat .....	35
3.5.4	SAMMate.....	36
3.5.5	SAMtools .....	36
3.5.6	BLAST.....	36
3.5.7	MEGAN.....	37
3.5.8	ABYSS.....	37
<b>Chapter 4 Key Features.....</b>		<b>39</b>
4.1	Introduction .....	39
4.2	Investigation of exogenous sequences of non-host origin .....	40
4.2.1	Feature: Alignment of short RNA sequences against human, virus and bacterial genomes.....	40
4.2.2	Feature: Searching unmapped sequences against human RNA and NT databases .....	42
4.2.3	Feature: Visualization of taxonomic distribution of reads using MEGAN .....	43
4.2.4	Feature: Assembling pools of exogenous reads into longer transcripts with ABYSS .....	45
4.3	Performs extensive endogenous analysis for the host organism .....	46
4.3.1	Feature: calculation of genomic feature abundance scores at gene level ...	46
4.3.2	Feature: calculation of genomic feature abundance scores at isoform level .....	49
4.3.3	Feature: generation of signal map for peak detection.....	50
4.3.4	Feature: generation of wiggle files for visualization .....	50
4.3.5	Feature: generation of alignment report.....	52
<b>Chapter 5 Performance Results.....</b>		<b>53</b>
5.1	The performance comparison in analyzing human organism dataset .....	53
5.2	The performance comparison in analyzing non-human organism dataset.....	58
5.3	The performance comparison between time cost and speedup on local cluster.	61
5.4	The performance comparison between time cost and speedup on grid system .	63
5.5	Microarray platform versus next generation sequence platform.....	64
5.6	Expression analysis .....	67
5.7	3' UTR reporter analysis .....	68
5.8	Splicing evidence in Mutu I and Akata.....	69
5.9	Performance comparison of SAMMate, TopHat and Novoalign.....	72

5.10	Genome-wide change-point analysis to identify potential miRNA targets.....	74
5.11	iQuant algorithm to quantify transcriptomes at isoform-level .....	76
<b>Chapter 6</b>	<b>Conclusion .....</b>	<b>78</b>
6.1	Conclusion.....	78
<b>Appendix A</b>	<b>.....</b>	<b>81</b>
	Tables .....	81
	Glossary.....	84
<b>Appendix B</b>	<b>.....</b>	<b>97</b>
	Important codes .....	97
<b>References</b>	<b>.....</b>	<b>103</b>
<b>Vita</b>	<b>.....</b>	<b>116</b>

## List of Figures

Figure 1.1 Combination of exon reads with junction reads to accurately calculate gene expression RPKM scores (a) A unique challenge for researchers working with RNA-seq data. The junction reads (red) fail to map back to the reference genome because exons are separated by introns. (b) A demonstration of the ideas of combing exon reads (black) and junction reads (red) to calculate gene expression RPKM scores [Xu et al., 2011].	5
Figure 1.2 Endogenous sequence analyses.	11
Figure 2.1 Overview of RNA CoMPASS structure.	15
Figure 2.2 Overview of RNA CoMPASS deployed on a local cluster with JPPF structure.	16
Figure 2.3 Overview of RNA CoMPASS deployed on a grid system managed by PBS submission.	17
Figure 2.4 The layer structure of RNA CoMPASS: the Presentation Layer consists of View module and Controller module, the Business layer consists of Model module. The Third Party Layer consists of existing bioinformatics tools and framework used in RNA CoMPASS.	19
Figure 3.1 Data flow through RNA CoMPASS.	21
Figure 3.2 The structure of Initial serial pipeline. Novoalign and BLAST are bottleneck in this version.	22
Figure 3.3 Investigation of exogenous sequences. On average 87.5% of our reads from RNA-seq experiments in human are identified as mapped reads. The remaining 12.5% of reads that are unmapped could potentially indicate bacterial or viral sequences from exogenous sources.	24
Figure 3.4 Overview of RNA CoMPASS workflow. After deduplication, raw sequence data in FASTQ format is aligned against the reference genome using Novoalign and TopHat. Reads which are mapped are used for endogenous analysis by SAMMate, and unmapped reads are used for exogenous analysis by searching against an optional RNA database and the NCBI NT database by BLAST. Taxonomical analysis is performed on the BLAST results using MEGAN and reads from a given taxon of interest can be extracted for assembly into longer transcripts by assembly tools.	27
Figure 3.5 Overview of coverage file in wiggle format.	33
Figure 4.1 Key features of RNA CoMPASS: A schematic diagram of the two key features of RNA CoMPASS. (1) Discovery and visualization of exogenous sequences of non-host origin. (2) Performs extensive endogenous RNA-Seq analysis for the host organism.	40
Figure 4.2 A screen shot of RNA CoMPASS. In the pre-built indexes selection panel, users can select multiple pre-built index files into system. RNA CoMPASS align the sequence data against the selected reference genomes with Novoalign.	41
Figure 4.3 A screen shot of RNA CoMPASS. In BLAST Parameters panel, users can input different e-value for each step. The feature of using Human RNA database is optional.	42
Figure 4.4 A NCBI tree output by MEGAN. Each node in this tree is labeled by a taxon and the size of a given node represents the number of reads assigned to that taxon.	44



Figure 4.5 The pie chart is another representation of the MEGAN output. The pie chart represents the portion of reads that was assigned to each category.....	45
Figure 4.6 RNA CoMPASS uses ABySS to assemble reads that was assigned to each category into longer transcripts.....	46
Figure 4.7 A unique challenge for researchers working with RNA-seq data. The junction reads (red) fail to map back to the reference genome because exons are separated by introns. ....	48
Figure 4.8 Combination of exon reads with junction reads to accurately calculate gene expression RPKM scores. A demonstration of the ideas of combing exon reads (black) and junction reads (red) to calculate gene expression RPKM scores.....	48
Figure 4.9 Visualization of gene structure variation. Gene CXorf39 was called by the Change Point Analysis as a potential miRNA-155 target due to it's abrupt read dropout on the 3'-UTR end. ....	51
Figure 4.10 Visualization of gene structure variation. Gene LBA1 was called by the Differential Expression Analysis as a potential miRNA-155 target due to the overall read coverage decrease in codon region. ....	52
Figure 4.11 The overview of alignment report. ....	52
Figure 5.1 The box plot of quality scores across all bases for the sample file SRR032238. The horizontal axis corresponds to the base position of sequence. The vertical axis corresponds to the quality score.....	54
Figure 5.2 The overview of GC distribution over all sequences for the sample file SRR032238. The curve marked by red color is GC count per read and the curve marked by blue color is the theoretical distribution. The horizontal axis corresponds to mean GC content (%). The vertical axis corresponds to the number of GC count. ....	55
Figure 5.3 The box plot of quality scores across all bases for the sample file SRR032246. The horizontal axis corresponds to the base position of sequence. The vertical axis corresponds to the quality score.....	56
Figure 5.4 The overview of GC distribution over all sequences for the sample file SRR032246. The curve marked by red color is GC count per read and the curve marked by blue color is the theoretical distribution. The horizontal axis corresponds to mean GC content (%). The vertical axis corresponds to the number of GC count. ....	57
Figure 5.5 Performance and speed up for sample SRR032238 running on a local cluster with 3 node machines. The horizontal axis corresponds to the modules used in RNA CoMPASS. The left vertical axis shows the run time of each module and total time spent processing the sample. The right vertical axis shows the corresponding speedup of the parallelized version for each module.....	58
Figure 5.6 The box plot of quality scores across all bases for the sample file SRR006514. The horizontal axis corresponds to the base position of sequence. The vertical axis corresponds to the quality score.....	59
Figure 5.7 The overview of GC distribution over all sequences for the sample file SRR006514. The curve marked by red color is GC count per read and the curve marked by blue color is the theoretical distribution. The horizontal axis corresponds to mean GC content (%). The vertical axis corresponds to the number of GC count. ....	60

Figure 5.8 Performance and speed up for sample SRR006514 running on a grid system with 24 cores. The horizontal axis corresponds to the modules used in RNA CoMPASS. The left vertical axis shows the run time of each module and total time spent processing the sample. The right vertical axis shows the corresponding speedup of the parallelized version for each module.....	61
Figure 5.9 Performance of a single machine versus a local cluster with three machines for sample SRR032238. The horizontal axis shows the modules used in RNA CoMPASS. The vertical axis shows the speedup of each module for this sample...	62
Figure 5.10 Performance of a single machine versus a local cluster with three machines for sample SRR032246. The horizontal axis shows the modules used in RNA CoMPASS. The vertical axis shows the speedup of each module for this sample...	63
Figure 5.11 Performance of a single machine versus a grid system with 6 and 24 cores allocated for sample SRR006514. The horizontal axis shows the modules used in RNA CoMPASS. The vertical shows the speedup of each module for this sample.	64
Figure 5.12 Cross-platform comparison of targetome prediction using bitmap. Downregulated genes were identified at a false discovery rate (FDR) = 0 for NGS and each microarray platform. Each gene was determined to be significantly down-regulated (at FDR =0) or not in each of the four platforms; down-regulated genes were assigned to one of the 24 = 16 possible clusters, represented by color/white patterns and corresponding to 16 rows in the bitmap. Numbers at the top refer to the total number of down-regulated genes for the indicated platform (summation of the number of genes represented by all colored patterns in column). Numbers to the right refer to the number of genes common to platforms with colored patterns in each respective row.....	66
Figure 5.13 The total number of genes and the number of genes containing any MIR155 seed type that are expressed above and below the indicated RPKM cutoffs in control Mutu I cells were counted and graphed. ....	68
Figure 5.14 Comparison between 3' UTR analysis and RNA-seq analysis. Distribution of 3' UTR suppression by MIR155 in reporter assays.....	69
Figure 5.15 Visualization of junction evidence for EBNA1 (A), BZLF1 (B), and BLLF1/BLLF2 (C). Coverage file in wiggle format was generated by SAMMate which has been integrated into RNA CoMPASS. ....	71
Figure 5.16 Pie chart of percentages of gene fold changes calculated by each tool that is closest to the 3'-UTR experimental results. SAMMate is superior to the other competing tools. ....	73
Figure 5.17 Comparison of Differential Expression Analysis (DEA) and Change Point Analysis (CPA) in Prediction of miRNA-155 Targets. ....	76
Figure 5.18 Simulation study to evaluate the performance of iQuant algorithm implemented in RNA CoMPASS using FluxSimuloator. We plot predicted isoform abundance scores against true abundance scores.....	77

## List of Tables

Table 1.1 BLAST results in hit table format.....	6
Table 5.1. Overview of samples SRR032238 and SRR032246. For each sample, total number of reads contained in the file and the number of unique mapped reads aligned by Novoalign is shown.....	53
Table 5.2 The sample SRR032238 has been split into 3 pieces. For each piece, we list the total number of reads contained and the number of unique mapped reads aligned by Novoalign.....	63
Table 5.3 The sample SRR032246 has been split into 3 pieces. For each piece, we list the total number of reads contained and the number of unique mapped reads aligned by Novoalign.....	63
Table 5.4 The sample SRR006514 has been split into 3 pieces. For each piece, we list the total number of reads contained and the number of unique mapped reads aligned by Novoalign.....	64

## **Abstract**

The main theme of this dissertation is to develop a distributed computational pipeline for processing next-generation RNA sequencing (RNA-seq) data. RNA-seq experiments generate hundreds of millions of short reads for each DNA/RNA sample. There are many existing bioinformatics tools developed for the analysis and visualization of this data, but very large studies present computational and organizational challenges that are difficult to overcome manually. We designed a comprehensive pipeline for the analysis of RNA sequencing which leverages many existing tools and parallel computing technology to facilitate the analysis of extremely large studies. RNA CoMPASS provides a web-based graphical user interface and distributed computational pipeline including endogenous transcriptome quantification and additionally the investigation of exogenous sequences.

## **Keywords**

Parallel Computing

GUI User Interface

Transcriptomic

Next-Generation Sequencing

RNA-seq Pipeline

Exogenous Sequences

# **Chapter 1 Background and Introduction**

## **1.1 Microarray technology**

Microarray technology is widely applied in the research of molecular biology by using a multiplex lab-on-a-chip technology. It typically uses either a one-color or a two-color design to measure mRNA abundance [Patterson et al., 2006]. As the name implies, one-color design refers to that one sample is used and two-color design refers to that two independent samples are used [Shalon et al., 1996]. A small solid glass slide or silicon thin-film cell attaches a large amount of different nucleic acid probes to hybridize a cDNA or cRNA sample (called target) under high-stringency conditions. The relative abundance of nucleic acid sequences in the target can be usually determined by detection and quantification of the probe-target hybridization [Scholin et al., 1997]. Microarray technology widely involves gene discovery, disease diagnosis, drug discovery, toxicological research and so on [Liu 2007]. The platform of microarray typically includes Affymetrix, GeneChip, Illumina and BeadArra. Microarray provides analogue measures of sequence abundance by measuring the fluorescence intensity of arrayed probe sequences because of the intrinsic design. The technology has several limitations, for example, limited to known genome and transcriptomes, limited dynamic range and sensitivity [Russo et al., 2003]. Besides, when researchers design microarray experiments they rely on the gene annotations which may be incorrect or outdated. Therefore, microarray technology has a limited ability to detect alternatively spliced transcripts [Roy et al., 2011]. Microarray was once as the experiment of choice for transcriptome analysis. Although the use of microarray technology remains active in a number of research areas, the promising Next Generation Sequencing (NGS) technology is rapidly becoming an instrumental assay for transcriptomics research.

## 1.2 Next-generation sequencing technology

Next-generation sequencing (NGS) technology or High-throughput sequencing (HTS) technology is rapidly becoming transformative in many areas of biology because NGS has significantly improved throughput and dramatically reduced the cost compared with microarray technology [Corrinne et al., 2012]. NGS technology generates hundreds of millions of short fragments from a library of nucleotide sequences in a single experiment with fewer biases. Currently, NGS platforms support a range of genetic analyses, including whole genome resequencing, gene expression analysis and small ribonucleic acid (RNA) analysis [Liu et al., 2011]. For example, using the Illumina (<http://www.illumina.com/>) Genome Analyzer platform, recent applications include sequencing mammalian transcriptomes [Mortazavi et al., 2008], ABI Solid Sequencing to profile stem cell transcriptomes [Cloonan et al., 2008] or Life Science's 454 Sequencing to discover SNPs in maize [Barbazuk et al., 2007]. Even though technical differences or applications exist in each platform, the information gathered from each platform shares similar principle. Compared with microarray technology, NGS experiments also provide much higher resolution measurements of expression at comparable costs [Marioni et al., 2008]. High-throughput sequencing technologies have overcome many limitations of microarray technology [Russo et al., 2003]. For example, not limited to known genomes/transcriptomes, potential for surveying entire genomes/transcriptomes, including novel un-annotated regions, providing dynamic range and allowing detection of rare sequences, high sensitivity and low background noise, particularly potential for determining gene structure and so on.

The main theme of this dissertation is to develop computational software tools for RNA-sequencing technology, which utilizes high-throughput sequencing technologies to sequence cDNA in order to get information about a sample's RNA content. The following sections will

introduce a series of relevant methods and analysis of whole-transcriptome sequencing data (RNA-seq).

### 1.3 RNA-seq technology

RNA-seq, also called "Whole Transcriptome Shotgun Sequencing" [Ryan et al., 2008] ("WTSS") and dubbed "a revolutionary tool for transcriptomics" [Wang et al., 2009], quickly becomes an instrumental assay for transcriptomics research. For many years, the standard method for determining the sequence of transcribed genes has been to capture and sequence messenger RNA using expressed sequence tags (ESTs) [Adams et al., 1993] or full-length complementary DNA (cDNA) sequences using conventional Sanger sequencing technology. RNA-seq, as an emerging new high-throughput technology, generates far more data per experiment than the conventional EST sequencing, and it generates data that can be used as a direct measure of the level of gene expression [Trapnell et al., 2009]. RNA-seq uses next-generation sequencing (NGS) technologies that can sample the mRNA with fewer biases and low background noise. We take the example of Illumina platform, the initial step is known as building a DNA library and then genomic DNA is randomly fragmented and ligated adapters to both ends of the fragments. By attaching DNA fragments to surface of the flow cell channel in the second step, the third step is bridge amplification in order to amplify each fragment into a cluster by adding enzyme and unlabeled nucleotides. The purpose of amplification is to make the strength of signal stronger so that it is easy to detect the signal from background noise. In the following steps, it is to scan each base of the clusters from each chemical circle to generate the sequences using laser. By using this way, RNA-seq can quickly generate hundreds of millions of short reads in a single run. Therefore, RNA-seq has been rapidly revolutionized the field of

transcriptomics allowing researchers to characterize gene expression within an organism under a variety of conditions [Costa et al., 2010].

## 1.4 Sequence alignment

The first step of any NGS platform consists of sequence alignment and assembly [Valerio et al., 2010, Metzker 2010]. In an RNA-seq experiment, the computing power to track all the possible alignments is nontrivial when aligning hundreds of millions of short reads to a reference genome [Paşaniuc et al., 2010; Manske et al., 2009; Kircher et al., 2011]. In a typical RNA-seq experiment, hundreds of millions of short reads are generated from a library of nucleotide sequences. We need to map these short reads of mRNA to identify regions of similarity on the reference genome. Due to the length of short read, aligning a huge volume of short reads to a long reference genome poses a great challenge to analysis of RNA-seq data. There are several tools MAQ [Li H et al., 2008], SOAP [Li R et al., 2008], RMAP [Smith et al., 2008], Bowtie [Langmead et al., 2009] and Novoalign (<http://www.novocraft.com>) available for aligning genomic reads to a reference genome. The Needleman–Wunsch algorithm [Needleman and Wunsch 1970] and the Smith-Waterman algorithm [Smith and Waterman 1981] are mainly used to perform sequence alignment. The Needleman–Wunsch algorithm performs a global alignment on two sequences [Durbin et al., 1998]. It is an example of dynamic programming which was the first application of dynamic programming to biological sequence comparison and it is suitable when the two sequences are of similar length with a significant degree of similarity throughout. The Smith-Waterman algorithm performs a local alignment on two sequences. It is also an example of dynamic programming and used for dissimilar sequences that are suspected to contain regions of similarity or similar sequence motifs within their larger sequence context [Durbin et al., 1998].



## 1.5 Junction mapping

Since short reads are generated from mRNA, which consists exclusively of exons with all introns removed. Aligning reads originating from exon-exon junctions to reference genome is also a hard nut to crack for researchers. Although most of the short reads can be mapped on exon regions, there are still a few of short reads originating from exon-exon junctions still cannot be aligned against to reference genome [Chepelev et al., 2009]. Thus, working with the short reads originating from exon-exon junctions in cDNA (around 10%) is a unique challenge for researchers. However, millions of unmapped short reads originating from exon-exon junctions, denoted as Initially Unmapped Reads (IUM's), need to be accounted for when measuring gene expression. To address the IUM problem, ERANGE [Mortazavi et al., 2008], Tophat [Trapnell et al., 2009] and rSeq [Jiang et al., 2008] are among the recently developed approaches to map IUM's originating from exon-exon junctions back to individual genes. ERANGE uses a union of known and novel junctions while Tophat *de novo* assembles IUM's using a module in Maq [Li H et al., 2008].

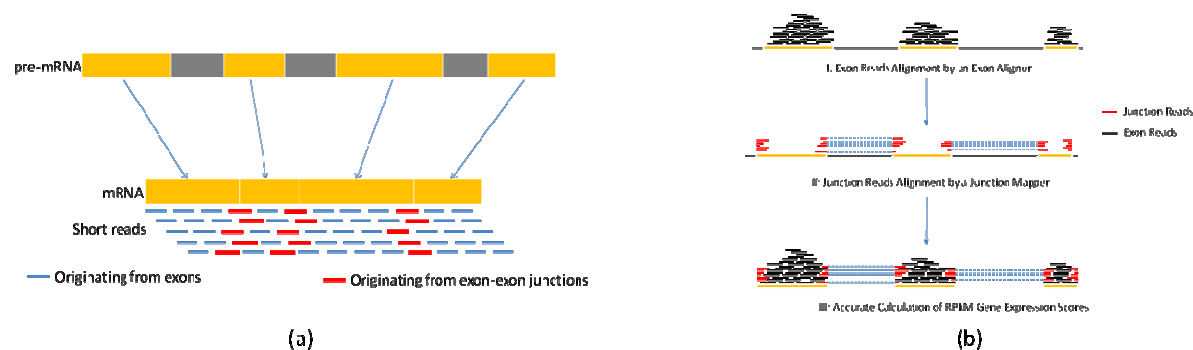


Figure 1.1 Combination of exon reads with junction reads to accurately calculate gene expression RPKM scores (a) A unique challenge for researchers working with RNA-seq data. The junction reads (red) fail to map back to the reference genome because exons are separated by introns. (b) A demonstration of the ideas of combining exon reads (black) and junction reads (red) to calculate gene expression RPKM scores [Xu et al., 2011].

## 1.6 Sequence searching

BLASTN, a nucleotide alignment tool, is a very important component of the BLAST+ suite. It can search any properly formatted database and there are several regularly updated versions of common databases available online. It is compatible with multiple platforms such as MacOS, Linux, and Windows systems. The time cost and memory requirements of BLAST are greatly impacted by both the number of sequences being searched against the database and the size of the database being searched. BLASTN is used to infer sequence function, taxonomy, and phylogeny. Output in several formats is provided with varying degrees of information, for example: pairwise (for human readability), BLASTTAB (for ease of parsing by scripts), BLASTXML (for universal ability to be parsed) and Hit table format. The following table is an example of hit table format.

Query id	Subject id	identity	alignment length	mismatches	gap openings	q.start	q.end	s.start	s.end	evalue	bit
HWI-EAS185:2:1:123:702#count=1#0/1	gi 346421552 gb JN204881.1	100.00	40	0	0	1	40	186	147	2e-11	75.0 1084337
HWI-EAS185:2:1:123:702#count=1#0/1	gi 342675415 gb HQ683722.1	100.00	40	0	0	1	40	2090	2051	2e-11	75.0 1048777
HWI-EAS185:2:1:123:702#count=1#0/1	gi 342675411 gb HQ683721.1	100.00	40	0	0	1	40	7487	7448	2e-11	75.0 1048776
HWI-EAS185:2:1:123:702#count=1#0/1	gi 341873824 gb JN195815.1	100.00	40	0	0	1	40	3001	2962	2e-11	75.0 1070442
HWI-EAS185:2:1:123:702#count=1#0/1	gi 316980675 dbj AB570081.1	100.00	40	0	0	1	40	9042	9003	2e-11	75.0 860080
HWI-EAS185:2:1:123:702#count=1#0/1	gi 316980672 dbj AB570080.1	100.00	40	0	0	1	40	3780	3741	2e-11	75.0 860079

Table 1.1 BLAST results in hit table format

Query id: Name of the sequence which was used for the search

Subject id: The name of the sequence found in the BLAST search

Identity: The number of identical residues in the query and hit sequence

Score: The bit score of the local alignment generated through the BLAST search

Hit start: The start position in the hit sequence

Hit end: The end position in the hit sequence

Query start: The start position in the query sequence

Query end: The end position in the query sequence

E-value: Measure of quality of match

## 1.7 Taxonomical analysis

Taxonomical analysis is performed to categorize potential exogenous sequences results generated by BLAST [Altschul et al., 1997]. The BLAST results in hit table format is imported into MEGAN [Huson et al., 2007] which is a tool developed for metagenomic analysis. MEGAN [Huson et al., 2007] automatically calculates a taxonomic classification of the reads or a functional classification using either the SEED or KEGG classification, or both [Huson et al., 2007]. The SEED classification outputs a tree and each node of the NCBI taxonomical tree is labeled with a taxon. The size of a given node represents the number of short reads assigned to that taxon. The results can be interactively viewed and inspected. For example, the researcher can export all reads that were assigned to a specific taxon for assembling these reads into longer transcripts using ABySS [Birol et al., 2009]. This provides the researcher with an overview of reads found in their data of possible exogenous origin. The NCBI classification tree would be helpful for the researcher to understand the prior biological knowledge of the experiment at hand, or hypotheses that the researcher wants to test given the taxonomic classification displayed by MEGAN. Additionally, one can select a set of taxa and then use MEGAN to generate different types of charts for them.

## 1.8 *De novo* assembly

Each pool of reads exported from a given tax can be subsequently assembled into longer transcripts using some *de novo* assembly tools. For example, ABySS [Birol et al., 2009] is a *de novo* parallel sequence assembler. In the final phase of RNA CoMPASS, the researcher can extract reads from a taxon of interest to assemble them into longer transcripts [Birol et al., 2009], using a *de novo* parallel sequence assembler. This process provides the researcher with a broader

view of the particular transcripts that were found within a given taxon. This process can be repeated for each taxon of interest and the researcher can search the longer assembled transcripts against the databases again to get more precise hits. This process is the only step of RNA CoMPASS that is not automated and requires researcher intervention. The researcher can export all extracted reads from MEGAN [Huson et al., 2007] in a file and then upload this file to the pipeline to perform the *de novo* assembly. The researcher can also assemble the reads exported from “not assigned” or “no hits” category to perform *de novo* assemble in case they were portions of longer transcripts that simply were not found by MEGAN [Huson et al., 2007].

## 1.9 Introduction to transcriptome

The term of transcriptome in genetics is defined as the complete set of all RNA molecules, including mRNA, rRNA, tRNA, and other non-coding RNA produced in one or a population of cells, or it can be referred to as the total of transcripts (or called isoform) or the specific subset of transcripts in a living cell [Pacheco et al., 2006]. Unlike the genome that nearly does not change in a living cell except for mutation cases, transcriptome is highly diverse, dynamic, complex and overlapping [Li et al., 2010]. The transcriptome dynamically varies under different external environmental conditions at a particular time, such as specialized tissues or cell lines. Most of the transcripts are processed by splicing to remove introns and generate a mature transcript or messenger RNA (mRNA) that only contains exons. Importantly, the range of transcriptome is enhanced by alternative splicing. Alternative splicing is a fundamental molecular process of multiple transcripts from a single gene due to variations in the splicing reaction of pre-mRNA [Garcia-Blanco et al., 2004]. An exon can be either included or excluded from the mature transcripts. Thus, different splicing variants are generated from the same gene.

Global transcriptome analysis is becoming important in understanding how altered expression of genetic variants contributes to complex diseases such as cancer, diabetes, and other genetic disease [Olden et al., 2011]. Analysis of the transcriptomes of human is used to understand the molecular mechanisms and biological signaling pathways controlling early embryonic development.

## 1.10 Analysis of exogenous sequence

Since the human body is a persistent host to a spectrum of not only bacterial organisms but also viruses, any biological contamination of foreign organisms occupying the host will be referred to as exogenous agents [Sekirov et al., 2010]. Harbor exogenous agents such as the human tumor viruses and bacterial have been found in many commonly used cell lines for biological studies. Therefore, contamination is an important factor that must be considered for any biological experiment [Coco et al., 2011]. However, the conventional approaches often have limitations in globally assessment of the presence of foreign organisms within cell model systems. With high-throughput sequencing technology, some significant evidences have been found in commonly used cell lines such as Epstein-Barr virus (EBV) virus transcription in type I Burkitt's lymphoma cells [Lin and Xu et al., 2010]. In a RNA-seq experiment, the researchers often extract the mapped reads for further analysis, for example transcriptome characterization and quantification [Xu et al., 2011]. And the reads that do not map well to the reference genome are often simply discarded. However, important information could be lost by ignoring these unmapped reads. Some of the reads that do not map to the host genome could be indicative of bacterial or viral sequences in the RNA-Seq experiment [Coco et al., 2011]. Analysis of exogenous sequence will be referred to as exogenous agent coding reads from analysis of the host organism which was introduced during the procedures involved in reading the RNA

sequences from the host organism. BLAST is used to search the possible exogenous sequence reads against a very large and broad database [Benson et al., 2010] in our computational pipeline.

## 1.11 Analysis of endogenous sequence

Quantifying genomic feature abundance in cells via measurement of mRNA levels arouses researchers' interest all the time [Tuller et al., 2007]. For analysis of transcriptome quantification, it includes estimation of gene abundance score at gene level and isoform (transcript) level.

For measuring gene expression, we often have to align short reads to original positions against a reference genome using alignment tools such as Bowtie, Novoalign and other sequence aligner. Then we can count the number of short reads mapped on gene regions based on gene annotation table to estimate the genomic feature abundance. In RNA-seq experiment, for instance, ERANGE reports the number of mapped **R**eads **P**er **K**ilobase of exon per **M**illion mapped reads (RPKM) for each gene, a measure of transcription activity [Trapnell et al., 2009]. For paired-end short reads, we measure the transcript-level relative abundance in **F**ragments **P**er **K**ilobase of exon model per **M**illion mapped fragments (FPKM).

$$\text{RPKM/FPKM} = 10^9 \times \frac{C}{L \times N}$$

$C$  is the total number of mapped short reads or fragments,  $L$  is the length of exons and  $N$  is the total number of short reads in one lane of one experiment. When scaled to range [0, 1000], this value stands for the normalized depth of coverage for each gene. Using this way, we can estimate the abundance of mRNA in the cells. Even though it has been shown that there is no strong correlation between the abundance of mRNA and the related proteins [Greenbaum et al.,

2003], measurement of mRNA levels is still very useful in determining how cells differ between a healthy state and a diseased state and other research problems.

Transcript quantification using RNA-seq plays a critical role in a wide range of transcriptomics research. Since the diverse alternative splicing mechanisms of transcripts, it can pose a challenge for researchers in transcript quantification. Fortunately, there are several existing bioinformatics tools and computational approaches have been developed to use high throughput gene expression profiling data collected RNA-seq experiments such as, Cufflinks [Trapnell et al., 2010], rQuant.web [Bohnert and Räscht 2010], RAEM [Deng et al., 2011] and iQuant [Nguyen et al., 2011]. In our pipeline, we implemented RAEM algorithm [Deng et al., 2011] and iQuant algorithm [Nguyen et al., 2011].

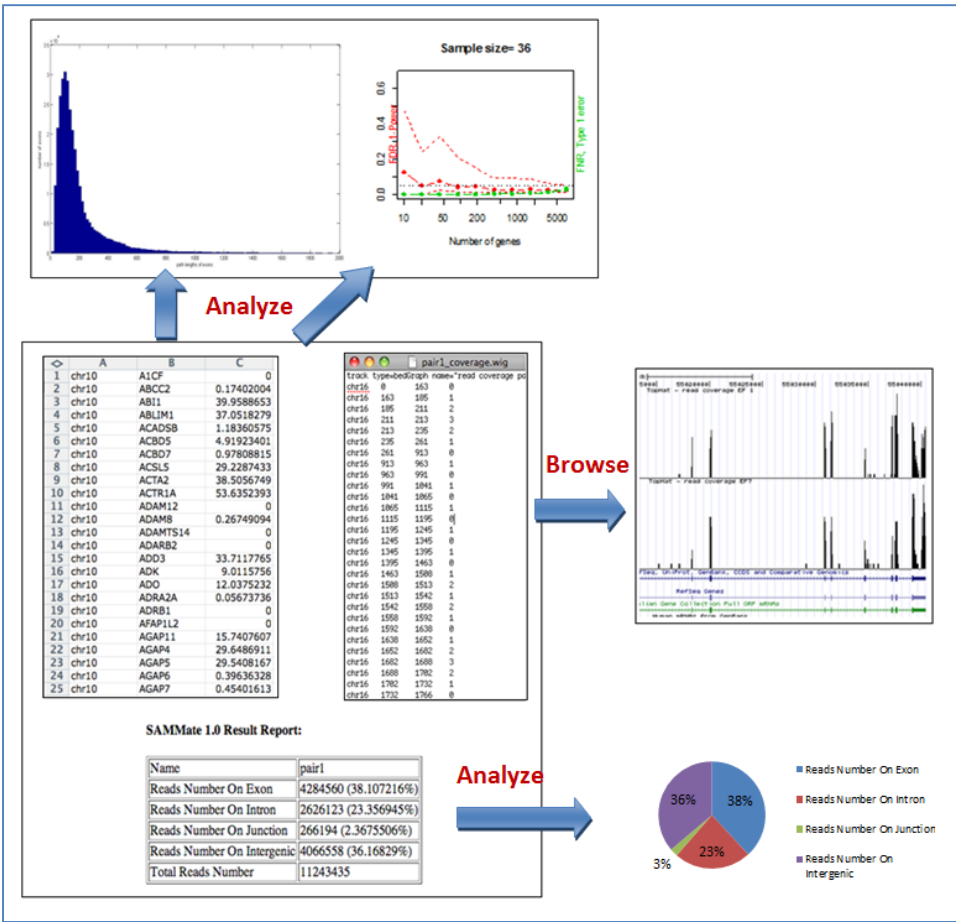


Figure 1.2 Endogenous sequence analyses.

## 1.12 Motivation

Technical limitations of microarray technology constrain its ability to comprehensive human transcriptome quantification and the typical endogenous RNA-Sequencing analysis along with the investigation of exogenous sequences. Fortunately, high-throughput multiplexed next-generation sequencing provides a digital readout of absolute transcript levels and imparts a higher level of accuracy and dynamic range than microarray platforms. High-throughput RNA sequencing has become an instrumental assay for transcriptomics research. There are many existing bioinformatics tools designed for analysis and visualization of this data, but very large studies present computational and organizational challenges that are difficult to overcome manually. A dramatic increase in the size of datasets often exceeds the computing capability of these tools run on a single workstation. We have designed a comprehensive pipeline for analysis of RNA sequencing which leverages many existing tools and parallel computing technology to facilitate the analysis of extremely large studies. RNA CoMPASS provides a web-based graphical user interface and distributed computational pipeline including endogenous transcriptome quantification along with the investigation of exogenous sequences. RNA CoMPASS is deployable on either a local cluster or a grid environment managed by Portable Batch System (PBS) submission.

## 1.13 Overview

This thesis is organized into 6 chapters. In Chapter 1 we introduce background, motivation and overview. We describe the next-generation technology and the related hot biological problems that biologists are interested in, and then we also present our motivation that we design a comprehensive pipeline for endogenous sequence analysis along with the



investigation of exogenous sequences. In Chapter 2 we describe methods and tools. In this chapter, we detailed introduce the methods and tools used in RNA CoMPASS, and we show the limitations of the existing tools and provide our novel solution to solve these challenges. In Chapter 3 we present design and implementation of RNA CoMPASS. In this chapter, we describe two frameworks used in RNA CoMPASS, JPPF framework and PBS framework. With these two frameworks, the pipeline can be deployed on either a local small cluster or a grid system managed by PBS scheduling submission. The system greatly facilitates the analysis of large RNA sequencing studies through automated dataflow management and acceleration of processing via distributed computing over a cluster. Besides, we also adopt the layer structure and Model-View-Controller (MVC) model to implement the pipeline. With this design model, the pipeline shows the advantages of implementation over other tools, for example, extensibility, reusability, scalability and functionality. In Chapter 4 we discuss the key features offered by the pipeline including the analysis of endogenous sequence and the investigation of exogenous sequence, and related useful applications. In Chapter 5, we list the important results of all key features offered by the pipeline and compare the performance of the pipeline with the extra-large dataset. In Chapter 6 we draw a conclusion of the thesis and present the contribution of the pipeline for transcriptomics research.

The thesis is largely based on the following list of relevant publications and software. In the category of discovery and visualization of exogenous sequences of non-host origin, the submitted paper 1) mainly focus on the implementation and contributions of the pipeline RNA CoMPASS which is designed and implemented by me; in the paper 2) I contributed part of results from the tool PARSES, which has been integrated into the new pipeline; in the paper 3) my contribution is that I assist them to generate most of analysis result from the pipeline RNA

CoMPASS. In the category of the performing of extensive endogenous RNA-Seq analysis for the host organism, I performed the sequence alignment, ranked the list of gene abundance and provided the important results in the paper 4). In the paper 5), the results are generated from the software SAMMate. The paper 6) is mainly focus on the implementation and contributions of the software SAMMate which is designed and developed by me.

- Discovery and visualization of exogenous sequences of non-host origin
- 1) **Xu Guorong**, Strong M, Flemington EK and Taylor C: RNA CoMPASS: RNA comprehensive multi-processor analysis system for sequencing. (In submission)
- 2) Lin,Z, Puetter A, Coco J, **Xu Guorong**, Strong M, Wang X, Fewell C, Baddoo M, Taylor C and Flemington EK: Detection of Murine Leukemia Virus in the Epstein-Barr Virus-Positive Human B-Cell Line JY, Using a Computational RNA-Seq-Based Exogenous Agent Detection Pipeline, PARSES. *J. Virology*, 2012; doi: 10.1128/JVI.06717-11
- 3) #Strong M, #**Xu Guorong**, Coco J, Concha M, Baribault C, Baddoo M, Taylor C and Flemington EK: Detection and transcriptome analysis of Epstein Barr virus in RNA-seq data from clinical Gastric Carcinoma samples using a computational pipeline RNACoMPASS (RNA comprehensive multi-processor analysis system for sequencing) (In preparation).
- Performs extensive endogenous RNA-Seq analysis for the host organism
- 4) **Xu Guorong**, Fewell C, Taylor C, Deng N, Hedges D, Wang X, Zhang K, Lacey M, Zhang H, Yin Q, Cameron J, Zhen L, Zhu D and Flemington EK: Transcriptome and targetome analysis in mir155 expressing cells using rna-seq. *RNA* (New York, N.Y.), 2010; 16(8):1610-1622.
- 5) #Lin Z, #**Xu Guorong**, Deng N, Taylor C, Zhu D and Flemington EK: Analysis of EBV transcriptome using RNA-seq. *J. Virology*, 2010; doi:10.1128/JVI.01521-10.
- 6) **Xu Guorong**, Deng N, Zhao, Z, Flemington EK and Zhu D: SAMMate: A GUI tool for processing short read alignment information in SAM/BAM format. *Source Code for Biology and Medicine*, 2011; 6:2
- Software

RNA CoMPASS [available from, <http://rnacompass.sourceforge.net>]

SAMMate [available from, <http://SAMMate.sourceforge.net>]

## Chapter 2 Architecture of RNA CoMPASS

### 2.1 Introduction

RNA CoMPASS uses a Client-Server (C/S) structure to simplify installation and maintenance so that the researcher just needs to open a web browser to access RNA CoMPASS via the internet. The researcher interacts with a Graphical User Interface (GUI) to upload their data to the Tomcat web server portion of RNA CoMPASS and initiate the analysis. The Tomcat web server connects to the computational cluster to perform processing of the analysis and results are stored on the Tomcat web server. The researcher can now download analysis results from the server for visualization and/or further downstream analysis. By using this architecture, RNA CoMPASS provides cross-platform compatibility. The researcher doesn't need to install or upgrade any software; all installation and upgrade procedures are performed on the Tomcat server by the administrator. RNA CoMPASS has a very user-friendly GUI and an embedded FTP server providing secure and fast data transfer.

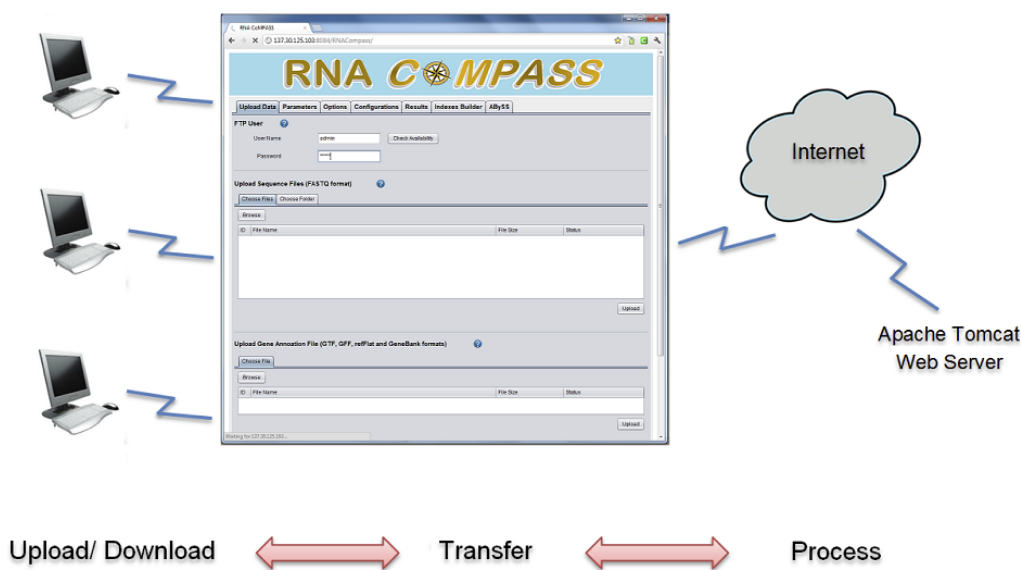


Figure 2.1 Overview of RNA CoMPASS structure.

## 2.2 Java Parallel Processing Framework (JPPF)

Java Parallel Processing Framework (JPPF) is a distributed parallel processing framework based on a Client and Server architecture. A JPPF grid consists of 3 kinds of components that communicate with each other: client submits work to the cluster, and the server receives work from the clients and distributes it to the computational nodes which execute the work in parallel.

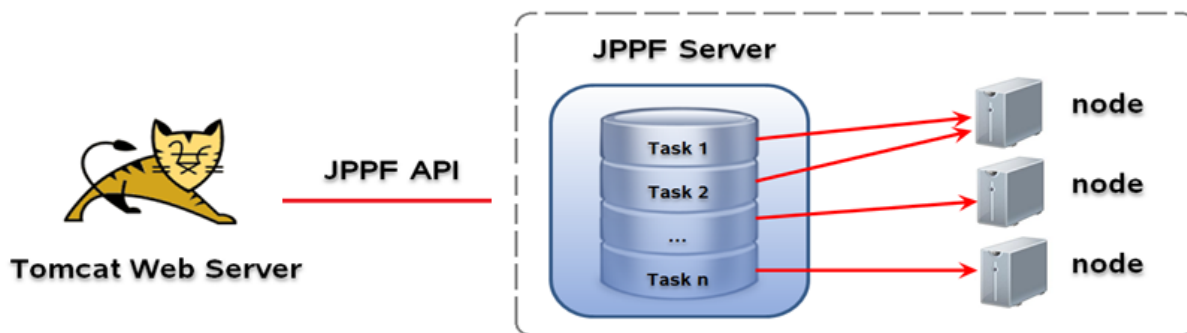


Figure 2.2 Overview of RNA CoMPASS deployed on a local cluster with JPPF structure.

The administrator can assign one machine in a lab as the JPPF server, and assign the others as computational nodes. Setup a local cluster in this fashion is very simple. The administrator downloads one package and unzips it on the JPPF server which RNA CoMPASS is running on, then launches it by typing the command "ant". RNA CoMPASS can automatically detect the JPPF server and send tasks to the JPPF server via the JPPF API. Then the JPPF server assigns tasks to its attached node machines and collects results from each node machine after processing.

## 2.3 Grid system managed by Portable Batch System (PBS) submission

A computer grid environment consists of a set of loosely connected computers that work together so that they can be treated as a single system in many respects. Portable Batch System (PBS) is software that performs job submission and scheduling. This software is in charge of

allocating computational tasks such as batch jobs across the available computing resources of the grid.

RNA CoMPASS is easy to deploy to a grid system managed by PBS submission. The administrator copies the Novoalign and BLAST executable files, related Novoalign index and NT database files, and some shell scripts to the grid data storage center. Then the administrator enables the PBS feature by modifying a property in an RNA CoMPASS system file. A grid system with more than one hundred machines can greatly accelerate the data processing.

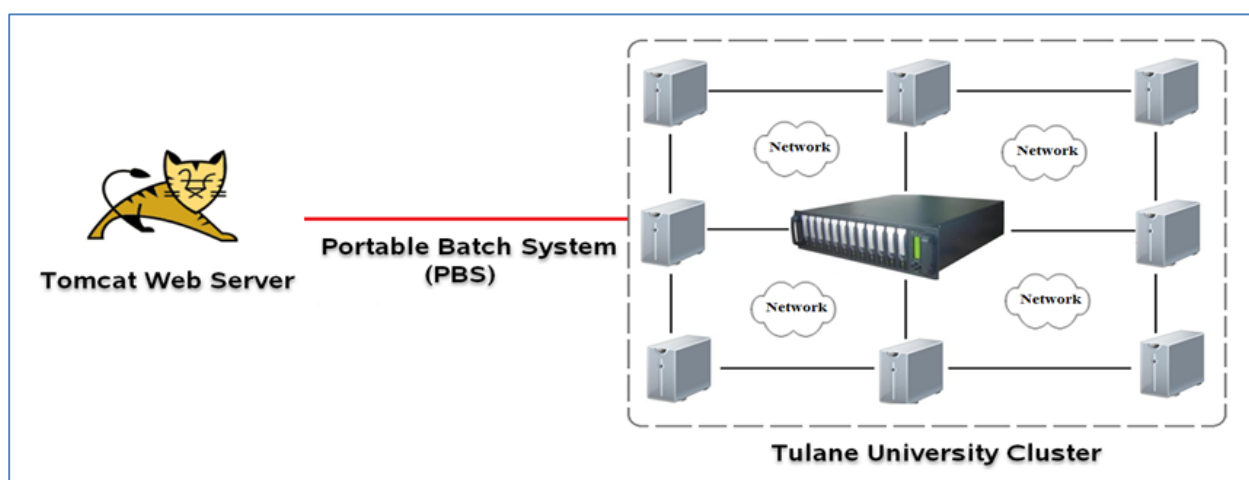


Figure 2.3 Overview of RNA CoMPASS deployed on a grid system managed by PBS submission.

## 2.4 Layer structure of RNA CoMPASS

We now provide a description of the software architecture to enlighten users about key RNA CoMPASS modules and their interconnections. The architecture of RNA CoMPASS follows the standard Model-View-Controller (MVC), a common architectural pattern used in software engineering. The MVC approach decomposes the problem into input data (model), presentation of the data (view), and business logic (controller) (Figure 2.4). The basic software architecture has three components: JSP pages package (Presentation of the data processing module), Calculator package (Business logic processing module) and Action package (Input data

processing module). The advantages of adopting the MVC approach are malleability, modularity, reusability, flexibility and extensibility.

RNA CoMPASS's reusability is very robust as developers can easily reuse existing classes by using the new method to create an instance of a class. RNA CoMPASS is also very extensible. If a user wishes to expand upon a component, the user may simply use the "*extends*" key word to inherit the methods and properties of the desired class. Developers may also conveniently add classes to implement new features. For example, developers may add a new parser to process a gene annotation file in a new format by adding a class to the Alignment package. Furthermore, a developer may add a new tab in the Applet interface of JSP page to support new features. One other important feature of RNA CoMPASS is its configurability. For example, RNA CoMPASS allows users to flexible switch RNA CoMPASS to deploy on a local cluster (JPPF framework) to a grid system (PBS submission) in a configuration file. In summary, the RNA CoMPASS software architecture implements a number of Applied Programming Interfaces (API's) so that other software developers may easily extend and build more utilities.

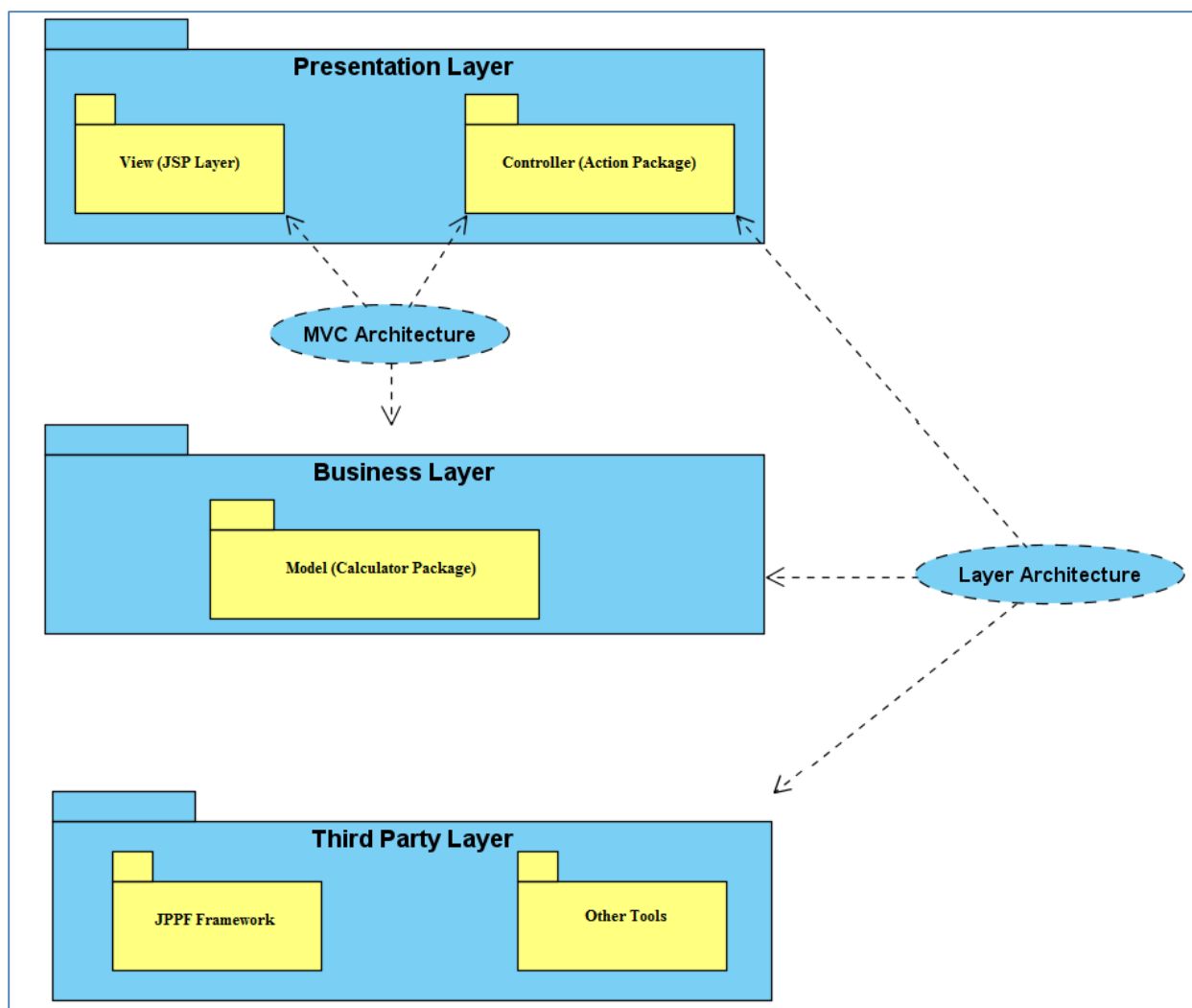


Figure 2.4 The layer structure of RNA CoMPASS: the Presentation Layer consists of View module and Controller module, the Business layer consists of Model module. The Third Party Layer consists of existing bioinformatics tools and framework used in RNA CoMPASS.

## Chapter 3 Methods and Tools

### 3.1 Introduction

High-throughput RNA sequencing has revolutionized the field of transcriptomics allowing researchers to characterize gene expression within an organism under a variety of conditions [Costa et al., 2010]. New sequencing technologies can rapidly and inexpensively generate hundreds of millions of short reads from a single experiment. There are a plethora of quality bioinformatics tools available for mapping these reads to a host genome and performing additional downstream analyses. However, the management of large studies and heavy computational burden can pose a challenge for researchers who may not be familiar with command-line tools or notions of distributed computing. To address this growing problem, we have developed a comprehensive system for RNA-seq analysis (RNA CoMPASS) that is accessed via an easy to use web-based graphical user interface. This system is deployable to a local cluster or a grid environment to address the growing need for computing power brought about by larger studies. RNA CoMPASS leverages some of the most useful open source tools and automates the distribution of the computational burden over the available computing resources. In addition to performing the typical analyses of endogenous sequences, RNA CoMPASS also incorporates a new method to discover exogenous agents [Coco et al., 2011].

### 3.2 Data flow through RNA CoMPASS

The data flow through RNA CoMPASS is shown in the Figure 3.1. At first, the new machine generates hundreds of millions of short reads; then we use Novoalign or TopHat to align short reads against the reference genome. The reads mapped on exon or junction regions are used for transcriptome quantification, for example, to calculate genomic feature abundance at



gene level or isoform level. Then, we extract unmapped reads and de-duplicate the short reads. In the following steps, we use BLAST to search unmapped reads against human RNA database and NT database. This step is extremely computationally intensive. Taxonomical analysis is performed in the next step by importing the BLAST results into MEGAN to categorize exogenous sequences. MEGAN generates a tree and each node of this tree is labeled by a taxon. This provides the researcher with an overview of reads found in their data of possible exogenous origin. In the last step, the researcher can export all reads that were assigned to a specific taxon for further analysis. For instance, we can use ABySS to assemble the reads into longer transcripts of specific species.

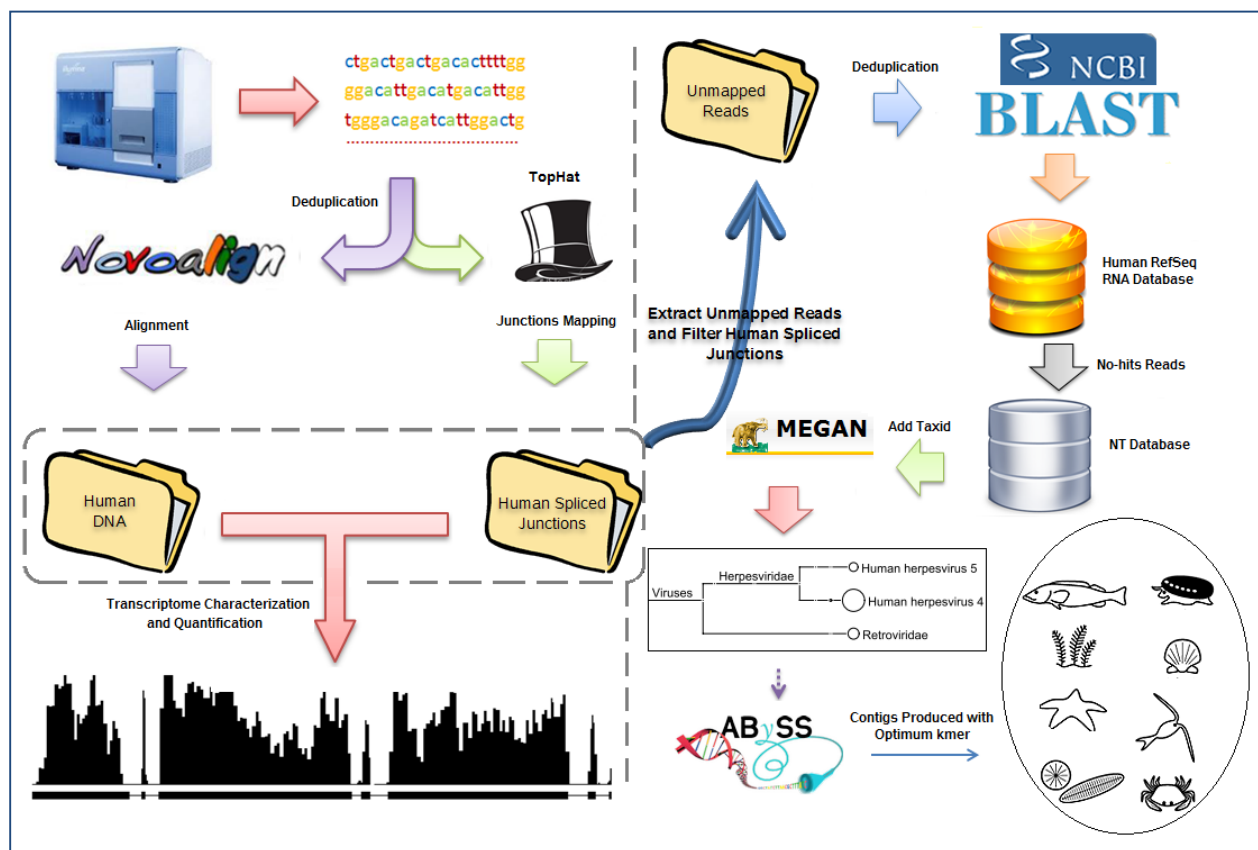


Figure 3.1 Data flow through RNA CoMPASS.

### 3.3 Initial Serial Pipeline

In the initial version, we serially run our pipeline and faced some technical challenges. For example, huge sized data file and many existing bioinformatics tools involved. It is hard for biomedical researchers to deal with many command-line tools. Some of tools need to run on a very powerful workstation with a lot of memory and they are time consuming. Novoalign and BLAST are bottle neck in the whole pipeline. We often spent more than one day to finish one small sample with less than 1 GB file on a very powerful Mac machine with 64G memory and 24 CPU processors. For solving this problem, the solution is the parallel computing.

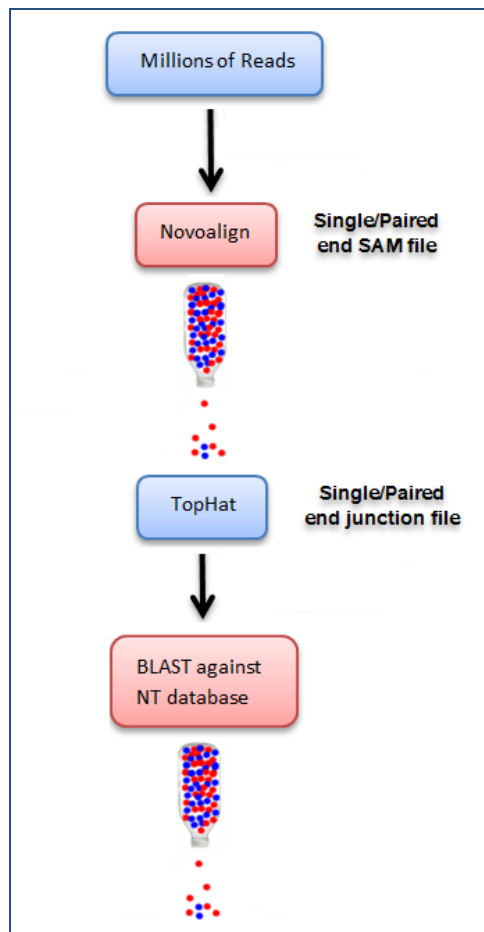


Figure 3.2 The structure of Initial serial pipeline. Novoalign and BLAST are bottleneck in this version.

The key to parallelization is to split the data into many pieces and allocate these small dataset to many machines, then gather and merge the results together and continue the next step. For example, we split the large data file into  $N$  small files and send 1 small file to 1 of  $N$  machines to run Novoalign. It is the same as BLAST. The parallelized pipeline has several advantages over the previous version. Each machine can use memory more efficient and can speed up  $N$  times theoretically. By adding one additional step that BLAST against RNA database, we can improve the accuracy of hits. In particular, the new parallelized version can handle extremely large data file from extensive sequencing projects. We only spend several hours to finish one real sample data with more than 20 million of short reads data file on a local cluster with 3 Mac machines.

## 3.4 Methods

RNA CoMPASS incorporates many existing sequence analysis tools for alignment of short reads, mapping of splice junctions, estimation of transcriptome abundance at both gene and isoform level, searching of sequence databases and assembly of reads into longer transcripts. The system facilitates analysis of large RNA sequencing studies through automated dataflow management, access through a convenient graphical user interface, and acceleration of processing via distributed processing over a cluster.

### 3.4.1 Exons reads alignment using Novoalign

The first phase of RNA CoMPASS is to perform the alignment of millions of short reads against the host genome using a very accurate aligner, Novoalign (<http://www.novocraft.com/>). Novoalign categorizes reads into four classes: uniquely mapped reads, repeat mapped reads, unmapped reads and quality controlled reads. Typical endogenous RNA sequencing analysis concentrates on the uniquely mapped reads and repeat mapped reads, discarding the unmapped

reads. For the investigation of exogenous sequences, we extract unmapped reads from the alignment results, which typically comprise 10% to 15% of total reads in human studies [Lin et al., 2012]. This initial mapping phase is accelerated by deduplicating input reads and distributing across many computing nodes to perform alignment in parallel.

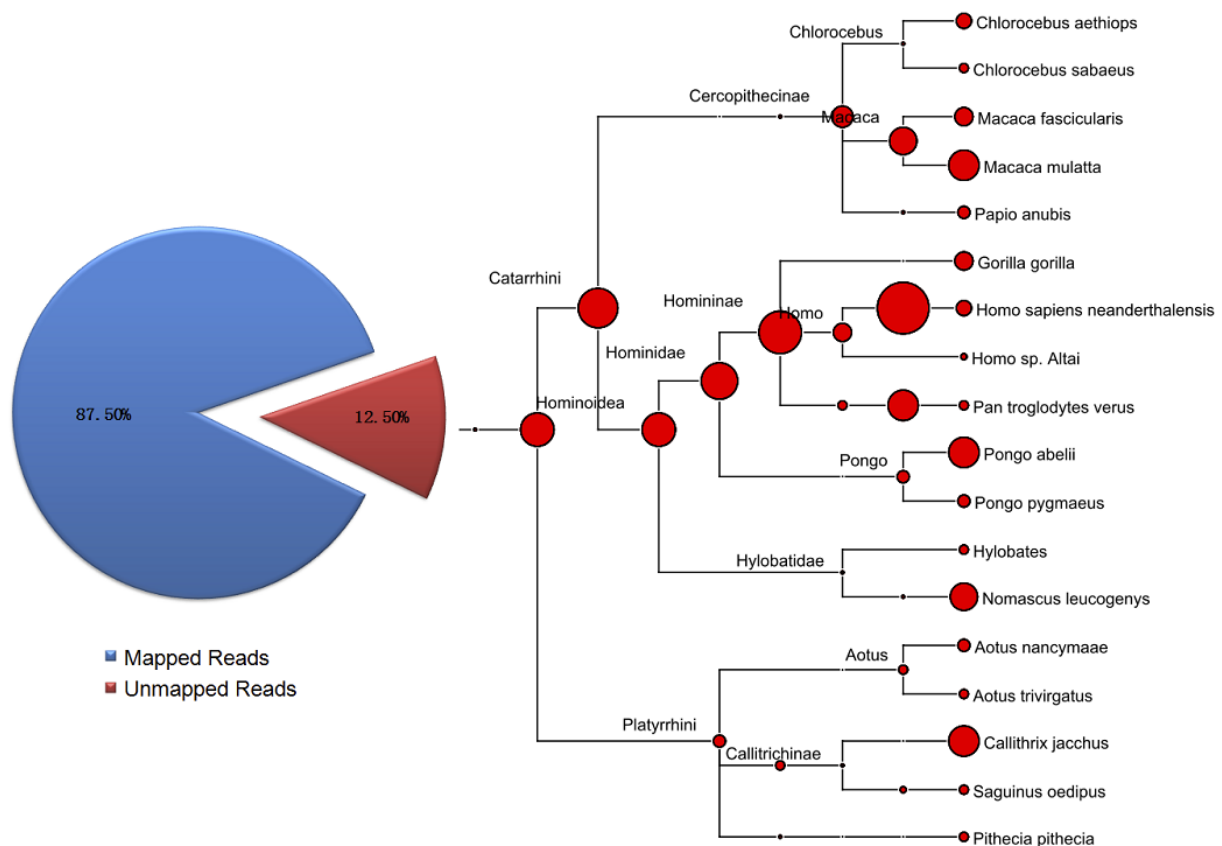


Figure 3.3 Investigation of exogenous sequences. On average 87.5% of our reads from RNA-seq experiments in human are identified as mapped reads. The remaining 12.5% of reads that are unmapped could potentially indicate bacterial or viral sequences from exogenous sources.

### 3.4.2 Junctions reads alignment using TopHat

The second phase utilizes TopHat [Trapnell et al., 2009] to identify splice junction reads that span multiple exons. These splice junction reads are not detected by Novoalign and typically comprise 5% to 6% of the original reads [Xu et al., 2011]. Up to half of the unmapped reads from Novoalign are identified as splice junction reads using TopHat. RNA CoMPASS removes splice junction reads from the unmapped category of Novoalign by using an intermediate file

from TopHat. These reads would not be mapped by Novoalign to the human reference genome because of the presence of introns. Since this phase is independent of the Novoalign mapping, it is started in parallel with the first phase and upon completion of these two mapping tasks, the processing is bifurcated into analysis of endogenous sequences and investigation of exogenous reads. The second phase is optional and it is only designed for reference genomes which have transcripts that are alternatively spliced from individual genes.

### 3.4.3 Sequence searching using BLAST

Exogenous sequence analysis proceeds concurrently with the endogenous analysis. This phase of RNA CoMPASS utilizes BLAST [Altschul et al., 1990] to search unidentified reads from the initial mapping stages against the NCBI NT database for identification. This process is extremely computationally intensive and is distributed across the computing cluster to minimize processing time and memory requirements. The running time and memory requirements of BLAST rely on the number of reads being searched and the size of the NCBI NT database. RNA CoMPASS filters out reads originating from the human genome prior to search against the NCBI NT database to help manage the computational requirements of this intensive BLAST analysis. We have also discovered that many of the reads that were not mapped to the host genome in the first two stages of analysis are, in fact, identified by BLAST as mapping to the host genome since BLAST is a more permissive search. In order to further reduce the computational burden, RNA CoMPASS offers an optional stage prior to the BLAST against NT where the user can BLAST against a host transcript database to further filter these reads from consideration. This stage takes advantage of the lower computational burden of BLASTing reads against a smaller database (the host transcript database) before BLASTing the remaining reads against a larger database (the NT database).

### 3.4.4 Taxonomical analysis using MEGAN

Taxonomical analysis is performed in the next phase by importing the BLAST results into MEGAN [Huson et al., 2007] to categorize exogenous sequences. To allow MEGAN to determine the taxon associated with matches, the NCBI taxon id number is appended to each BLAST hit by looking up the GI accession number of the hit in the GI to TaxID file. MEGAN then determines the taxon associated with matches based on the hit table using a lowest common ancestor algorithm. MEGAN categorizes the exogenous sequences and outputs a NCBI taxonomy tree. Each node of the output tree is labeled by a taxon and the size of a given node represents the number of reads assigned to that taxon. This provides the researcher with an overview of reads found in their data of possible exogenous origin. The NCBI classification tree helps the researcher to evaluate exogenous sequence content armed with their biological knowledge of the experiment at hand. The researcher can also formulate hypotheses to test given the taxonomic classification displayed by MEGAN (Huson et al., 2007). The researcher can export all reads that were assigned to a specific taxon for further analysis including de novo assembly of transcripts from a given taxon.

### 3.4.5 *De novo* assembly using ABySS

In the final phase of RNA CoMPASS, the researcher can choose taxons of interest to extract reads from and assemble them into longer transcripts [Birol et al., 2009] using a de novo parallel sequence assembler. This process provides the researcher with a broader view of the particular transcripts that were found within a given taxon. De novo assembly can be repeated for each taxon of interest and the researcher can search the longer assembled transcripts against the databases again to get more precise hits.

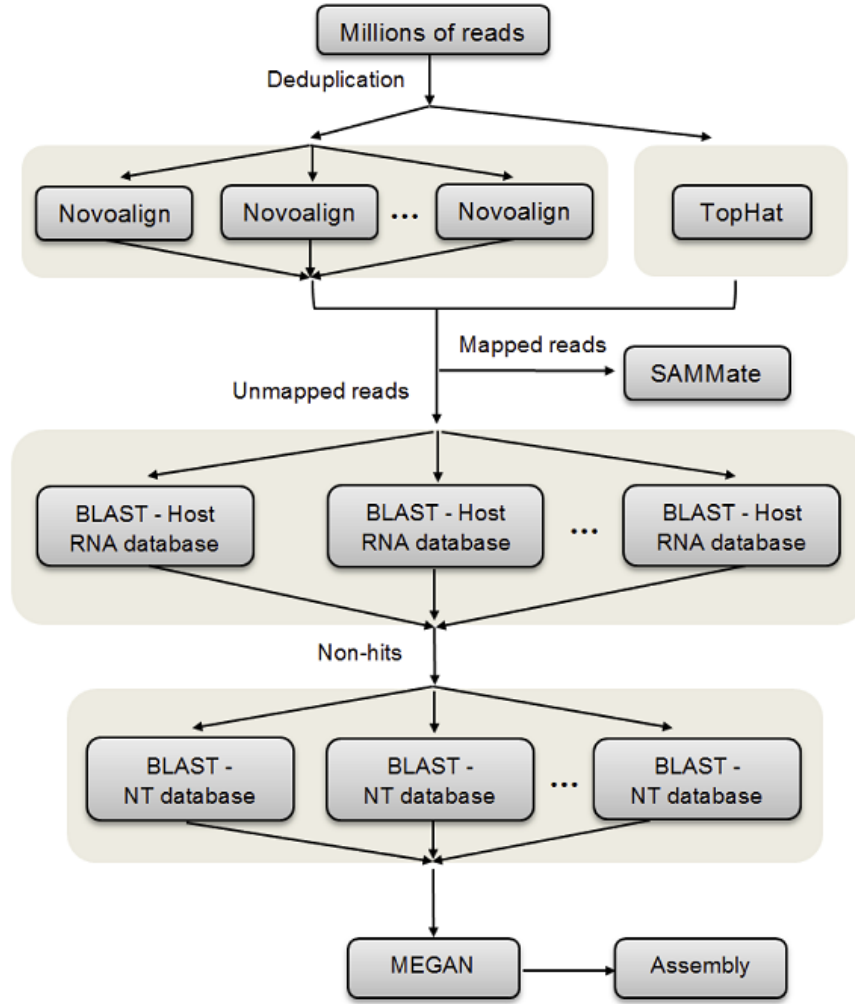


Figure 3.4 Overview of RNA CoMPASS workflow. After deduplication, raw sequence data in FASTQ format is aligned against the reference genome using Novoalign and TopHat. Reads which are mapped are used for endogenous analysis by SAMMate, and unmapped reads are used for exogenous analysis by searching against an optional RNA database and the NCBI NT database by BLAST. Taxonomical analysis is performed on the BLAST results using MEGAN and reads from a given taxon of interest can be extracted for assembly into longer transcripts by assembly tools.

### 3.4.6 Gene expression calculation

Genomic feature abundance score at gene level is performed via the SAMMate analysis pipeline [Xu et al., 2011]. Gene expression is calculated using Reads/Fragments Per Kilobase of exon model per million Mapped reads (RPKM/FPKM) [Trapnell et al., 2009].

$$\text{RPKM/FPKM} = 10^9 \times \frac{C}{L \times N}$$

$C$  is the total number of mapped short reads or fragments,  $L$  is the length of exons and  $N$  is the total number of short reads in one lane of one experiment. When scaled to range  $[0, 1000]$ , this value stands for the normalized depth of coverage for each gene.

For accurate calculation of the expression abundance score for annotated genes, we need a gene annotation file for the host genome to calculate the total length of exons. Using the uniquely mapped reads and repeat mapped reads from Novoalign along with splice junction reads identified by TopHat to count the total number of mapped reads on the exons of annotated genes. Then we are able to calculate the gene RPKM/FPKM value using the RPKM/FPKM formula.

### 3.4.7 Transcript expression calculation

Transcript quantification is computed via RAEM algorithm [Deng et al., 2011] and iQuant procedure [Nguyen et al., 2011] to accurately estimate the relative proportions and transcript abundance scores.

For RAEM algorithm, we infer an unobserved fragment-originating matrix ( $Z$  and  $Z'$ ) from the observed fragment-compatible matrix ( $Y$  and  $Y'$ ) for each gene. A row of the matrix represents one fragment and a column of the matrix represents one transcript. Then we implement the EM algorithm as the following [Deng et al., 2011],

$$\begin{aligned} \text{E-step:} \quad z_{i,j}^{(k+1)} &= \frac{y_{i,j} p_j^{(k)}}{\sum_{j=1}^J y_{i,j} p_j^k}, \quad \forall i, j \\ \text{M-step:} \quad n_j^{(k+1)} &= \sum_{i=1}^N z_{i,j}^{(k+1)}, \quad \forall j \\ \rho_j &= \frac{\mu p_j^{(K)} l_g}{l_j}, j = (1, 2, \dots, J), \end{aligned}$$



$i = (1, 2, \dots, I)$  is row index of the mapped fragments,  $j = (1, 2, \dots, J)$  is the column index of transcript index,  $l_j$  is the length of  $j$ th isoform and  $P = (p_1, p_2 \dots p_J)$ , and  $\sum_{j=1}^J p_j = 1$ .

Then the expression abundance for each transcript is:

$$\rho_j = \frac{\mu p_j^{(K)} l_g}{l_j}, j = (1, 2 \dots, J)$$

And  $l_g$  is the sum of total exon length of a gene, and  $l_j$  is the sum of total exon length of a transcript in this gene.

We implemented the EM algorithm [Deng et al., 2011] using Java language in RNA CoMPASS as the following,

```
/**
 * The following method is to estimate the proportions using EM procedures
 * @param E
 * @param gene
 * @return
 */
private Matrix calculate(Matrix E, double countedReadsNum, Gene gene)
{
    long isoNum = E.getColumnCount();
    long readsNum = E.getRowCount();
    Matrix newIsoProps = DenseMatrix.factory.ones(1, isoNum);
    newIsoProps = newIsoProps.divide(isoNum);
    Matrix z = DenseMatrix.factory.ones(1, isoNum);
    Matrix n = DenseMatrix.factory.zeros(1, isoNum);
    Matrix isoProps = DenseMatrix.factory.ones(1, isoNum);
    try
    {
        while(isoProps.minus(newIsoProps).normF() > 0.0001)
        {
            isoProps = newIsoProps.clone();
            n = DenseMatrix.factory.zeros(1, isoNum);
            for (int i = 0; i < readsNum; i++)
            {
                double temp = 0;
                for (int j = 0; j < isoNum; j++)
                {
                    z.setAsDouble(E.getAsDouble(i, j) * (isoProps.getAsDouble(0, j)), 0, j);
                    temp += z.getAsDouble(0, j);
                }
                for (int j = 0; j < isoNum; j++)
                {
                    if (temp == 0)
                        z.setAsDouble(0, 0, j);
                    else

```

```

        z.setAsDouble(z.getAsDouble(0, j)/temp, 0, j);
        n.setAsDouble(n.getAsDouble(0,j) + z.getAsDouble(0,j), 0, j);
    }
}
for (int j = 0; j < isoNum; j++)
    newIsoProps.setAsDouble(n.getAsDouble(0,j)/readsNum , 0, j);
}
} catch (Exception e)
{
    System.out.println("The gene cannot be calculated: " + gene.m_geneID);
    System.err.println(e.getMessage());
    return null;
}
gene.m_predictedExpValue = 1000000000 *
(double)gene.m_copies/(double)(gene.getTotalExonLength() * countedReadsNum);
gene.m_expressionValue = gene.m_predictedExpValue;
return newIsoProps;
}

```

For iQuant algorithm, we construct a relationship matrix of transcripts and exons for each gene from genomic annotations. In this matrix, each row represents one base of an exon and each column represents one transcript. Then we observe the depth at each base by calculating the coverage of mapped reads on genes. The mathematical framework is as the following [Nguyen et al., 2011],

$$E = W \cdot R + \varepsilon = \begin{bmatrix} p_1 & 0 & \dots & p_l \\ 0 & p_2 & \dots & p_l \\ \dots & \dots & \dots & \dots \\ p_1 & p_2 & \dots & 0 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & \dots & r_n \\ r_1 & r_2 & \dots & r_n \\ \dots & \dots & \dots & \dots \\ r_1 & r_2 & \dots & r_n \end{bmatrix} + \varepsilon$$

In the matrix  $E$ , it has  $N$  samples and  $M$  rows, each row corresponds one base of exons. Matrix  $W$  encodes the proportions between the isoforms and the output signals. Matrix  $R$  consists of samples of  $L$  isoforms.

Then we formulated this mathematical framework into a constrained convex quadratic problem:

$$\min ||E - WR||^2 \text{ s.t. } \sum_{i=1}^l p_i = 1, \quad p_i \geq 0, i = 1, 2, \dots, l.$$

It is known that the optimal solution to the above equality-only constrained estimation exists. And we implemented the iQuant algorithm [Nguyen et al., 2011] using Java language in RNA CoMPASS as the following,

```
/**
 * The following method is to estimate the proportions using one step procedure with
 * known gene expression score.
 * @param E
 * @param W
 * @param score
 * @param gene
 * @return proportions
 */
private Matrix oneStep(Matrix E, Matrix W, double score, Gene gene)
{
    long sampleNum = E.getColumnCount();
    long lengthExons = E.getRowCount();
    long isoNum = W.getColumnCount();
    Matrix S = W.clone();
    Matrix A = DenseMatrix.factory.ones(1,isoNum);
    int b = 1;
    Matrix z = E.toRowVector(Ret.NEW);
    Matrix isoProps = DenseMatrix.factory.ones(1,isoNum);
    try
    {
        Matrix H = DenseMatrix.factory.zeros(0,0);

        for (int i = 0; i < sampleNum; i++)
            H = H.appendVertically(S.times(score));

        Matrix temp0 = (H.transpose().mtimes(H));
        Matrix xHatLS = null;

        if (!temp0.isSingular())
            xHatLS = temp0.inv().mtimes(H.transpose()).mtimes(z);
        else
            return null;

        Matrix temp5 = (H.transpose().mtimes(H));

        if (temp5.isSingular())
            return null;

        Matrix B = temp5.inv();
        Matrix temp4 = (A.mtimes(B).mtimes(A.transpose()));

        if (temp4.isSingular())
            return null;

        Matrix part1 = B.mtimes(A.transpose()).mtimes(temp4.inv());
        Matrix temp1 = DenseMatrix.factory.eye(isoNum,isoNum);
        Matrix temp2 = temp1.minus(part1.mtimes(A)).mtimes(xHatLS);
        Matrix xHatCLS = temp2.plus(part1.times(b));
    }
}
```

```

        isoProps = xHatCLS.transpose();
        W = DenseMatrix.factory.zeros(lengthExons,isoNum);
        gene.setPredictedExpValue(score);
    } catch (Exception e)
    {
        System.out.println("The gene cannot be calculated: " + gene.m_geneID);
        System.err.println(e.getMessage());
        return null;
    }

    Matrix E_mean = DenseMatrix.factory.ones(lengthExons,1).times(E.getMeanValue());
    Matrix rs_v = DenseMatrix.factory.ones(isoNum,1).times(gene.m_predictedExpValue);
    Matrix dif1 = E.minus(W.mtimes(rs_v));
    Matrix dif1_square = dif1.transpose().mtimes(dif1);

    Matrix dif2 = E.minus(E_mean);
    Matrix dif2_square = dif2.transpose().mtimes(dif2);
    gene.m_Rsquare = 1 - dif1_square.doubleValue()/dif2_square.doubleValue();

    return isoProps;
}

```

### 3.4.8 Detection of differentially expressed genes and isoforms

In RNA CoMPASS, we allow users to detect differentially expressed genes and isoforms with support for edgeR. For each gene or transcript, we can count the total number of mapped reads and construct a matrix. In this matrix, each row represents one gene or one transcript, and each column represents one sample. The total number of column should be greater than 6. The entries of this matrix represent the total number of mapped reads for each gene or transcript. After we construct this matrix, we can import it to edgeR to calculate differentially expressed genes and isoforms.

	S1	S2	S3	S4	S5	S6
<i>gene 1</i>	$n_{11}$	$n_{12}$	$n_{13}$	$n_{14}$	$n_{15}$	$n_{16}$
<i>gene 2</i>	$n_{21}$	$n_{22}$	$n_{23}$	$n_{24}$	$n_{25}$	$n_{26}$
<i>gene 3</i>	$n_{31}$	$n_{32}$	$n_{33}$	$n_{34}$	$n_{35}$	$n_{36}$
<i>gene 4</i>	$n_{41}$	$n_{42}$	$n_{43}$	$n_{44}$	$n_{45}$	$n_{46}$
<i>gene 5</i>	$n_{51}$	$n_{52}$	$n_{53}$	$n_{54}$	$n_{55}$	$n_{56}$
...	...	...	...	...	...	...
<i>gene m</i>	$n_{m1}$	$n_{m2}$	$n_{m3}$	$n_{m4}$	$n_{m5}$	$n_{m6}$

### 3.4.9 Generation of reads coverage file for visualization

RNA CoMPASS allows users to generate a reads coverage file in wiggle format for other important visualization tools. For example, a wiggle file can be used for UCSC Genome Browser visualization of gene structure variation and a signal map file at single base pair resolution can be used for peak detection [Xu et al., 2011]. For each strand, RNA CoMPASS generates one array to save the coverage depth at each base for one chromosome. For example, we have 24 chromosomes and each chromosome has two strands. All these reads coverage information will be outputted into one text file following a standard wiggle format. The users can import the wiggle file to UCSC Genome Browser to visualize the gene structure variation.

```
track type=bedGraph name="read coverage reads_ADRB1.sam_coverage.wig"
chr10 0 59 22
chr10 59 115795342 0
chr10 115795342 115795402 1
chr10 115795402 115795435 0
chr10 115795435 115795495 1
chr10 115795495 115795664 0
chr10 115795664 115795722 1
chr10 115795722 115795724 2
chr10 115795724 115795726 1
chr10 115795726 115795782 2
chr10 115795782 115795786 1
chr10 115795786 115795910 0
chr10 115795910 115795970 1
chr10 115795970 115796080 0
chr10 115796080 115796140 2
chr10 115796140 115796334 0
chr10 115796334 115796394 2
```

Figure 3.5 Overview of coverage file in wiggle format.

## 3.5 Existing bioinformatics tools used in RNA CoMPASS

As the RNA sequencing technology has become an instrumental assay for transcriptome research. There is a plethora of quality bioinformatics tools available for mapping these reads to a host genome and performing additional downstream analyses. However, the management of many existing command-line tools can pose a challenge for researchers who may not be familiar with command-line tools or notions of distributed computing. To address this growing problem, we have developed a comprehensive system for RNA-seq analysis (RNA CoMPASS) that is

accessed via an easy to use web-based graphical user interface. In this pipeline, we have integrated many popular bioinformatics tools.

### 3.5.1 Novoalign

Novoalign is a highly accurate sequence alignment tool which is optimized for short sequences. We need to build an index file based on the reference genome using Novoindex script, and then we import the sequence data files by running Novoalign against the reference genome. The alignment results are restored a text file in SAM format. If the sequence data is paired-end, we need to input one pair of sequence files into Novoalign, and Novoalign will generate one alignment resulting file in SAM format.

Using different parameters of k-mer length and step size, the index file generated by Novoindex will have different size. For example, for searching the full human genome on a 16 GB RAM workstation, the recommended parameters are k-mer length = 14 and step size=2, the theoretical index size is around 13.5GB.

The step is very computationally intensive and we have parallelized this module in RNA CoMPASS to accelerate the processing. For the local cluster, we install Novoalign on each machine with Novoalign index files. Then we copy the executable shell scripts on each machine same as the Novoalign index files. The workhorse server allocates all tasks to each attached node machines, then the server collects the results from each node machine after the node machine finishes the task. After merging all results into one single file, the workhorse server will continue to precede the next step. For the grid system managed by Portable Batch System (PBS), we just need to copy the Novoalign executable file, related index files and some shell scripts to the grid system. Then we just need to turn on PBS feature by modifying a property in one configuration

file on the Tomcat web server. RNA CoMPASS can work with the grid system. Usually, a grid system has more than one hundred machines. That would greatly speed up the processing.

### 3.5.2 Bowtie

Bowtie is an ultrafast and memory-efficient short sequence aligner for aligning short sequences against large reference genome. Bowtie uses Burrows-Wheeler algorithm [Burrows and Wheeler, 1994] to build the index file, Burrows-Wheeler transformation (BWT) is a reversible permutation of the characters in a text. BWT-based indexing allows large texts to be searched more efficiently in memory usage. Bowtie performs a quality-aware, greedy, randomized, depth-first search through the space of possible alignments [Langmead et al., 2009]. In our pipeline, TopHat invokes Bowtie to align short sequences against reference genomes and constructs the contigs so that TopHat identifies all possible splice junction reads.

### 3.5.3 TopHat

TopHat is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to reference genomes using the short read aligner Bowtie, and then analyzes the mapping results to identify splice junctions between exons [Trapnell et al., 2009]. In RNA CoMPASS, all splice junctions can be used for the typical endogenous RNA-Sequencing analysis along with the investigation of exogenous sequences. After combine the exons reads from Novoalign and junction reads from TopHat, we can accurately calculate the genomic feature abundance at gene level and isoform level. Then we can filter the splice junction reads from unmapped reads, we can proceed to search unmapped reads against NCBI NT database using BLAST to categorize exogenous reads.

### 3.5.4 SAMMate

SAMMate, a Graphical User Interface (GUI) RNA-seq analysis pipeline, allows biomedical researchers to quickly process Fasta/Fastq and SAM/BAM files, and is compatible with both single-end and paired-end sequencing technologies. SAMMate automates some of more standard procedures in RNA-seq analysis [Xu et al., 2011]. This pipeline calculates genomic feature abundance score at both gene and isoform level and detects differentially expressed genes and isoforms with support for edgeR, and it also generates wiggle files for visualization and signal maps for peak detection and generates alignment report summarizing distribution of read mappings. In RNA CoMPASS, the human transcriptome quantification is performed by using SAMMate.

### 3.5.5 SAMtools

SAM Tools provide various utilities for manipulating alignments in the SAM format, including sorting, merging, indexing and generating alignments in a per-position format [Li and Handsaker et al., 2009]. In RNA CoMPASS, we use SAMtools to convert alignment results in BAM format outputted by TopHat into SAM format.

### 3.5.6 BLAST

BLAST is a nucleotide alignment tool from the BLAST+ suite of tools. It can search any length of sequence against a properly formatted database and there are several regularly updated versions of common databases available online. It can provide output in several formats with varying degrees of information. In RNA CoMPASS, BLAST output the results in hit table format and the BLAST results are further import into MEGAN to categorize exogenous sequences. This step is extremely computationally intensive. Therefore, we need to parallelize this module to minimize processing time. For the local cluster, we install BLAST on each



machine with BLAST database files. Then we copy the executable shell scripts on each machine same as the BLAST database files. The workhorse server allocates all tasks to each attached node machines, then the server collects the results from each node machine after the node machine finishes the task. After merging all results into one single file, the workhorse server will continue to process the next step. For the grid system managed by Portable Batch System (PBS), we just need to copy the BLAST executable files, related database files and some shell scripts to the grid system. Then we need to turn on PBS feature by modifying a property in one configuration file on the Tomcat web server. RNA CoMPASS can work with the grid system.

### 3.5.7 MEGAN

MEGAN is a metagenomic analysis tool which means it can also perform taxonomical analysis. MEGAN also offers two algorithms to classify sequences, SEED algorithm and KEGG algorithm. In RNA CoMPASS, we import BLAST results into MEGAN to generate a NCBI tree. Each node in this tree is labeled with a taxon and the size of a given node represents the number of sequences assigned to that taxon. This provides the researchers an overview of reads found in their data of possible exogenous origin. The researchers can export all reads that were assigned to a specific taxon for assembling into longer transcripts using ABySS.

### 3.5.8 ABySS

ABySS is a *de novo* parallel sequence assembler designed for short reads and large genomes. ABySS provides two version of assembler, single-processor and parallel version. The single-processor version allows assembling genomes up to 100 Mbp in size. The parallel version is capable of assembling mammalian-sized genomes using MPI. The output of ABySS is a set of contigs assembled from short reads (the input). In the final of phase of RNA CoMPASS, the

researchers can export reads from a given taxon of interest to assemble them into longer transcripts for further analysis.

## Chapter 4 Key Features

### 4.1 Introduction

In chapter 2, we introduced methods used in RNA CoMPASS, and in Chapter 3, its architecture. A unique challenge for working with RNA-seq data is to extract useful information from short reads stored in FASTQ format. Even though there are some existing bioinformatics tools designed for processing RNA-seq data, most of them are not very convenient for researchers because they are implemented with command-line interface. Besides, large studies also present computational challenges and it is not easy for researchers to overcome manually. In this section we introduce RNA CoMPASS which is a web-based GUI pipeline that allows biomedical researchers to comprehensively analyze RNA-seq data. A detailed documentation and a quick walkthrough are available at RNA CoMPASS's homepage [<http://rnacompass.sourceforge.net>]. RNA CoMPASS possesses the following key features (Figure 4.1): (1) Discovery and visualization of exogenous sequences of non-host origin. a) Aligns short RNA sequences against human, virus and bacterial genomes, b) Searches unmapped sequences against human RNA and NT databases, c) Visualizes taxonomic distribution of reads using MEGAN, d) Assembles pools of exogenous reads into longer transcripts with ABySS. (2) Performs extensive endogenous RNA-Seq analysis for a host organism. a) For RNA-seq alignment RNA CoMPASS uses short reads originating from both exons and exon-exon junctions to calculate gene expression scores. RNA CoMPASS's versatility allows biomedical researchers to combine the output from an exon alignment program, such as Novoalign [<http://www.novocraft.com/>], with the output of a splice junction analysis program, such as Tophat [Trapnell et al., 2009]. This intuitive combination results in a more accurate estimation of gene expression abundance scores. b) We have implemented two algorithms to estimate the

genomic feature abundance at isoform level: iQuant algorithm [Nguyen et al., 2011] and RAEM algorithm [Deng et al., 2011]. c) Using SAM or BED files generated from short read alignments, RNA CoMPASS implements an efficient and fast algorithm to calculate a base-wise signal map for peak detection analysis. d) RNA CoMPASS also exports a coverage file in wiggle format for visualization of alignment results on the UCSC genome browser. e) Lastly, RNA CoMPASS generates alignment report summarizing distribution of read mappings.

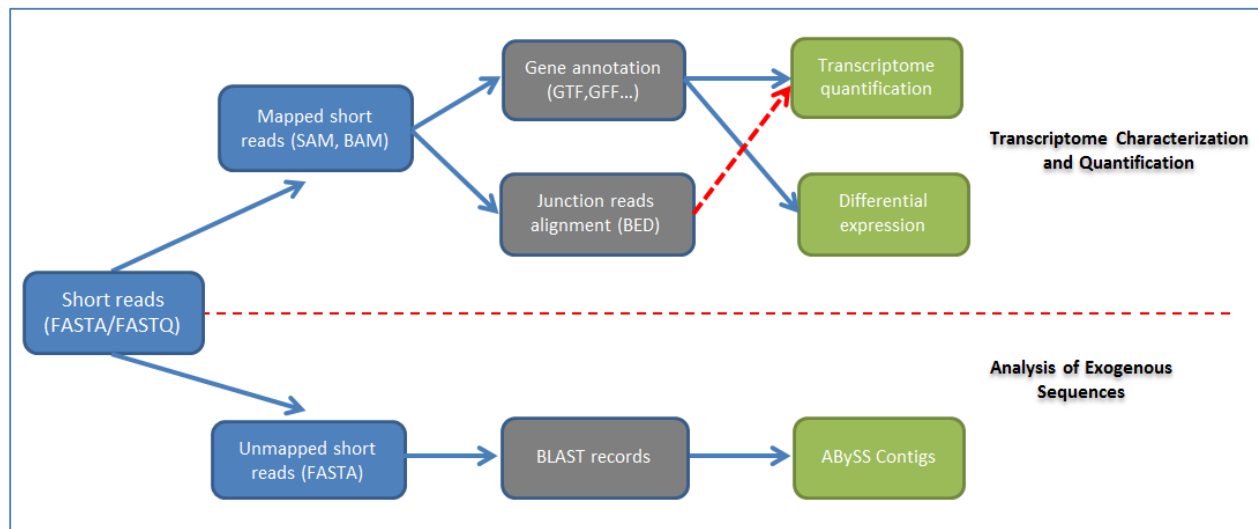


Figure 4.1 Key features of RNA CoMPASS: A schematic diagram of the two key features of RNA CoMPASS. (1) Discovery and visualization of exogenous sequences of non-host origin. (2) Performs extensive endogenous RNA-Seq analysis for the host organism.

## 4.2 Investigation of exogenous sequences of non-host origin

### 4.2.1 Feature: Alignment of short RNA sequences against human, virus and bacterial genomes

In RNA CoMPASS, the first phase is to perform the alignment of millions of short reads against the host genome using a very accurate aligner, Novoalign (<http://www.novocraft.com/>). In this phase, the administrator can build Novoalign index files with different reference genome, such as human, virus and bacterial genomes (Figure 4.2). Then the users can select different

combination of genome index files to perform the sequence alignment with Novoalign. RNA CoMPASS align the sequence data against these selected reference genomes and output the alignment results with the file names which append the suffix based on the related index file names. For example, the alignment result file name will be “sequenceFileName.fastq.sam.human” if run against human reference genome. After performed the Novoalign alignment phase, RNA CoMPASS will extract all non-mapped reads from Novoalign alignment resulting file and continue to perform the TopHat junction mapping phase if this option is checked. The junction mapper TopHat will output a junction mapping resulting file in SAM and BED format. This option is only designed for human reference genome because no alternatively spliced isoforms exist in virus and bacterial genomes. In the following step, RNA CoMPASS will filter out the human spans sequences from non-mapped reads which are extracted from Novoalign alignment resulting file. The rest of non-mapped reads will be used for the investigation of exogenous reads analysis.

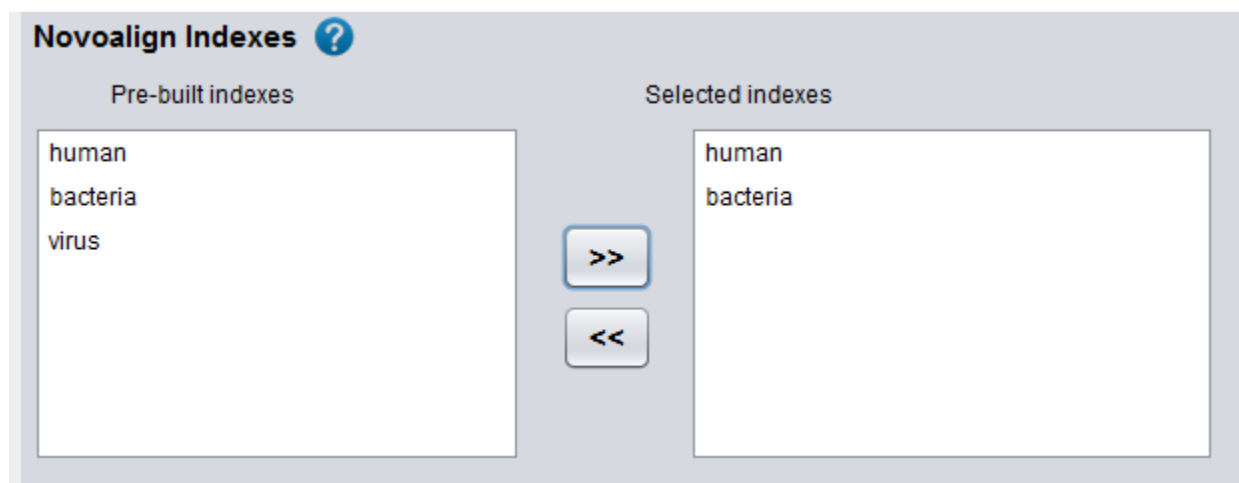


Figure 4.2 A screen shot of RNA CoMPASS. In the pre-built indexes selection panel, users can select multiple pre-built index files into system. RNA CoMPASS align the sequence data against the selected reference genomes with Novoalign.

#### 4.2.2 Feature: Searching unmapped sequences against human RNA and NT databases

RNA CoMPASS utilizes BLAST [Altschul et al., 1997] to search unidentified reads from the initial mapping stages against the human RNA database and NCBI NT database for identification. Many of the reads that were not mapped to the host genome in the first two stages of analysis are still identified by BLAST as mapping to the host genome. In order to reduce the computational burden further, RNA CoMPASS offers an optional stage prior to the BLAST against NT where the user can BLAST against a host transcript database to further filter these reads from consideration. After added an additional stage of BLAST against human RNA database, the time cost of searching unmapped sequences against NT database is greatly reduced. The feature of using Human RNA database is optional.

The users can run BLAST against Human RNA database and NT database with different e-value and other options. These options allow users conveniently increase the accuracy of hits by BLAST.

**BLAST Parameters** ?

☒ Use Human refseq RNA Database

e-Value:

☒ Use Soft Masking ☐ DUST Options

NT Database

e-Value:

☒ Use Soft Masking ☐ DUST Options

Figure 4.3 A screen shot of RNA CoMPASS. In BLAST Parameters panel, users can input different e-value for each step. The feature of using Human RNA database is optional.

### 4.2.3 Feature: Visualization of taxonomic distribution of reads using MEGAN

On average 87.5% of our reads from RNA-seq experiments in human are identified as mapped reads. About 12.5% of total numbers of reads are non-mapped reads. In previous studies, we often discarded the non-mapped reads and concentrated on the mapped reads for further analysis. However, important information could be lost by ignoring these unmapped reads. Since the human body is a persistent host to some bacterial organisms and viruses, some of the reads that do not map to the host genome could be indicative of bacterial or viral sequences. For the exogenous sequences analysis, at the first step, we extract non-mapped reads from alignment results then de-duplicate the redundant reads. After filtering out human spliced junction reads, we use BLAST to search unmapped sequences against human RNA database with specific e-values, and extract the non-hits sequences from the BLAST results against human RNA database and set the hits aside. Then we continue to run BLAST to search these non-hits sequences against NT database. By appending taxonomical ID as final column to BLAST results in hit table, we perform metagenomic analysis with MEGAN in the following step. MEGAN categorizes these exogenous sequences into many different categories and outputs a NCBI tree (Figure 4.4). In this tree, each node of output tree is labeled by a taxon and the size of a given node represents the number of reads assigned to that taxon. The researcher can extract reads from a taxon of interest to assemble them into longer transcript for further analysis using assembler, for example, ABySS. The pie chart is another representation of the MEGAN output (Figure 4.5). The MEGAN output and the pie chart is from the same sample. The pie chart represents the portion of reads that was assigned to each category (human, bacteria, etc).

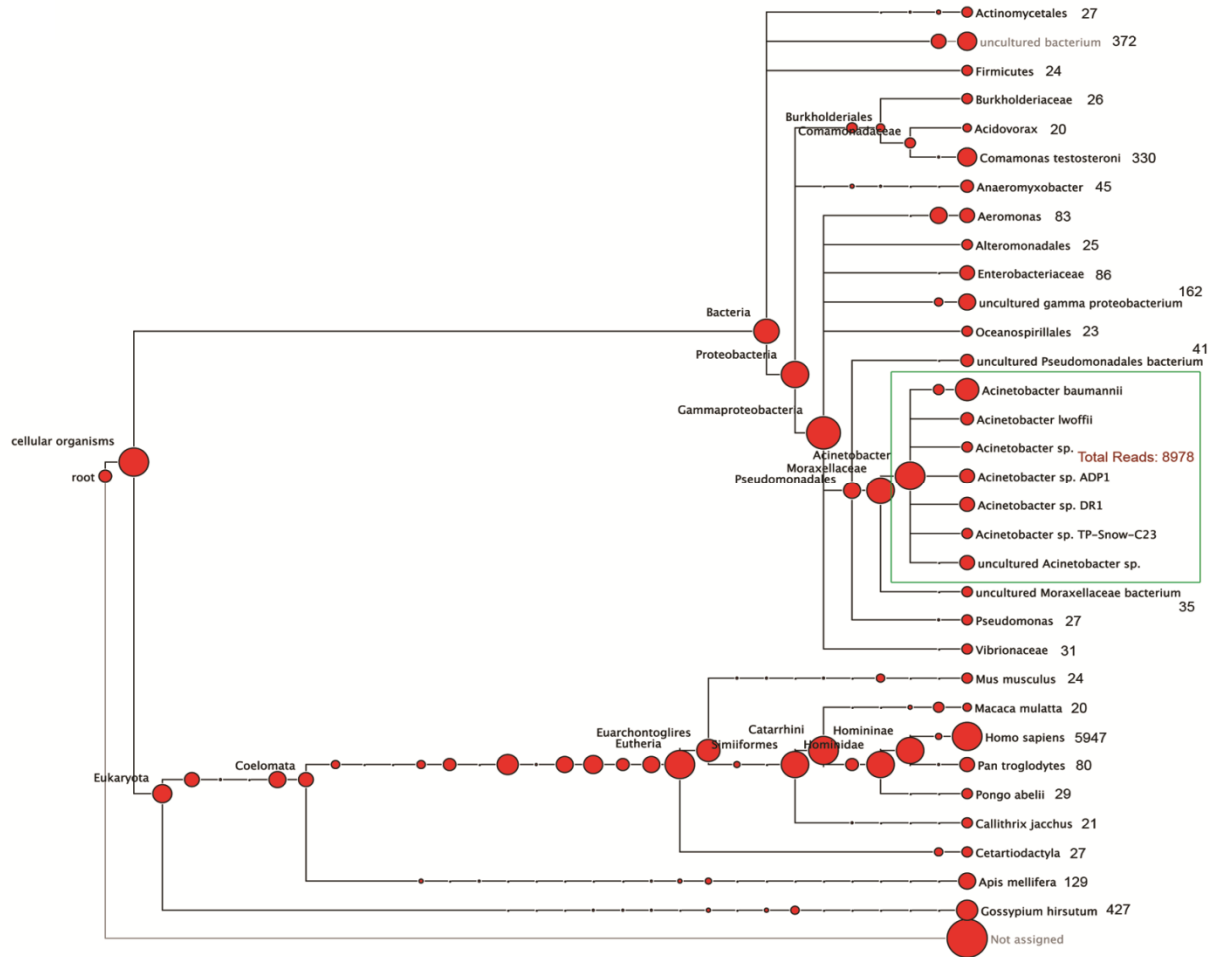


Figure 4.4 A NCBI tree output by MEGAN. Each node in this tree is labeled by a taxon and the size of a given node represents the number of reads assigned to that taxon.





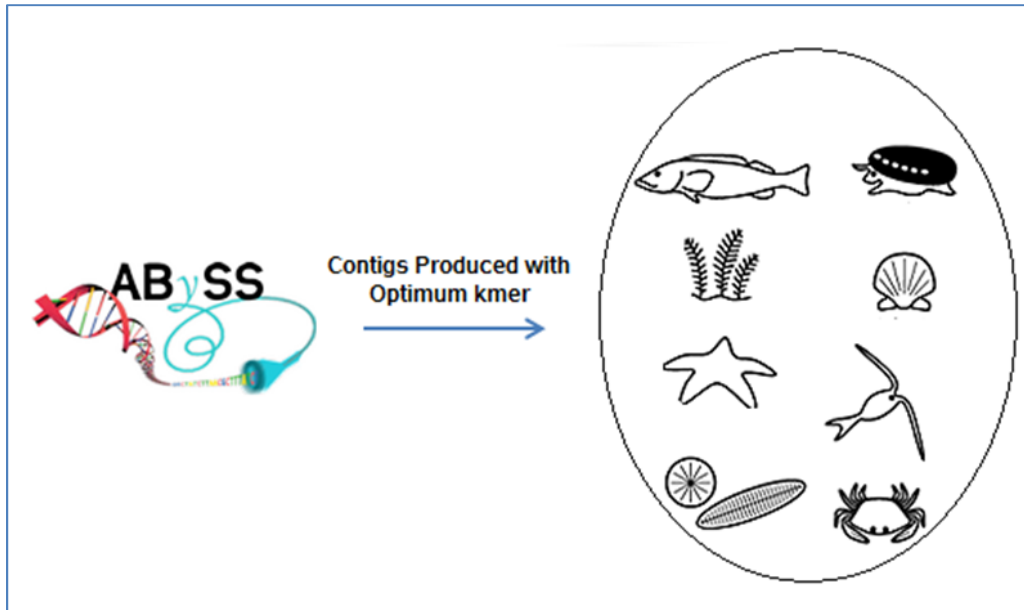


Figure 4.6 RNA CoMPASS uses ABySS to assemble reads that was assigned to each category into longer transcripts.

### 4.3 Performs extensive endogenous analysis for the host organism

#### 4.3.1 Feature: calculation of genomic feature abundance scores at gene level

Taking gene expression profiling using Illumina Genome Analyzer as an example, Read Per Kilobase of exon model per million Mapped reads (RPKM) was used to score gene expression abundance. The values obtained can be interpreted as the number of copies of each transcript in the living cell where the average length of transcripts is 1KB [Mortazavi et al., 2008]. The RPKM scores can range from  $< 0.01$  to  $> 10,000$ . There are now unprecedented and unparalleled opportunities to detect novel transcripts with ultra-low or ultra-high abundance.

A unique challenge for researchers working with RNA-seq data are short reads originating from exon-exon junctions in cDNA (around 10%). These short reads fail to map back to the reference genome since the exons are separated by introns (Figure 4.7). The millions of

unmapped short reads originating from exon-exon junctions, denoted as Initially Unmapped Reads (IUM's), need to be accounted for when calculating RPKM scores [Trapnell et al., 2009]. Unfortunately, most alignment tools are only able to map the short reads originating from exons completely ignoring IUMs in the process. Hereinafter, we denote such aligners as "exon aligner". To address the limitations of "exon aligners", ERANGE [Mortazavi et al., 2008], Tophat [Trapnell et al., 2009] and rSeq [Jiang et al., 2008] are among the recently developed approaches to map IUM's originating from exon-exon junctions back to individual genes. ERANGE uses a union of known and novel junctions while Tophat *de novo* assembles IUM's using a module in MAQ [Li et al., 2008]. Hereinafter, we denote an aligner of this type as "junction mapper". Thus, there are now two types of aligners that complement each other.

Performance-wise, aligners vary vastly in accuracy as well as the underlying algorithms used. It is highly desirable for RNA-seq data analysis to allow users the freedom to choose and combine a pair of their favorite exon aligner and junction mapper to estimate gene expression scores. RNA CoMPASS fulfills this role by calculating and exporting a gene expression score matrix using a user-defined combination of an exon aligner and a junction mapper (Figure 4.8). RNA CoMPASS then calculates the gene expression RPKM or FPKM score for gene  $i$ ,  $R_i$  as  $R_i = \frac{10^9(C_i^A + C_i^B)}{NL_i}$ ; where  $i$  represents the gene index.  $C_i^A$  is the short read counts uniquely mapped to exons using an exon aligner (e.g. Novoalign), and  $C_i^B$  is the IUM short read counts uniquely mapped to the exon-exon junctions using a junction mapper (e.g. Tophat).  $N$  represents all uniquely mapped read counts in a cell extract sample, and  $L_i$  is the summation of the exon lengths. Thus, RNA CoMPASS combines short reads mapped to exons (e.g. available in SAM/BAM format) and to exon-exon junctions (e.g. available in BED format) to accurately estimate gene expression scores (Figure 4.8).

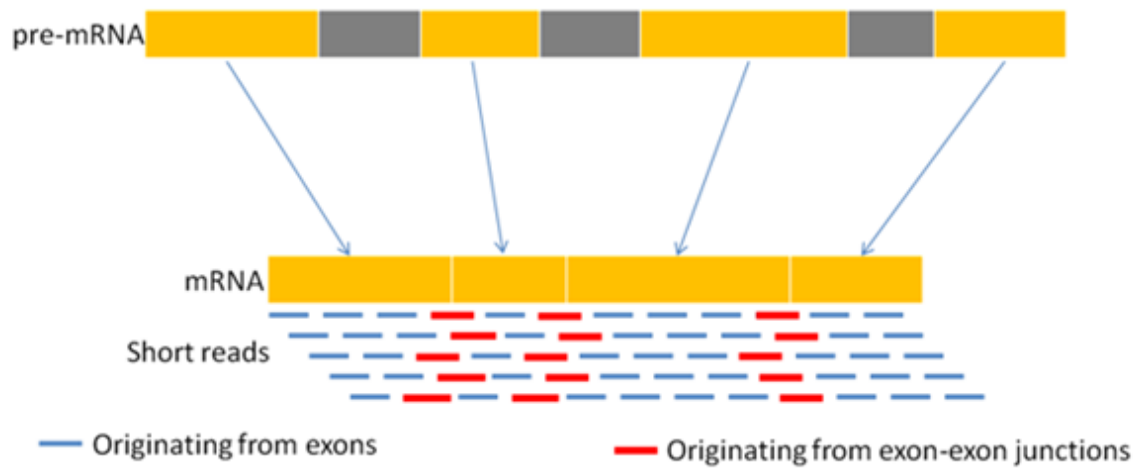


Figure 4.7 A unique challenge for researchers working with RNA-seq data. The junction reads (red) fail to map back to the reference genome because exons are separated by introns.

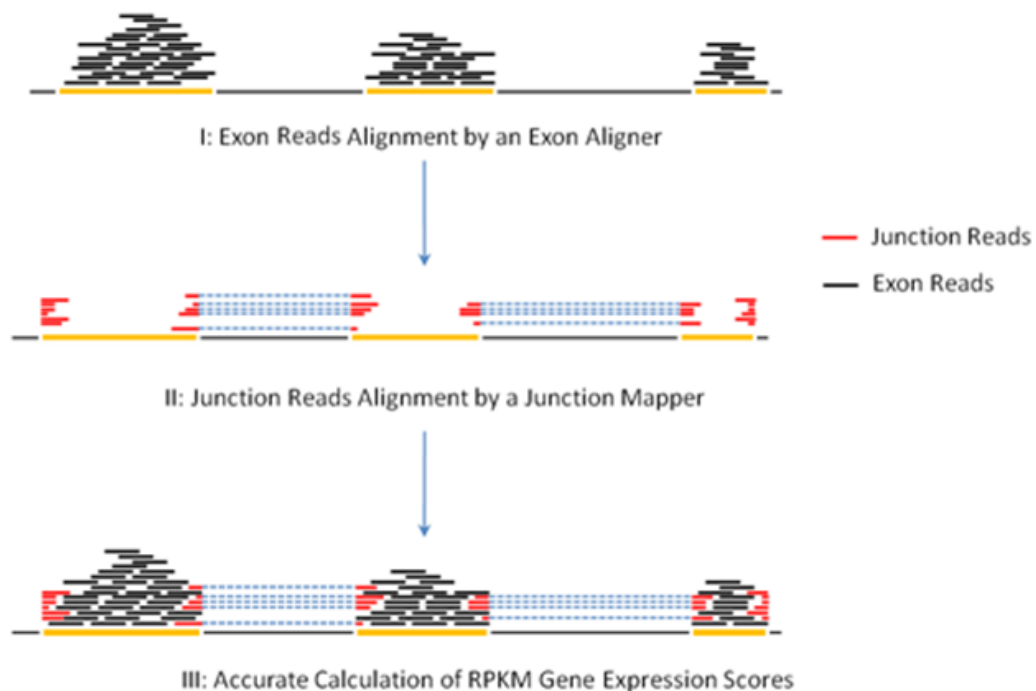


Figure 4.8 Combination of exon reads with junction reads to accurately calculate gene expression RPKM scores. A demonstration of the ideas of combining exon reads (black) and junction reads (red) to calculate gene expression RPKM scores.

RNA CoMPASS can also take many pairs of SAM(BAM)/BED files simultaneously, one for each cell sample, to calculate a Microsoft EXCEL compatible gene expression matrix. In this matrix rows correspond to genes or the customized genome coordinate intervals, and columns correspond to different cell samples. It must be noted that RNA CoMPASS is more flexible and accurate than other software, such as Tophat [Trapnell et al., 2009], that also export the gene expression scores. We validate our claim using experimental data obtained from 3' UTR assay as a case study shown below. RNA CoMPASS's reporting utility for gene expression abundance score is also quite versatile as this utility is not limited to the annotated genes. In fact, RNA CoMPASS calculates genomic feature abundance scores for any user-defined genomic intervals. This utility dramatically simplifies the technical burdens for discovering novel genes.

#### 4.3.2 Feature: calculation of genomic feature abundance scores at isoform level

Isoform quantification using RNA-seq is central to a wide range of transcriptomics research. The problem itself is challenging due to the fact that the observed exonic expression signal can be aggregated from a set of sibling isoforms encoded by the same gene with diverse alternative splicing mechanisms. In RNA CoMPASS, we have implemented two algorithms iQuant algorithm [Nguyen et al., 2011] and RAEM algorithm [Deng et al., 2011] to quantify transcript abundance score at isoform. For iQuant algorithm, we provided one-step procedure and iterative procedure to estimate genomic feature abundance scores at isoform level. One-step procedure can quickly calculate the genomic feature abundance score at isoform level but less accuracy than iterative procedure and RAEM algorithm [Deng et al., 2011]. For RAEM

algorithm [Deng et al., 2011], it is implemented using iterative Expectation-Maximization algorithm to accurately calculate genomic feature abundance scores at isoform level.

### 4.3.3 Feature: generation of signal map for peak detection

A signal map is a frequently demanded data format for NGS data analysis. In a signal map file, alignment results are represented in the per-base "pileup" format. In this format the single nucleotide short read coverage depth is calculated whereas the whole genome coverage is provided as a vector of integers with length  $3.2 \times 10^9$ . A signal map is a common input for a number of frequently performed sequential analyses to detect a wide range of genomic features. For ChIP-seq and Methyl-seq data, significant peaks in a signal map may indicate potential transcription factor binding sites and DNA methylation sites, respectively. For DNA-seq data, significant change points in the signal map might indicate a true copy number change, which is often a hallmark of cancer [Chen et al., 2009].

### 4.3.4 Feature: generation of wiggle files for visualization

Biomedical researchers also need to visualize the alignment results in order to examine possible gene structure alterations between case and control studies. For example, shortened 3'-UTR's in cancer cells are reflected as an abrupt dropout of the short read coverage. This visualization need is addressed by another key feature of RNA CoMPASS. RNA CoMPASS can take the alignment results and export the genome information to wiggle (.wig) files where the wiggle format is compatible with the UCSC genome browser and other browsers used for visualization. This feature will allow biomedical researchers to visually check the alignment quality of selected genes in selected genomic regions. For the miRNA-155 target prediction research, Figure 4.9 and Figure 4.10 presents two typical scenarios: the left and right panels

show the alignment results in the pile-up format for gene CXorf39 on Chromosome X and gene LBA1 on Chromosome 3, respectively. Figure 4.9 indicates no overall expression change in the codon regions, but a significant dropout in the 3'-UTR region occurs. On the contrary, Figure 4.10 shows no significant difference in the 3'-UTR region but a significant difference in the codon region instead. These two examples demonstrate RNA CoMPASS's ability to generate wiggle files for biomedical researchers allowing them to visually look for possible gene structure alterations. While there are a number of existing alignment visualization software (e.g. [Bao et al. 2009; Arner et al. 2010]), these systems do not allow many annotation tracks in parallel, which is the deterministic feature for knowledge discovery.

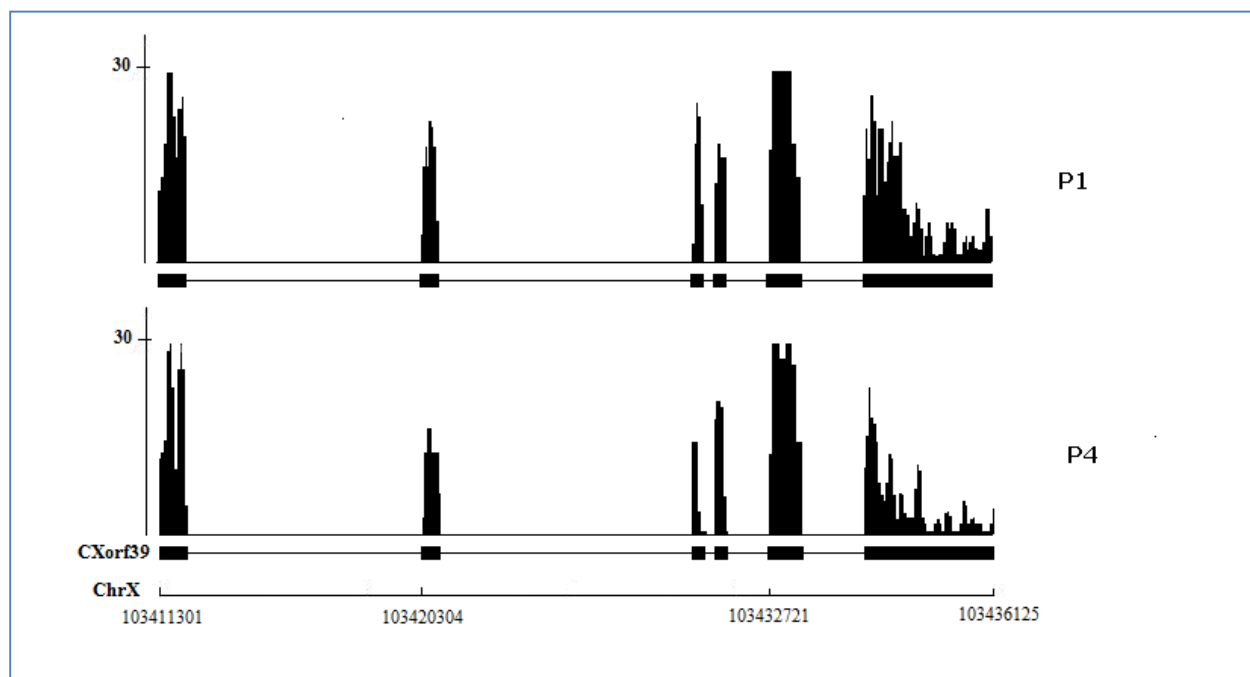


Figure 4.9 Visualization of gene structure variation. Gene CXorf39 was called by the Change Point Analysis as a potential miRNA-155 target due to its abrupt read dropout on the 3'-UTR end.

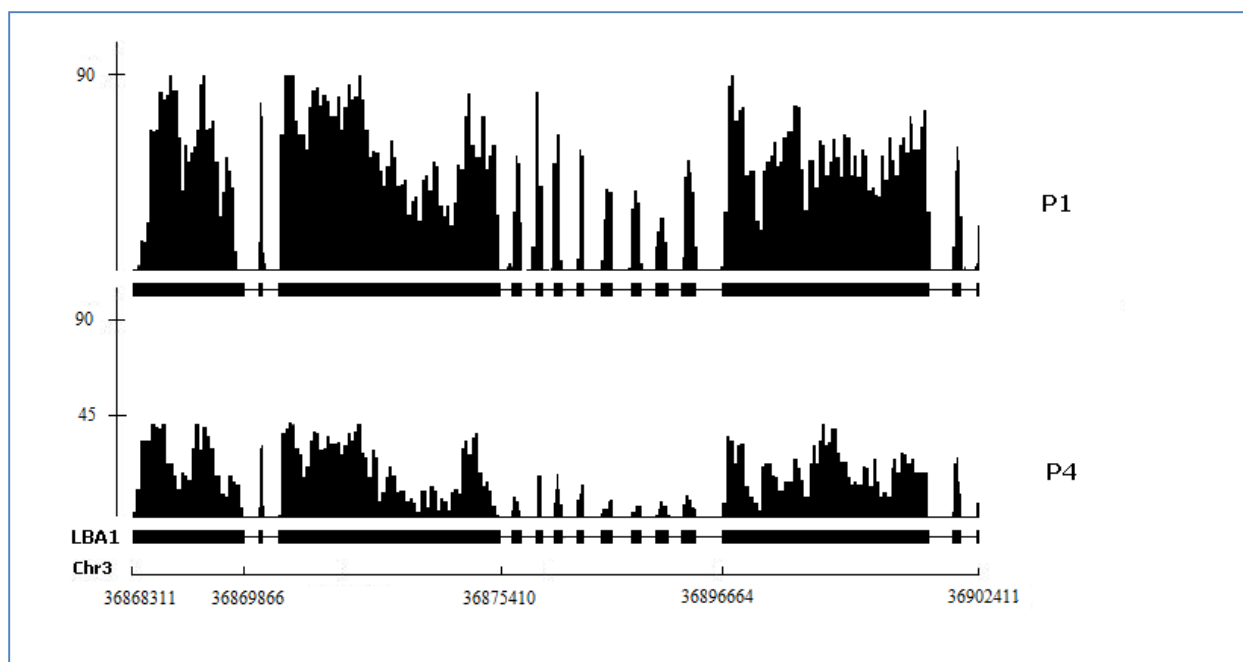


Figure 4.10 Visualization of gene structure variation. Gene LBA1 was called by the Differential Expression Analysis as a potential miRNA-155 target due to the overall read coverage decrease in codon region.

#### 4.3.5 Feature: generation of alignment report

Short read alignment statistics provide indispensable resources to examine the alignment quality as well as comparing alignment results. RNA CoMPASS calculates and exports a number of alignment statistics including the percentage of uniquely mapped short reads and the percentage of short reads mapped to intergenic, exonic and intronic regions.

##### **SAMMate 2.6.1 Result Report:**

Name	sample-single-end-GroupA-1.sam	sample-single-end-GroupB-1.sam
Reads Number On Exon	46230 (85.24173%)	46195 (85.196045%)
Reads Number On Intron	4342 (8.006048%)	4360 (8.041017%)
Reads Number On Junction	0 (0.0%)	0 (0.0%)
Reads Number On Intergenic	3662 (6.752222%)	3667 (6.7629375%)
Total Reads Number	54234	54222

Figure 4.11 The overview of alignment report.



## Chapter 5 Performance Results

### 5.1 The performance comparison in analyzing human organism dataset

To evaluate the performance of RNA CoMPASS in analyzing large datasets, we chose datasets to benchmark the performance on both a local cluster with 3 node machines and a grid system. On the local cluster, we processed two benchmark datasets. SRR032238 and SRR032246 were generated on the Illumina Genome Analyzer platform in a human experiment with 50bp reads (NCBI Short Read Archive, Accession Number SRA010302). In these two samples, the size of each file is more than 5GB and the total number of raw sequence reads is approximately 20 million (Table 5.1).

	<b>SRR032238 (5GB)</b>	<b>SRR032246 (5GB)</b>
<b>Reads Number</b>	24,221,278	22,161,215
<b>Unique Maps</b>	19,337,768	17,413,299

Table 5.1. Overview of samples SRR032238 and SRR032246. For each sample, total number of reads contained in the file and the number of unique mapped reads aligned by Novoalign is shown.

We used FastQC to scan the sample files SRR032238 and SRR032246 to check the validation of the data. From Figure 5.1, we used the box plot for the sample SRR032238 at each base to check the quality distribution. The first 23 base of read has high quality and then the qualities of read are gradually decreased after this point. From the Figure 5.2, we plot the GC distribution over all sequences. The curve of the GC count per read is very close to that of theoretical distribution. We also plot the quality distribution and GC distribution over all sequences for the sample file SRR032246 in Figure 5.3 and Figure 5.4. These figures show that our sample file SRR032238 and SRR032246 are valid to use for comparing the performance in analyzing human organism.

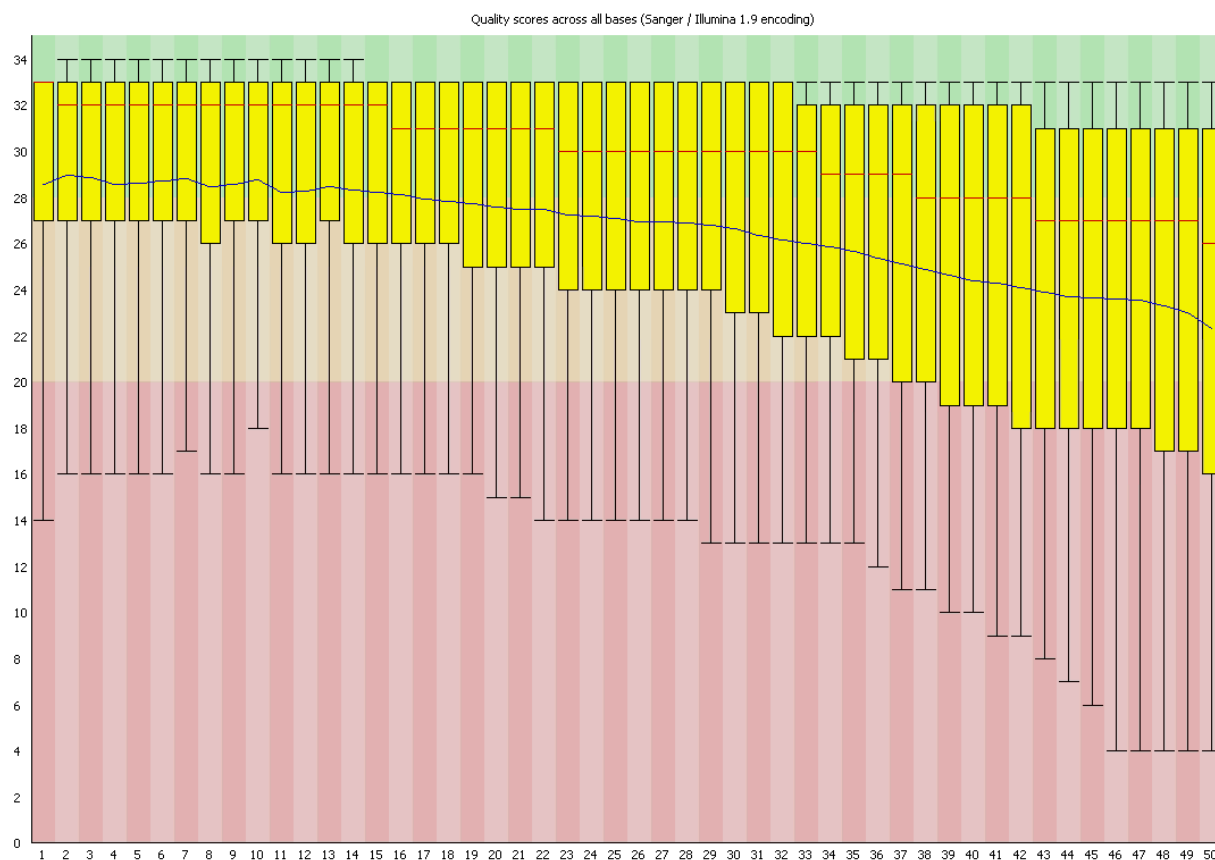


Figure 5.1 The box plot of quality scores across all bases for the sample file SRR032238. The horizontal axis corresponds to the base position of sequence. The vertical axis corresponds to the quality score.

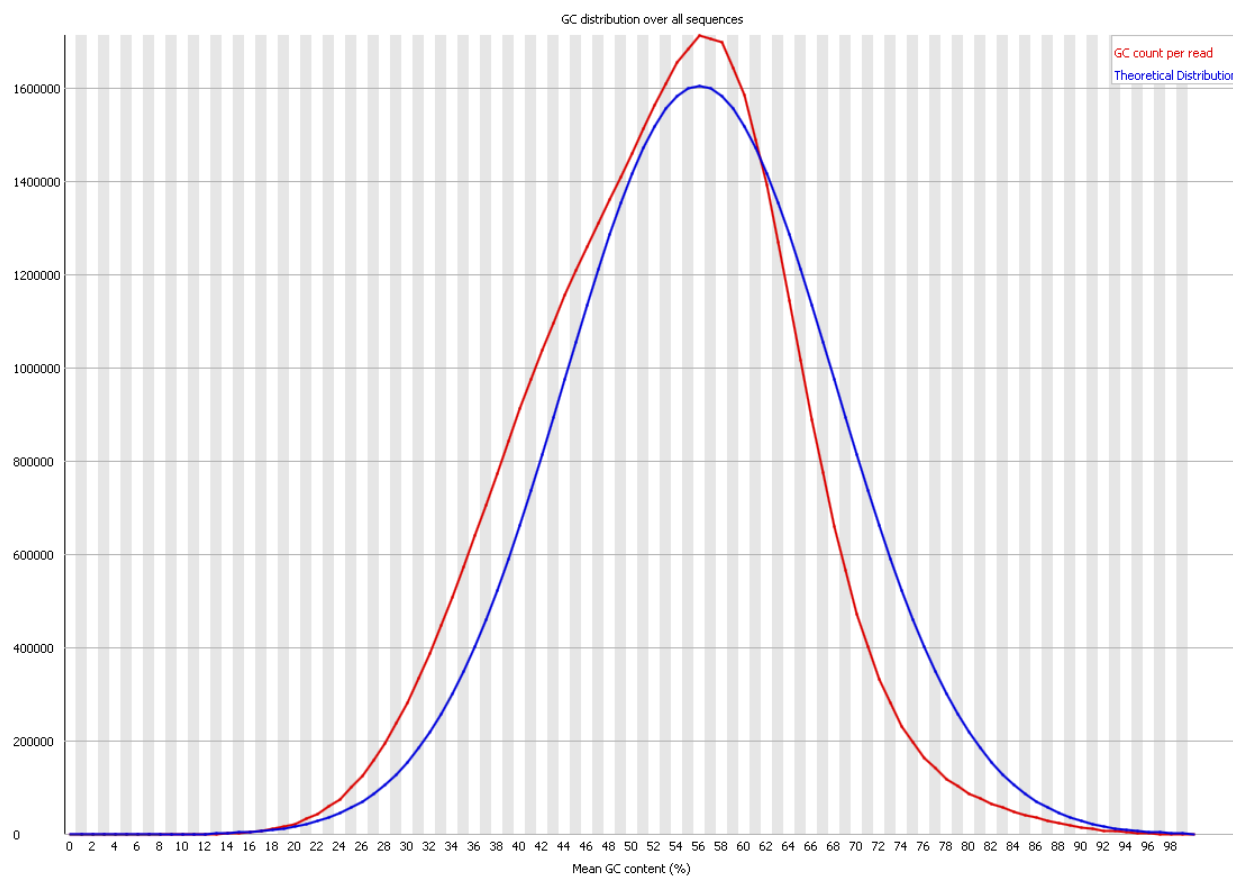


Figure 5.2 The overview of GC distribution over all sequences for the sample file SRR032238. The curve marked by red color is GC count per read and the curve marked by blue color is the theoretical distribution. The horizontal axis corresponds to mean GC content (%). The vertical axis corresponds to the number of GC count.

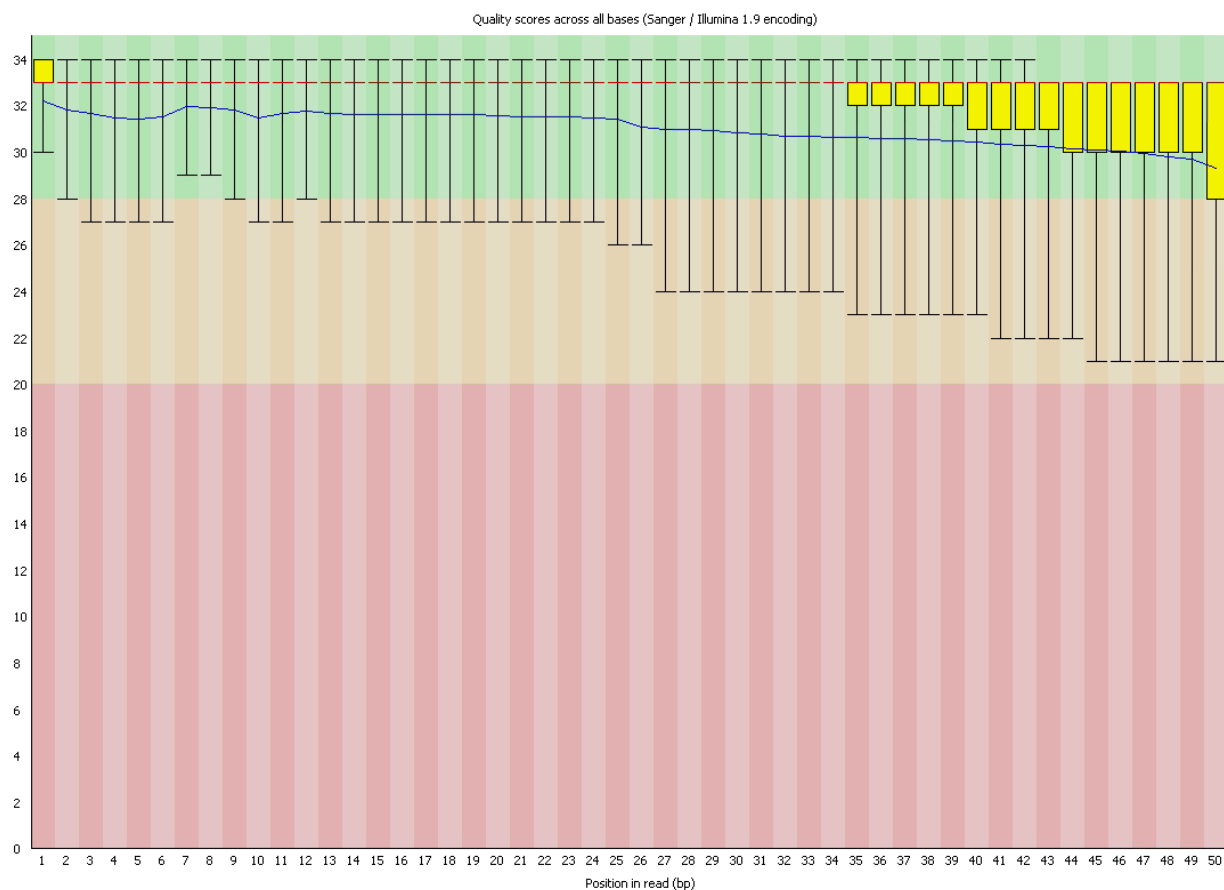


Figure 5.3 The box plot of quality scores across all bases for the sample file SRR032246. The horizontal axis corresponds to the base position of sequence. The vertical axis corresponds to the quality score.

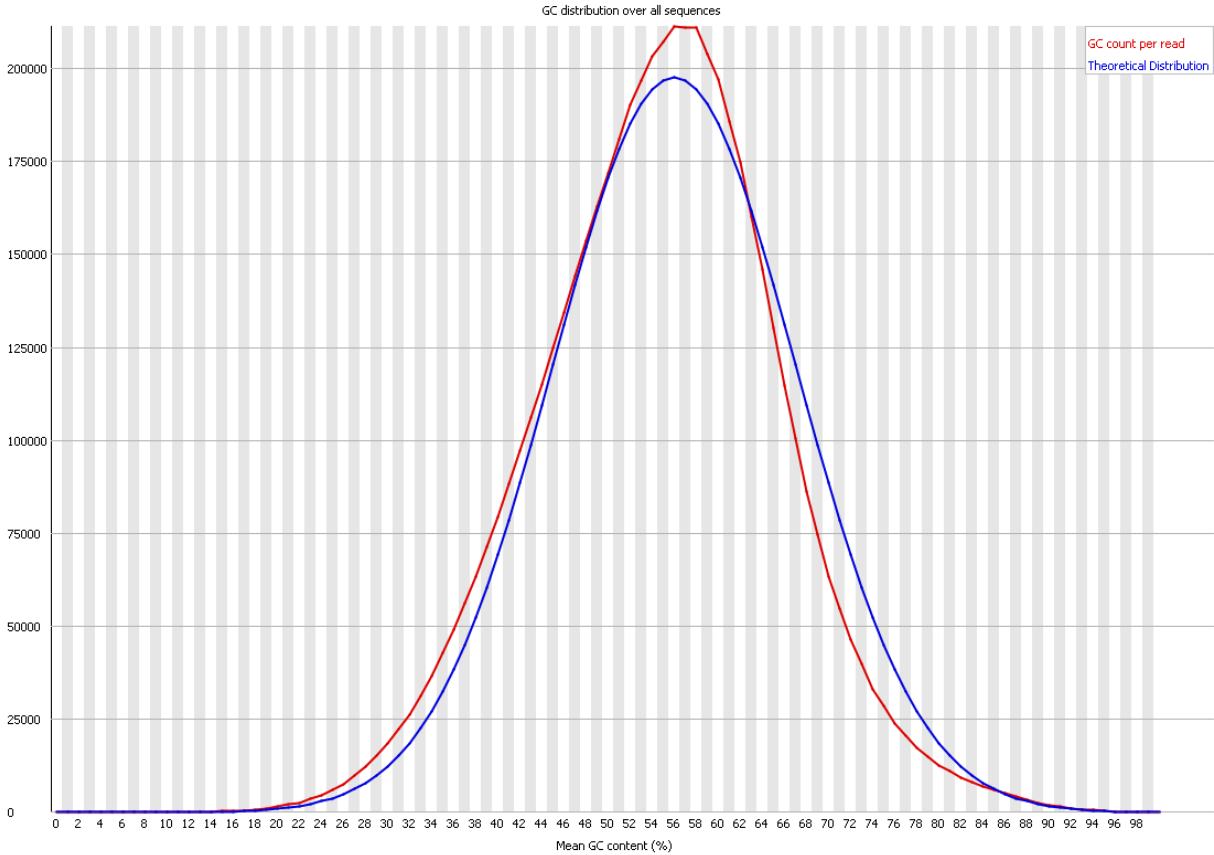


Figure 5.4 The overview of GC distribution over all sequences for the sample file SRR032246. The curve marked by red color is GC count per read and the curve marked by blue color is the theoretical distribution. The horizontal axis corresponds to mean GC content (%). The vertical axis corresponds to the number of GC count.

These data sets were processed in the same fashion on a single machine and using a local cluster of 3 machines. Figure 5.5 shows that the performance of the Novoalign and BLAST modules, which have been parallelized, has a speeded up of nearly  $N$  times (where  $N$  is the number of node machines in the parallel run). The performance of other non-parallelized modules has no speedup. The overall speedup of the entire process is roughly 2 times with only 3 machines in the cluster. For further comparing the performance, we have presented a detailed comparison of the performance between a single machine and local cluster of three machines as the file size increases in section 5.3.

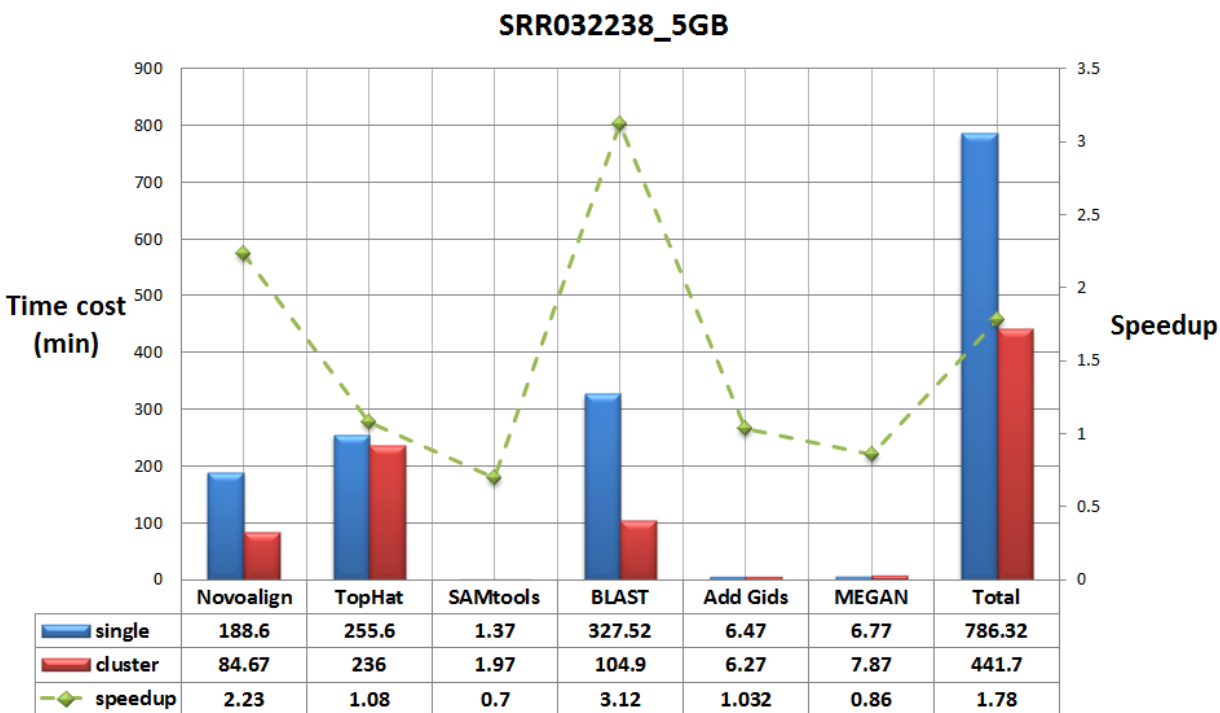


Figure 5.5 Performance and speed up for sample SRR032238 running on a local cluster with 3 node machines. The horizontal axis corresponds to the modules used in RNA CoMPASS. The left vertical axis shows the run time of each module and total time spent processing the sample. The right vertical axis shows the corresponding speedup of the parallelized version for each module.

## 5.2 The performance comparison in analyzing non-human organism dataset

In order to show the capability of RNA CoMPASS to process other non-human organisms, we also performed RNA CoMPASS on a grid system with a benchmarked dataset. SRR006514 is generated from the Illumina Genome Analyzer platform in a *caenorhabditis elegans* experiment with 36bp reads (NCBI Short Read Archive, Accession Number SRA003622). The size of the sample file is 1.6GB with about 10 million of reads. This dataset was run on both a single machine of the grid and on the grid system with 24 cores allocated for processing.

We used FastQC to scan the sample files SRR006514 to check the validation of the data. From Figure 5.6, we plot the box plot for the sample SRR006514 at each base to check the quality distribution. The first 10 base of read has high quality and then the qualities of read are gradually decreased after this point. From the Figure 5.7, we plot the GC distribution over all sequences. The curve of the GC count per read is very close to that of theoretical distribution. These figures show that our sample file SRR006514 is valid to use for comparing the performance in analyzing non-human organism.

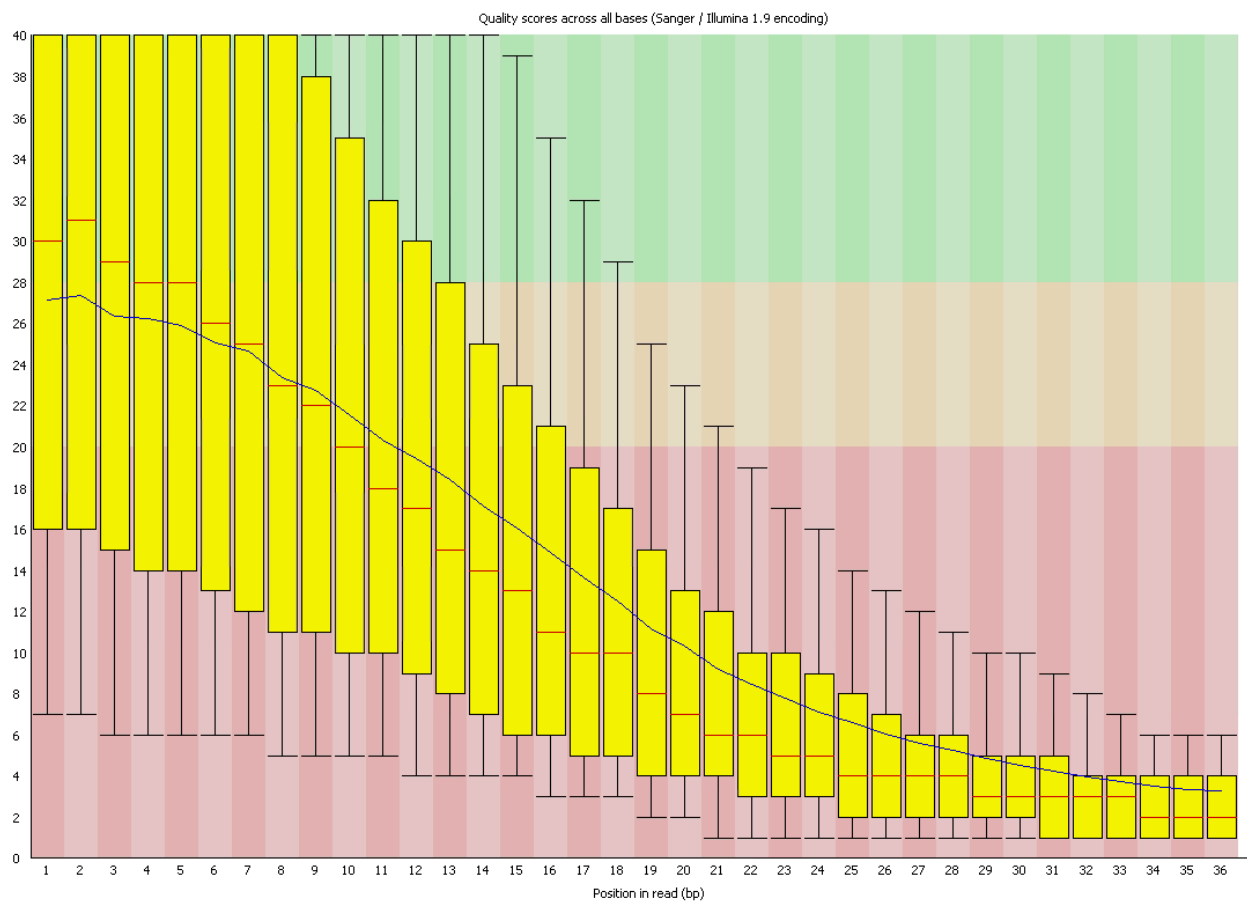


Figure 5.6 The box plot of quality scores across all bases for the sample file SRR006514. The horizontal axis corresponds to the base position of sequence. The vertical axis corresponds to the quality score.

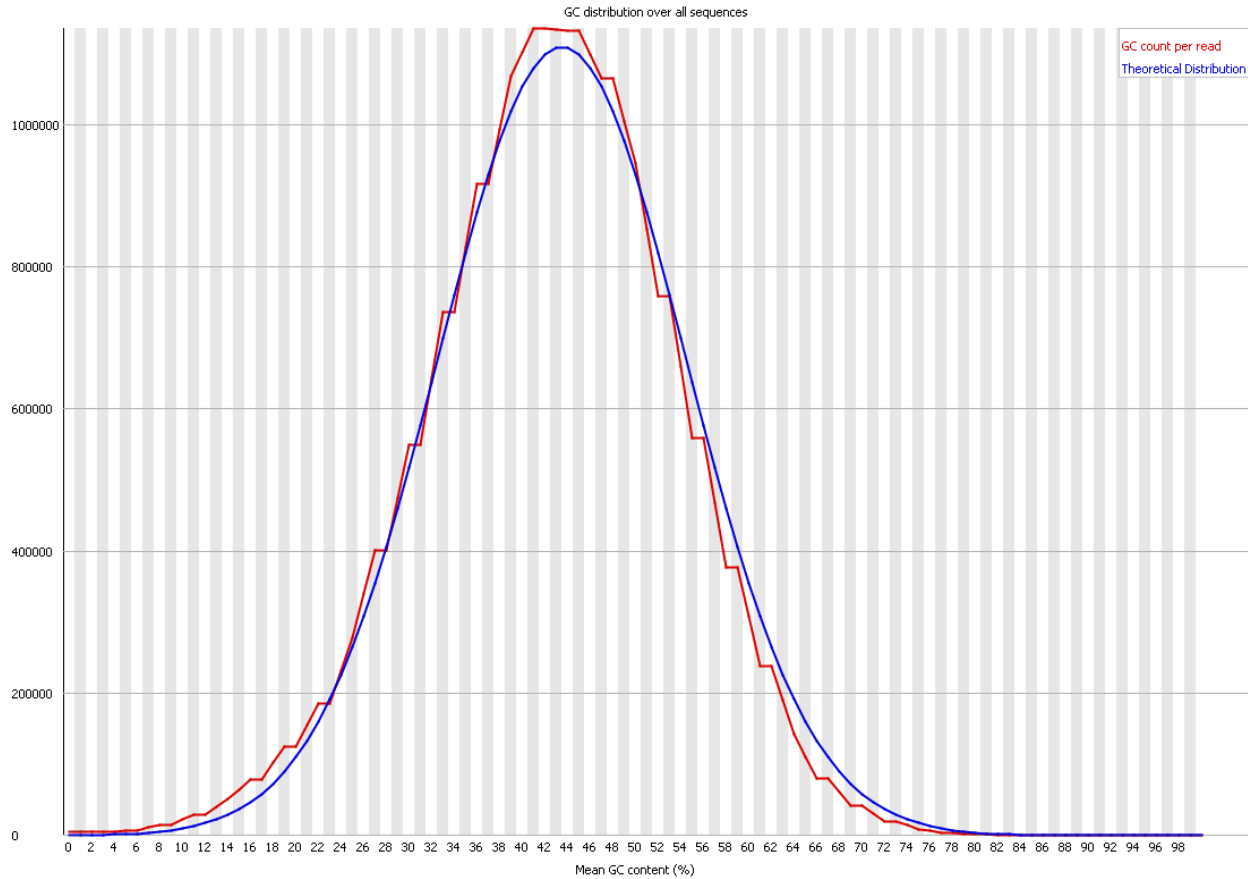


Figure 5.7 The overview of GC distribution over all sequences for the sample file SRR006514. The curve marked by red color is GC count per read and the curve marked by blue color is the theoretical distribution. The horizontal axis corresponds to mean GC content (%). The vertical axis corresponds to the number of GC count.

Figure 5.8 shows the performance of parallelized Novoalign and BLAST modules has a speedup of roughly  $\sqrt{N}$  times (where  $N$  is the number of cores used). The performance of other non-parallelized modules has a very slight change. Since the grid system is managed by PBS submission, the speedup can be affected by many factors including internet congestion, the load on the grid and the length of the PBS queue. In section 5.3, we have also presented a detailed comparison of the performance between single machine and a grid system as the file size increases.



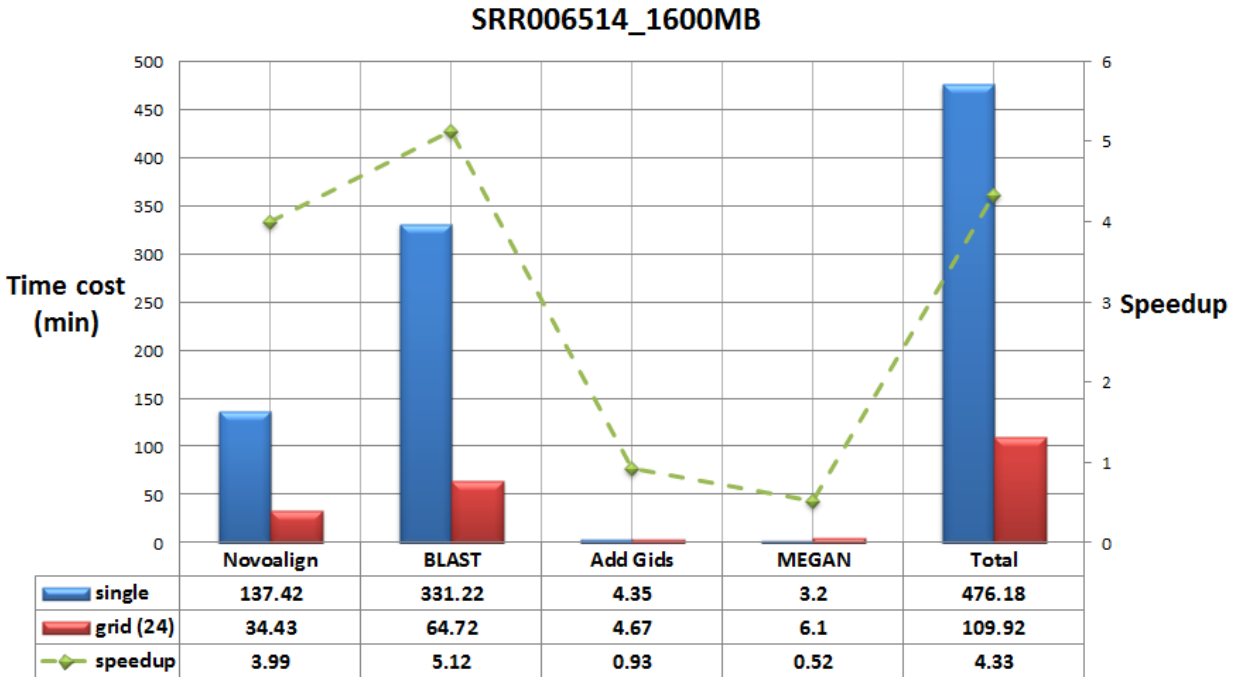


Figure 5.8 Performance and speed up for sample SRR006514 running on a grid system with 24 cores. The horizontal axis corresponds to the modules used in RNA CoMPASS. The left vertical axis shows the run time of each module and total time spent processing the sample. The right vertical axis shows the corresponding speedup of the parallelized version for each module.

### 5.3 The performance comparison between time cost and speedup on local cluster

To further assess the performance of RNA CoMPASS in analyzing large data sets, we performed a detailed comparison of the run time and speedup achieved as the file size increases. Each of the benchmarked data sets SRR032238 and SRR032246 (Table 5.2 and Table 5.3) from human experiment was split into three pieces. These files were then analyzed using RNA CoMPASS on both a single machine and a small local cluster with 3 node machines. Figure 5.9 and Figure 5.10 show that the run time of Novoalign on single machine is nearly twice as long as on local cluster using three machines. The run time of BLAST on single machine is nearly three times as long as on the local cluster with three machines. The performance of other non-parallelized modules has no significant change. This shows that the testing results between

sample SRR032238 and SRR032246 are consistent and parallelized modules can significantly accelerate the speed of processing a large data set up to the theoretical limit of N.

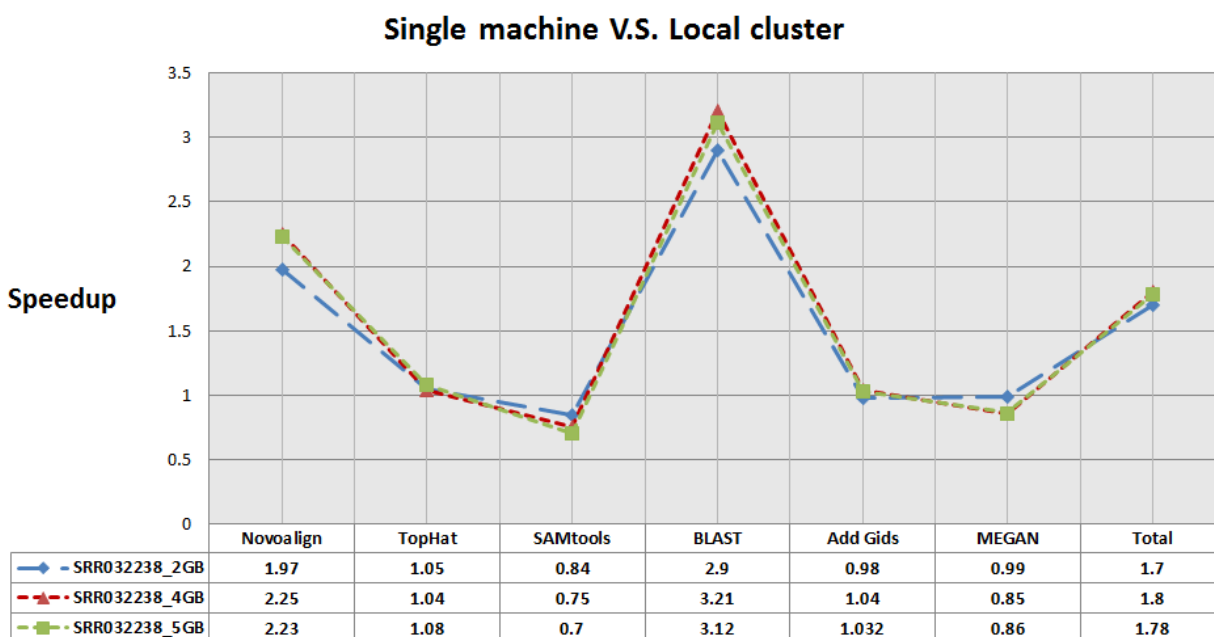


Figure 5.9 Performance of a single machine versus a local cluster with three machines for sample SRR032238. The horizontal axis shows the modules used in RNA CoMPASS. The vertical axis shows the speedup of each module for this sample.

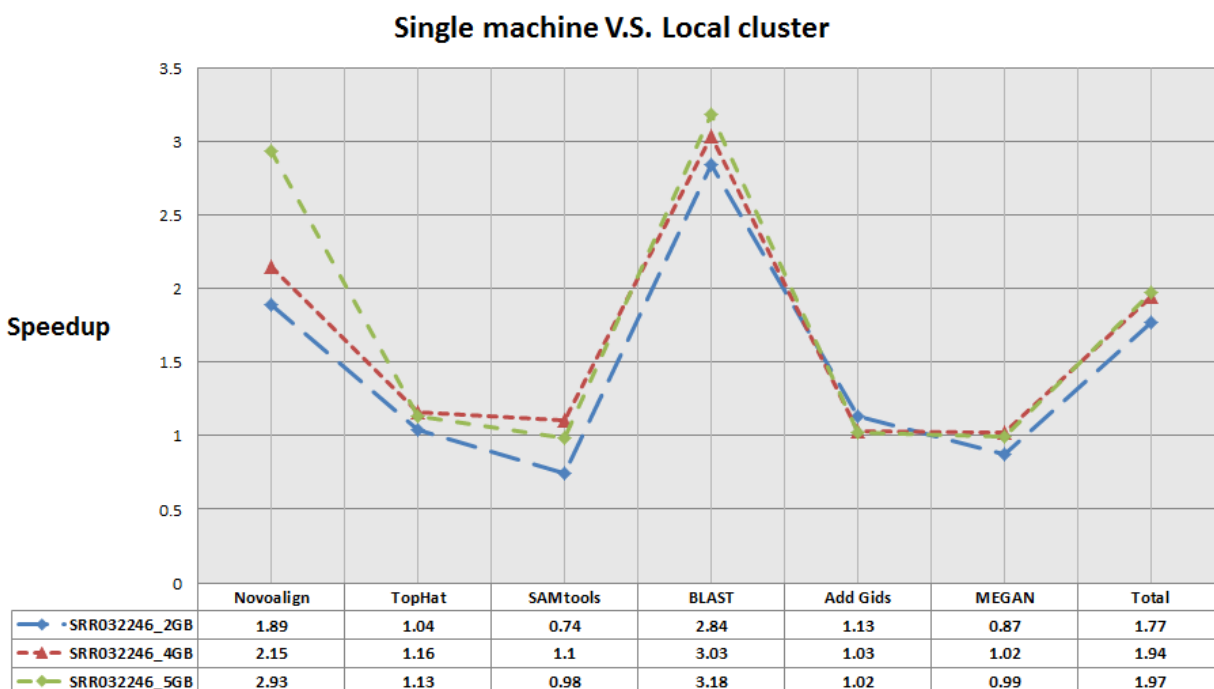


Figure 5.10 Performance of a single machine versus a local cluster with three machines for sample SRR032246. The horizontal axis shows the modules used in RNA CoMPASS. The vertical axis shows the speedup of each module for this sample.

	SRR032238		
	SRR032238(2GB)	SRR032238(4GB)	SRR032238(5GB)
Total Number	9,987,937	19,975,860	24,221,278
Unique Maps	7,973,379 (79.8%)	15,947,676 (79.8%)	19,337,768 (79.8%)

Table 5.2 The sample SRR032238 has been split into 3 pieces. For each piece, we list the total number of reads contained and the number of unique mapped reads aligned by Novoalign.

	SRR032246		
	SRR032246(2GB)	SRR032246(4GB)	SRR032246(5GB)
Total Number	8,869,340	17,738,741	22,161,215
Unique Maps	6,970,645 (78.6%)	13,942,557 (78.6%)	17,413,299 (78.6%)

Table 5.3 The sample SRR032246 has been split into 3 pieces. For each piece, we list the total number of reads contained and the number of unique mapped reads aligned by Novoalign.

## 5.4 The performance comparison between time cost and speedup on grid system

For the benchmarked data set SRR006514 from the *caenorhabditis elegans* experiment, we also performed a detailed comparison of the run time using a single machine versus a grid system as the file size increases. Again we split the dataset into three files with different sizes (Table 5.4). Then we ran these files both using a single machine of the grid and also using the grid system with 6 cores allocated. An additional test was also performed where we ran the file (SRR006514-1600MB) on a grid system with 24 cores allocated. Figure 5.11 shows the speedup of Novoalign on the grid system with 6 cores is roughly 2, and the speedup of BLAST on the grid system with 6 cores is nearly 3. When we run the file (SRR006514-1600MB) on the grid system with 24 cores allocated, the speedup of Novoalign is roughly 4, and the speedup of BLAST is roughly 3. Performance of the parallelized modules has attained a speedup of roughly

$\sqrt{N}$  times (where N is the number of used cores). The performance of other non-parallelized modules has no significant changes.

	SRR006514		
	SRR006514(400MB)	SRR006514(800MB)	SRR006514(1600MB)
Total Number	2,741,725	5,483,380	10,966,805
Unique Maps	971,410 (35.4%)	1,989,411 (36.2%)	4,142,311 (37.8%)

Table 5.4 The sample SRR006514 has been split into 3 pieces. For each piece, we list the total number of reads contained and the number of unique mapped reads aligned by Novoalign.

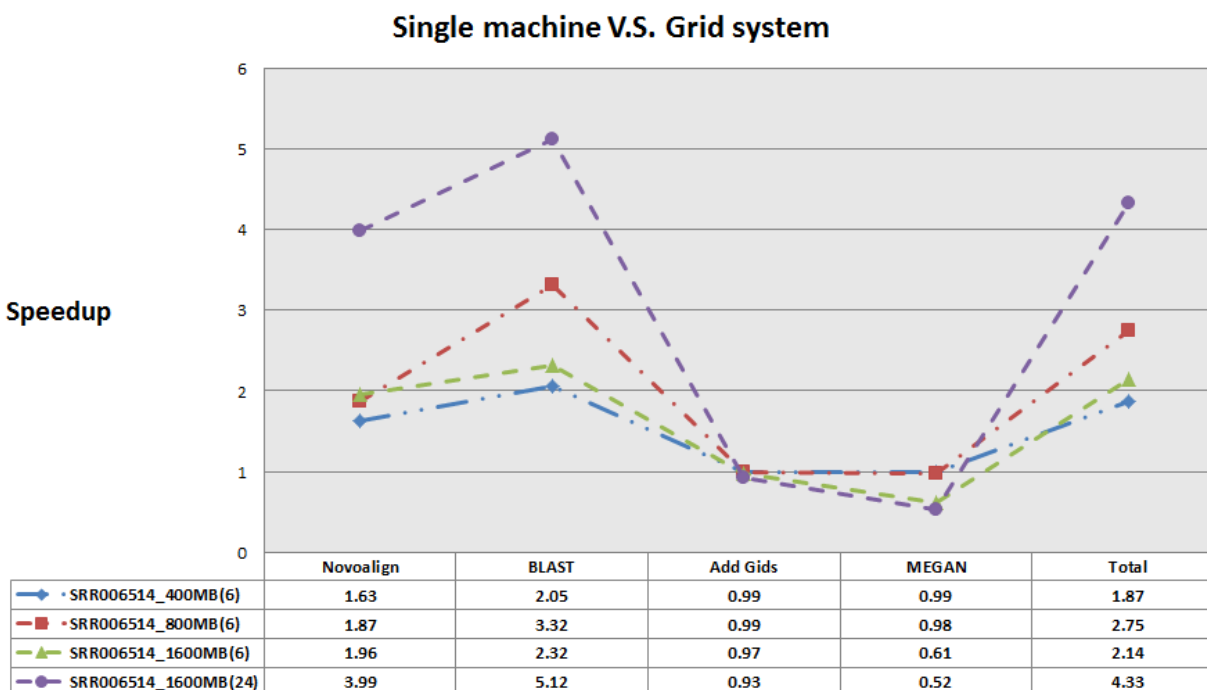


Figure 5.11 Performance of a single machine versus a grid system with 6 and 24 cores allocated for sample SRR006514. The horizontal axis shows the modules used in RNA CoMPASS. The vertical shows the speedup of each module for this sample.

## 5.5 Microarray platform versus next generation sequence platform

Having demonstrated a preferential distribution of genes with MIR155 seeds in down-regulated fractions, we next sought to assess the performance of next generation sequencing relative to microarray analysis. We performed differential expression analysis on our RNA-seq data set and on data sets from two previously published microarray studies in which MIR155

expression vectors were introduced into either a mouse macrophage cell line [O'Connell et al. 2008] or human 293 cells [Skalsky et al. 2007]. Using only gene identifiers common to all four platforms and using a false discovery rate (FDR) of zero, 2165 genes were determined to be down-regulated by NGS, whereas 38 and 58 down-regulated genes were identified in our analysis of the two published microarray data sets (Figure 5.12). NGS identified 102 down-regulated genes (FDR = 0) with 3' UTR 8-mer seeds, while seven and two down-regulated genes with 8-mer seeds were identified in the two microarray data sets. To more stringently assess the relative robustness of NGS for transcriptome and targetome studies, we generated additional control and MIR155 retrovirally transduced Mutu I cell lines and subjected four control and four MIR155 expressing cell lines to Agilent microarray analysis with dye swaps for each comparison. This resulted in better concordance relative to previous microarray studies, but the number of down-regulated genes and the number of down-regulated genes with 8-mer seed sequences were threefold and 2.6-fold lower than that observed using the NGS data set (Figure 5.6).

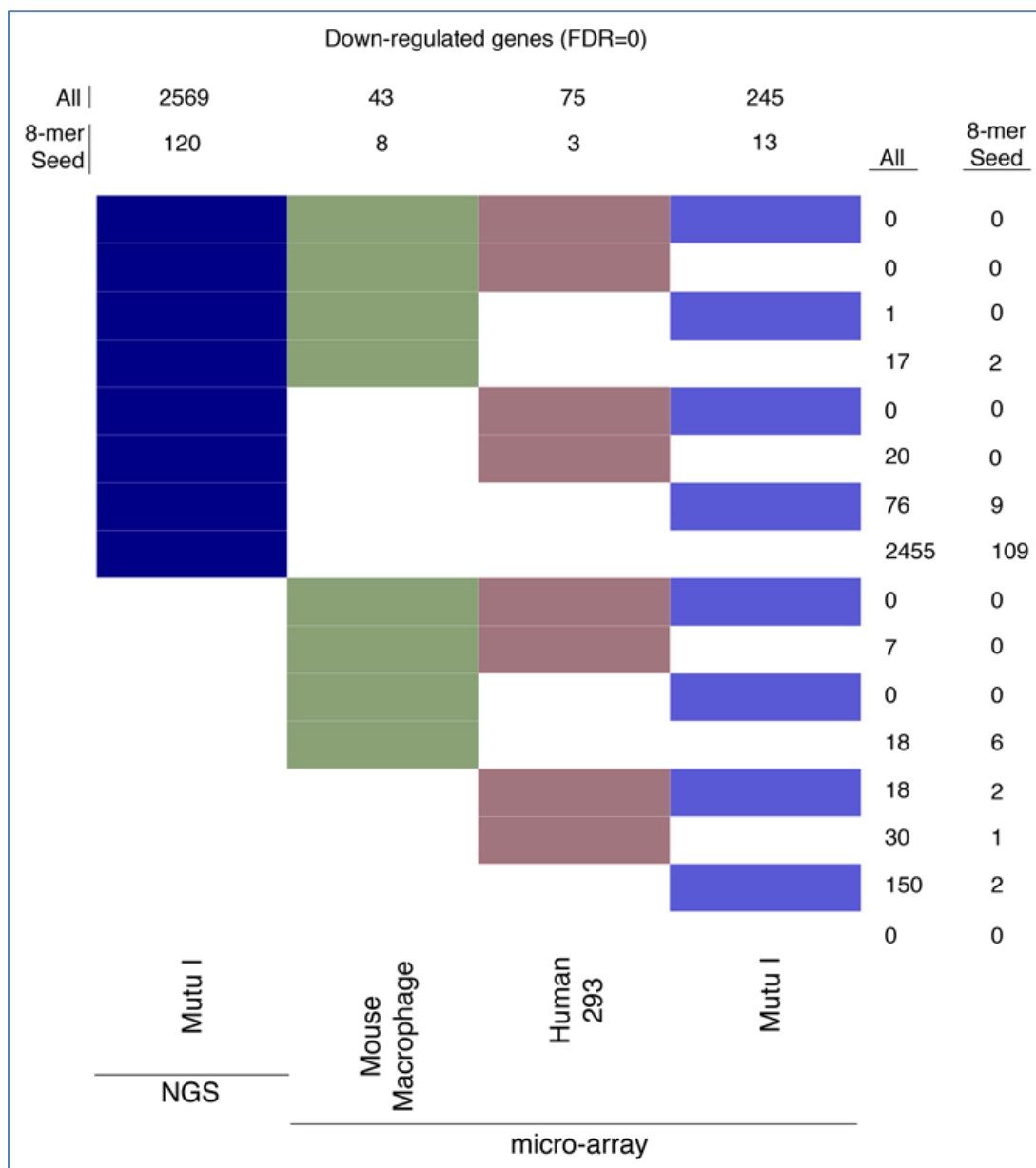


Figure 5.12 Cross-platform comparison of targetome prediction using bitmap. Downregulated genes were identified at a false discovery rate (FDR) = 0 for NGS and each microarray platform. Each gene was determined to be significantly down-regulated (at FDR = 0) or not in each of the four platforms; down-regulated genes were assigned to one of the 24 = 16 possible clusters, represented by color/white patterns and corresponding to 16 rows in the bitmap. Numbers at the top refer to the total number of down-regulated genes for the indicated platform (summation of the number of genes represented by all colored patterns in column). Numbers to the right refer to the number of genes common to platforms with colored patterns in each respective row.

## 5.6 Expression analysis

Microarray technologies can readily be used to generate information regarding the relative expression of genes between samples. Due in part to cross-hybridization and sensitivity issues as well as the analog nature of microarray platforms, however, the determination of absolute transcript levels is challenging. In contrast, NGS provides a digital readout of the number of reads mapping to each gene, and Li et al. (2010a) have shown that when the mean expressed transcript length is 1 kb, 1 RPKM corresponds to roughly one transcript per cell in mouse. It is reasonable to assume that transcript levels falling below this threshold may be of limited functional significance to the overall cell population. At the very fundamental level of RPKM analysis, NGS allows the user to tentatively absolve this group of genes from playing a direct global role in altering cell signaling/phenotype in a particular system. Such genes can be set aside and perhaps considered at a later point in the context of paracrine or subpopulation effects. In control Mutu I cells, approximately half of all annotated genes were found to be expressed below 1 RPKM (Figure 5.13). Approximately 800 genes bearing MIR155 3' UTR seed sequences were found to be expressed below 1 RPKM. Even at a 0.1 RPKM cutoff, more than 600 seed containing genes were found to fall below this threshold and are therefore likely to have limited functional significance in these cells irrespective of whether they are true MIR155 targets. Notably, a higher percentage of MIR155 seed containing genes are expressed in Mutu I B-cells compared to the percentage of all genes expressed, possibly reflecting the critical role of MIR155 in immune cell development [O'Connell et al. 2007; Rodriguez et al. 2007]. Alternatively, this enrichment may simply reflect a general bias toward targeting a group of genes that are universally expressed.

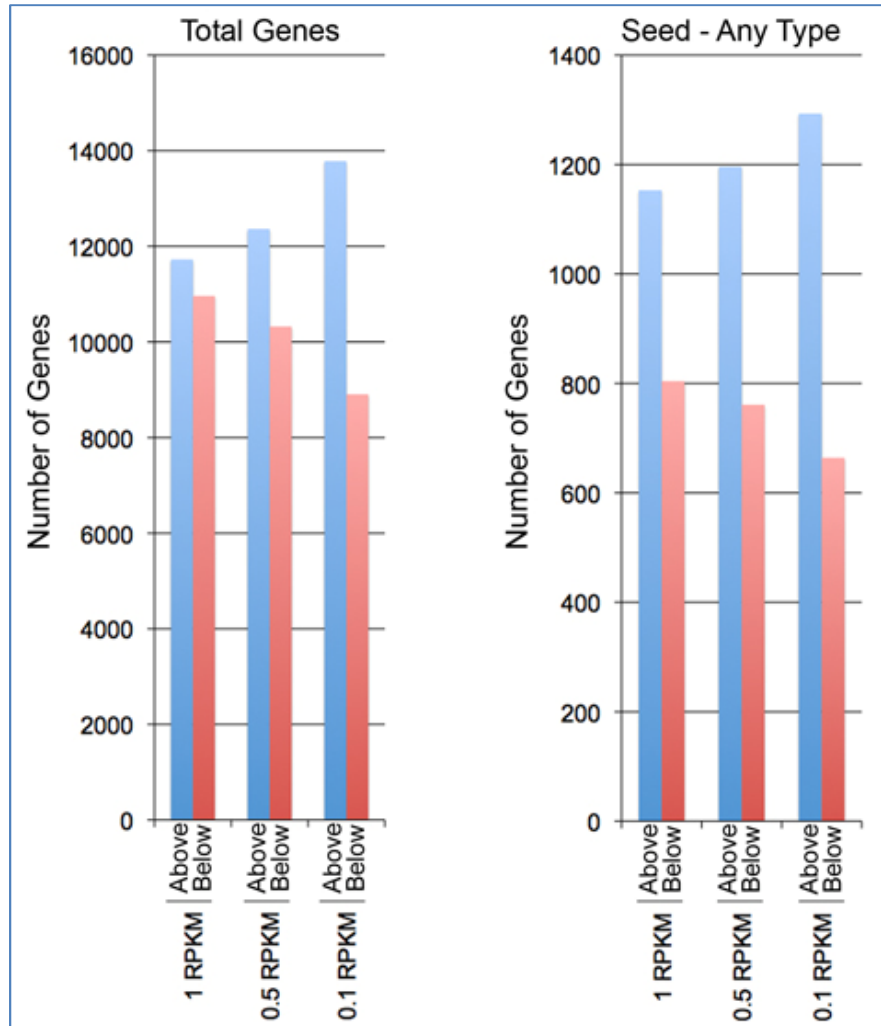


Figure 5.13 The total number of genes and the number of genes containing any MIR155 seed type that are expressed above and below the indicated RPKM cutoffs in control Mutu I cells were counted and graphed.

## 5.7 3' UTR reporter analysis

Luciferase reporter plasmids bearing ectopic 3' UTRs can be used to assess microRNA targeting through the respective 3' UTR. To further analyze the inferred targetome derived from NGS, a 3' UTR reporter data set was generated using 170 3' UTRs containing MIR155 7-mer or 8-mer seeds and nine 3' UTRs with no MIR155 seeds (Figure 5.14). The relative expression of reporters lacking MIR155 seeds in cells cotransfected with a MIR155 expression vector versus a control expression vector fell in the range of 0.8 to 1.04. The relative expression of genes with



MIR155 seed sequences spanned a range from 0.13 to 1.2, allowing us to analyze a spectrum of MIR155 target regulatory classes.

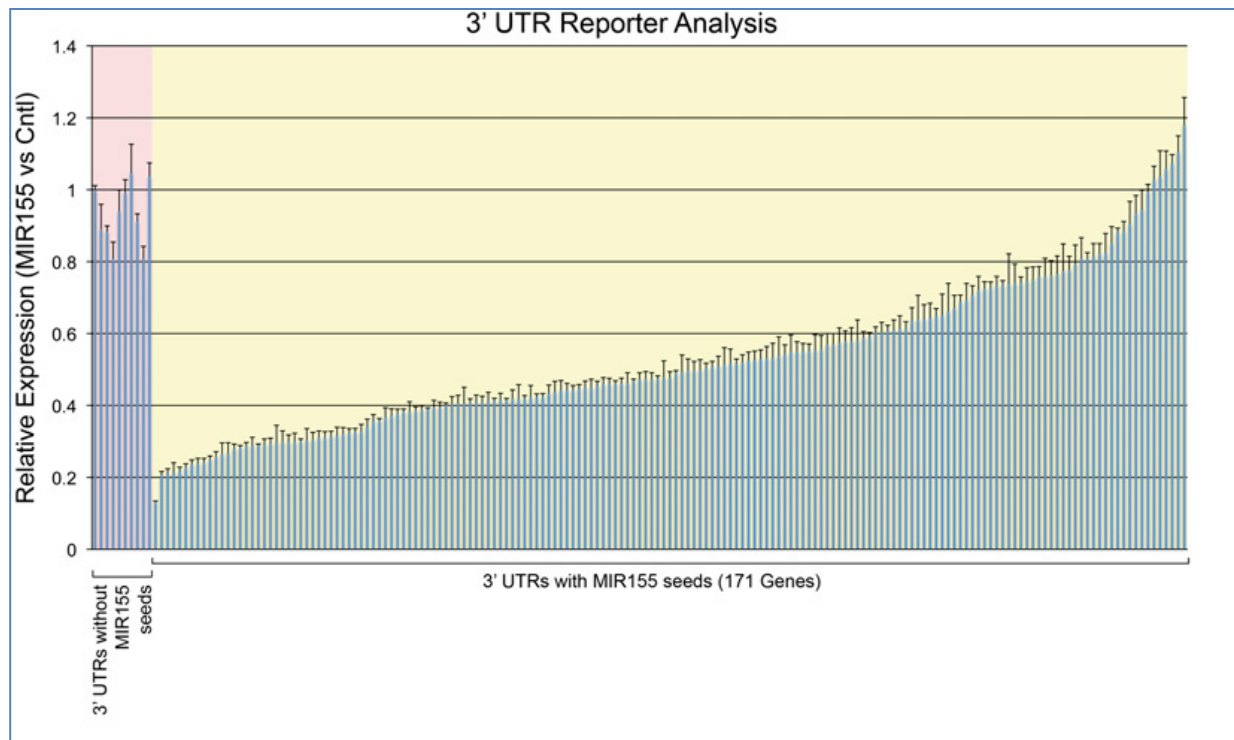


Figure 5.14 Comparison between 3' UTR analysis and RNA-seq analysis. Distribution of 3' UTR suppression by MIR155 in reporter assays.

## 5.8 Splicing evidence in Mutu I and Akata

While RNA-seq can provide digital quantification of gene expression, reads that span exon junctions can provide information about gene isoform usage. We used the junction mapper, Tophat [Trapnell et al. 2009], to identify junction mapped reads throughout the EBV genome for Mutu I and Akata. While no evidence of Cp or Wp derived EBNA1 transcripts was found, evidence for Qp derived EBNA1 splice junctions was observed in both Mutu I and Akata cells (Figure 5.15A). Junction reads were also detected for EBV lytic genes in both Mutu I and Akata cells including junction reads for both BZLF1 (Figure 5.15B) and BSLF2/BMLF1. Further,

evidence for multiple isoform expression (i.e. alternative splicing events) was detected for many genes such as BLLF1/BLLF2 (Figure 5.15C) as well as the complex BamHI A region [Edwards et al. 2008; Reddy et al. 2009]. Within the BamHI A region, for example, there is evidence for alternative splicing at the A73 gene in both Akata and in Mutu I with JUNC00000180 from Mutu I cells providing evidence of exon skipping (skipping of exons 2 and 3). Within the genomic regions spanning the two BART microRNA clusters, there are very few reads, consistent with these microRNAs being produced from excised introns that are presumably unstable and non-polyadenylated (and therefore not enriched during our poly(A)+ selection procedure). In both Mutu I and Akata, there is evidence for two large introns that span the entire region of both of these clusters of microRNAs (JUNC00000094 and JUNC00000178 in Mutu I and JUNC00000053 and JUNC00000084 in Akata). Consistent with this junction evidence, there are pronounced read spikes in Akata cells immediately upstream from the first junction (centered at position 139,270), between these two junctions (centered at position 147,770), and immediately downstream from the second junction (centered at position 151,115) supporting the idea that a stable, poly(A)+ spliced transcript is generated from this transcription unit. The two introns excised from this transcript can conceivably give rise to all BART microRNAs within these two clusters.

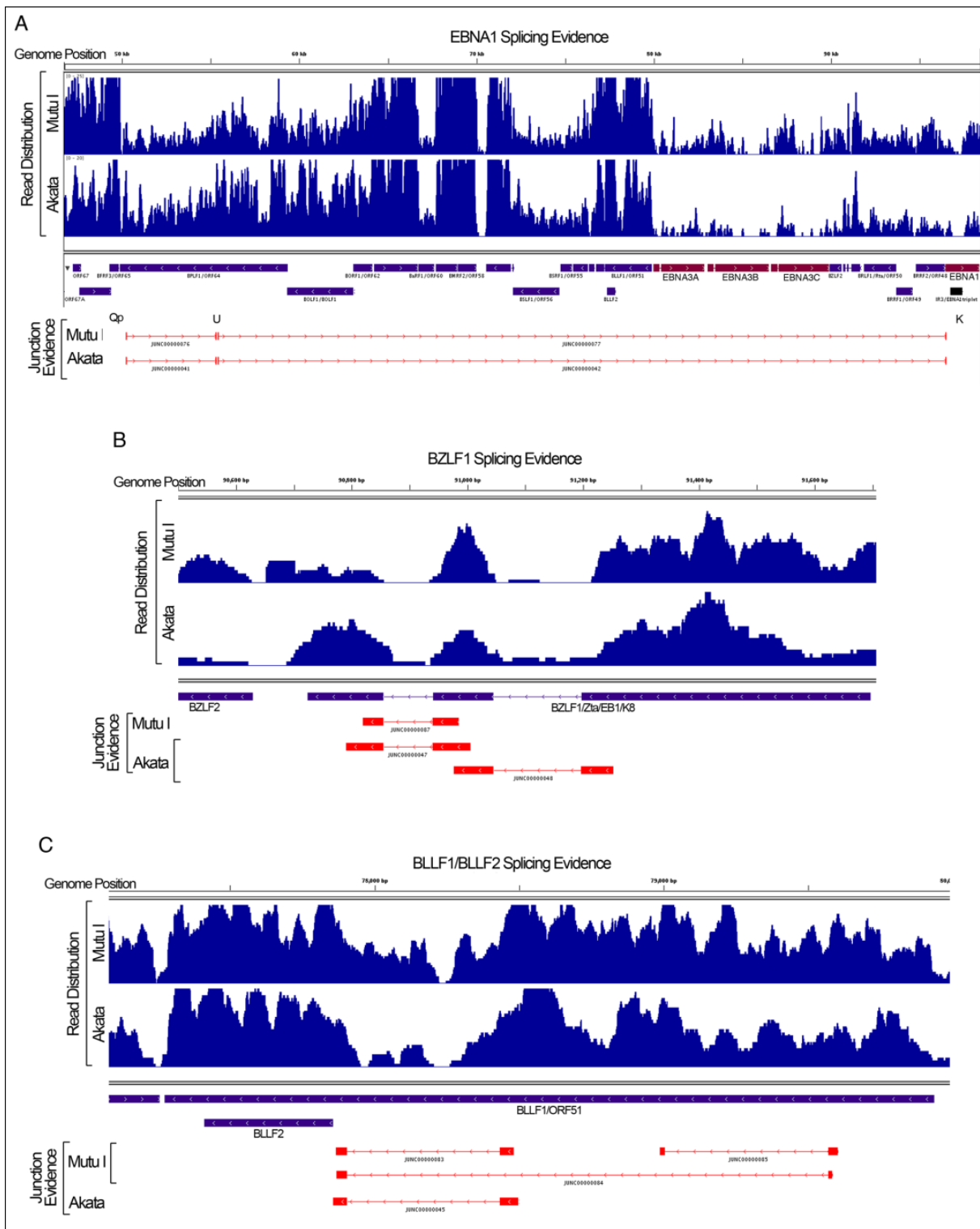


Figure 5.15 Visualization of junction evidence for EBNA1 (A), BZLF1 (B), and BLLF1/BLLF2 (C). Coverage file in wiggle format was generated by SAMMate which has been integrated into RNA CoMPASS.

## 5.9 Performance comparison of SAMMate, TopHat and Novoalign

SAMMate has been integrated into RNA CoMPASS and is also a central component of RNA CoMPASS for human transcriptome analysis. For comparing gene expression scores generated using SAMMate, TopHat and Novoalign to predict miRNA targets, we have studied a pair of control and treatment transcriptomes. The control transcriptome was derived from the wild-type MutuI cell line while the treatment transcriptome was derived from the miRNA-155 retrovirally transduced MutuI cell line [Xu et al., 2010]. MicroRNAs plays pivotal roles in controlling normal and pathology associated cellular processes. Moreover, the importance of miRNA dysregulation in cancer is well known and a number of tumor promoting miRNA's have been identified. As a member of this class of microRNAs, miR-155 is implicated in lymphomagenesis and a wide array of nonlymphoid tumors including breast, colon, and lung. Despite the strong evidence for miRNA-155 as an oncogene, the underlying pathological mechanisms remain unclear, possibly due to limited knowledge of miRNA-155 targets and how these targets are involved in tumorigenesis[Yin et al., 2010; Lin and Xu et al., 2010]. Both transcriptomes were profiled using the Illumina Genome Analyzer II platform with a 50-mer in read length. For each transcriptome, two biological replicates were used. Each biological replicate has 2 to 4 technical replicates nested within it. Each technical replicate of the transcriptome (a single lane in each instrument run) contains around 6 to 12 million short reads. This NGS data set is available at the National Center for Biotechnology Information (NCBI) Short Read Archive (SRA) website <http://www.ncbi.nlm.nih.gov/sra> with access code SRA011001. We aligned the short reads generated from each transcriptome to the reference human genome (Build 37 version 1) using Novoalign, Bowtie and TopHat respectively allowing up to two mismatches. The alignment information of the exons and exon-exon junctions were stored in

SAM/BAM and BED formats. Our biological goal is to predict a list of miRNA-155 direct targets on the genomic scale. Accurate calculation of the gene expression scores is a central problem for achieving this goal since down-regulated genes are likely to be potential miRNA-155 direct targets. We used SAMMate to calculate RPKM gene expression scores for each transcriptome and compared these results with RPKM gene expression score distributions calculated using TopHat and Novoalign alone. We compare the accuracy of the RPKM score reporting capability for the three tools using a selected set of 170 genes for which a 3'-UTR assay was performed for each gene. To make the RPKM scores calculated from the in vivo whole genome sequencing experiments comparable with the fluorescent intensity scores calculated from the in vitro 3'-UTR assay, we used the expression fold change method between treatment (miRNA-155 transduced) and control (wild type) for each gene. Thus, the best tool at reporting gene expression possesses fold changes that are closest to those calculated from 3'-UTR assays. Figure 5.10 shows SAMMate is more accurate than the competitors for gene expression score calculation in 84 out of 170 genes (49%). The numbers for TopHat and Novoalign are 46 (27%) and 40 (24%), respectively (Figure 5.16). Our validation study using 3'-UTR analysis provides compelling evidence that SAMMate exports gene expression RPKM scores more accurately than its competitors.

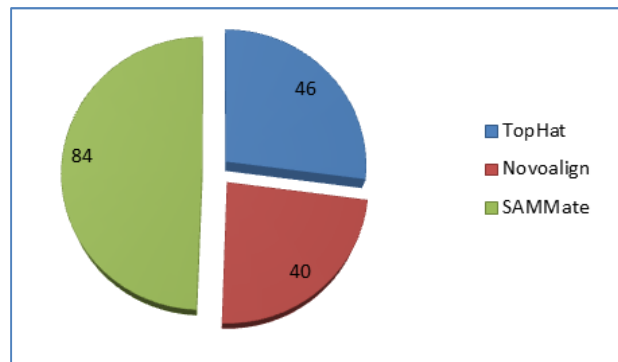


Figure 5.16 Pie chart of percentages of gene fold changes calculated by each tool that is closest to the 3'-UTR experimental results. SAMMate is superior to the other competing tools.

## 5.10 Genome-wide change-point analysis to identify potential miRNA targets

Other than down-regulation at the whole gene locus level, another characteristic response of potential miRNA-155 targets is only visible from the signal map: an abrupt drop-out of the base-wise coverage in the 3' prime end. For this case study, we have the same biological goal as the previous case study, i.e., to predict miRNA-155 targets. We used SAMMate to calculate a signal map for each biological replicate. We then applied a genome-wide change point analysis to all of the annotated 3' UTR regions to identify potential miRNA targets. We also compared the predicted miRNA-155 targets generated using RPKM gene expression scores (previous case study) with the ones generated using signal maps (this case study). We hope these two orthogonal and complementary case studies that share the same biological goal will be more than adequate to demonstrate the robustness and key features of SAMMate. For the sake of completeness, we briefly introduce the change point analysis method that we have applied [Chen et al., 2009]. For each annotated 3'UTR, we test the null hypothesis of the equal mean and variance parameters in the sequence of base-wise signal. The alternative hypothesis is the unknown number of change points' position exists. For statistical analysis we applied a Mean and Variance Change point Model (MVCM) based approaches [Chen et al., 2009]. The input of the change point analysis algorithm is two sets of signal map files generated by SAMMate where each set is a series of replicates of a genome sample. In each signal map file, there are over 249,000 bases. The output for each annotated 3' UTR is a list of change point positions sorted by ascending order according to their Schwarz Information Criterion (SIC) values [Chen et al., 2009]. The 3' UTR coordinates were determined by the human genome annotation file (version hg19) with over 34,000 annotated 3' UTR's in total. We calculated the difference

between the base-wise average for each set of signal map files, representing a single differential signal map for wild-type Mutu I cell line and miRNA-155 transfected Mutu I cell line. It was then followed by a calculation of the SIC values for each annotated 3' UTR. The change point analysis on the differential signal map between wild type and miRNA-155 transduced Mutu I cells was parallelized for multithreading with OpenMP to overcome computational challenges. Figure 5.17 presents a comparison of the ranked gene lists called by Differential Expression Analysis (DEA) and Change-Point Analysis (CPA). The horizontal axis represents the gene number cut-offs starting from 100 to 5,000. Three sets of genes are shown at each cut-off: putative miRNA targets predicted exclusively by CPA, exclusively by DEA, and by both DEA and CPA. Both methods for miRNA target prediction are able to identify a common set of genes, and each method has a unique gene list. Figure 5.11 shows the validation studies of the putative miRNA targets against the set of 170 genes for which 3' UTR assays were performed. Figure 4.9 and Figure 4.10 are consistent with one another. The results support the notion that a combination of DEA and CPA is able to predict a comprehensive and a conserved list of miRNA targets. SAMMate provided the essential input, i.e. gene expression scores and signal maps, for both DEA and CPA as reported in the two case studies. In summary, SAMMate is a highly valuable tool to compare two orthogonal and complementary approaches to detect gene structure alterations.

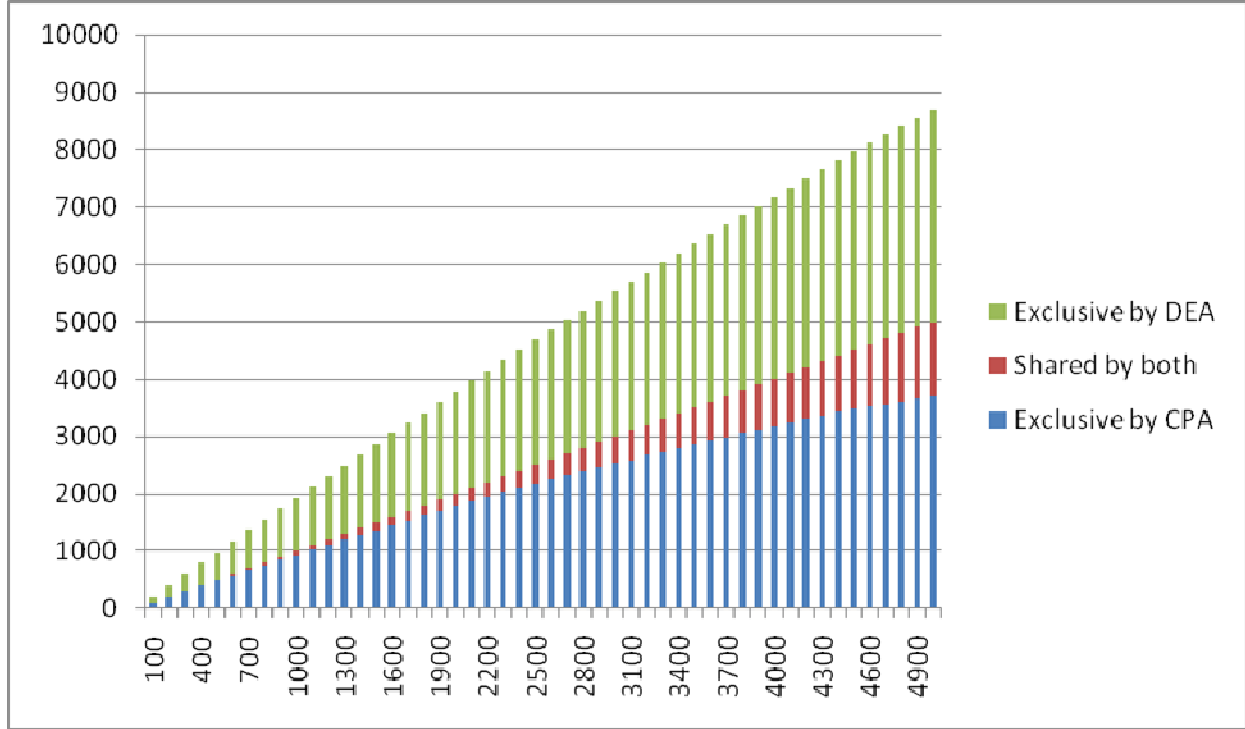


Figure 5.17 Comparison of Differential Expression Analysis (DEA) and Change Point Analysis (CPA) in Prediction of miRNA-155 Targets.

## 5.11 iQuant algorithm to quantify transcriptomes at isoform-level

For testing the performance of iQuant algorithm [Nguyen et al., 2011] to quantify transcriptomes at isoform level, we simulated RNA-seq experiments using FluxSimulator, which is able to simulate whole transcriptome sequencing experiments with the Illumina Genome Analyzer platform. From our simulation experiment, we used the mouse genome as the reference genome. Mouse annotation file (NCBI 37/mm9), 27, 150 protein coding isoforms corresponding to 21, 711 protein coding genes 41M paired-end short reads (75 bases) were simulated. After simulation, 18, 731 isoforms were expressed, corresponding to 15, 419 genes. From Figure 5.18, the calculated proportions of most of isoforms are closed to the true proportions. This simulation study provides a compelling evidence for the excellent accuracy of our algorithm in quantifying isoform abundance.



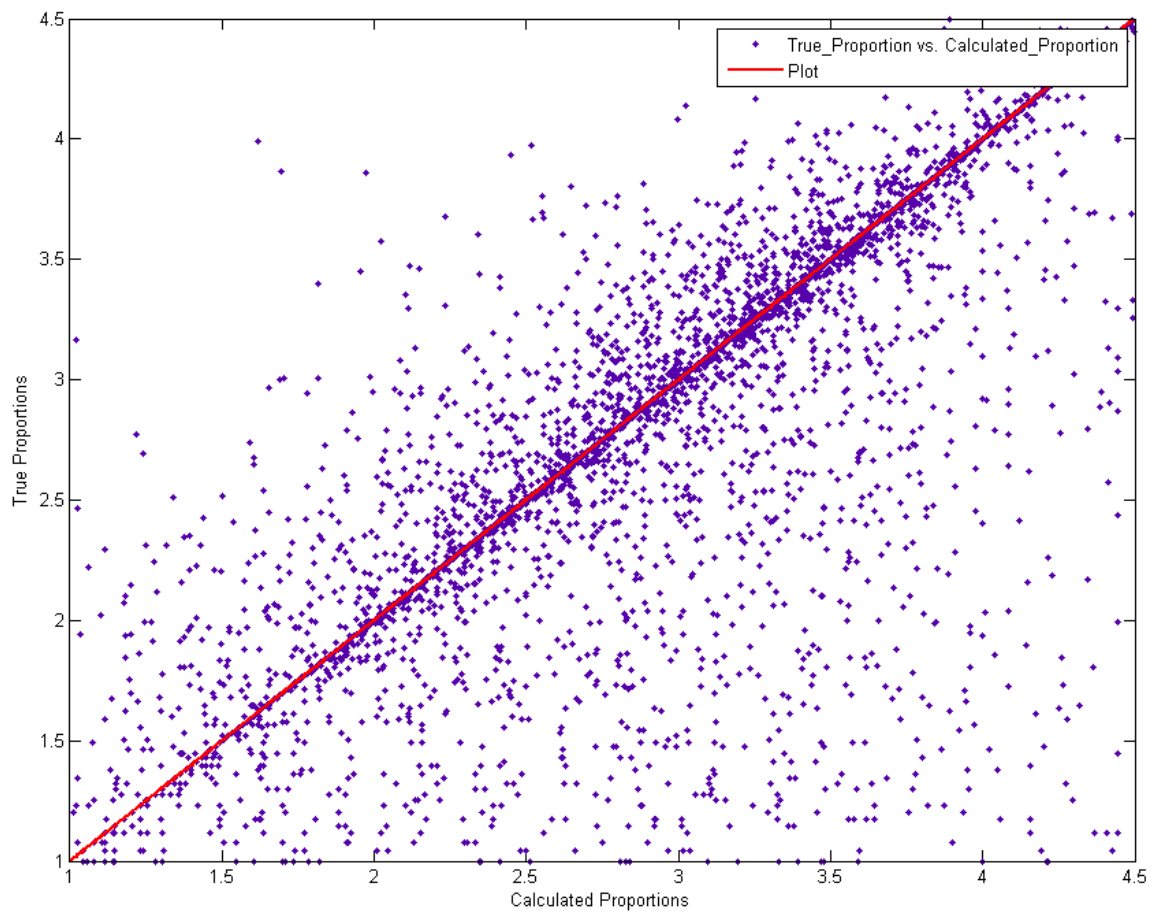


Figure 5.18 Simulation study to evaluate the performance of iQuant algorithm implemented in RNA CoMPASS using FluxSimuloator. We plot predicted isoform abundance scores against true abundance scores.

## Chapter 6 Conclusion

### 6.1 Conclusion

RNA CoMPASS is a comprehensive multi-processor analysis system for RNA sequencing. It provides a convenient graphical user interface and automates analysis and visualization of reads against the host transcriptome and investigation of exogenous sequence reads. This latter task has proven to be extremely computationally intensive, but can be instrumental in the identification of bacterial or viral sequences found in RNA sequencing experiments. For example, Lin *et al.*, (2012) utilized this approach to find Epstein-Barr virus and murine leukemia virus sequences in some commonly used cell lines. RNA CoMPASS manages the computational burden associated with such extensive analysis on large sequencing studies automatically through distribution of tasks over a computing cluster.

This computational pipeline provides biological researchers with convenient approaches to investigate the presence of exogenous sequences in RNA-seq data. Since RNA CoMPASS has been integrated many popular bioinformatics tools, it provides the typical endogenous RNA-Sequencing analysis along with the investigation of exogenous sequences. Besides, we parallelize the modules which are extremely computational intensive, it is able to greatly accelerate the progress of research for researchers.

This computational pipeline can be also used to analyze transcriptome changes induced by the human microRNA MIR155 using RNA-seq. A comparison with 3' UTR reporter assay demonstrated general concordance between NGS and corresponding 3' UTR reporter results. Nonharmonious results were investigated more deeply using transcript structure information assembled from the NGS data. This analysis revealed that transcript structure plays a substantial role in mitigated targeting and in frank targeting failures.

In analysis of EBV transcriptome, our results also showed robust detection of EBV derived transcripts by RNA-seq using the pipeline outlined here. From a quantitative standpoint, several studies have shown this approach to outperform microarrays since it is more accurate [Marioni et al. 2008; Mortazavi et al. 2008; Xu et al. 2010] and since there is an inherently broad dynamic range. The digital nature of RNA-seq allows users to better compare the relative expression of distinct genes through the calculation of RPKMs. This should result in an improvement over microarrays in the analysis of virus-associated transcriptomes not only for EBV but for other viruses. With its high level of accuracy, its broad dynamic range, its utility in assessing transcript structure, and its capacity to accurately interrogate global direct and indirect transcriptome changes, NGS is a useful tool for investigating the biology and mechanisms of action of microRNAs.

For efficiently processing NGS data, our GUI software RNA CoMPASS allows biomedical researchers to quickly process raw sequence data in FASTQ format and is compatible with multiple RNA-seq file formats. RNA CoMPASS also automates some standard procedures in DNA-seq and RNA-seq data analysis. Using either standard or customized annotation files, RNA CoMPASS allows users to accurately calculate the short read coverage of genomic intervals. In particular, for RNA-seq data RNA CoMPASS can accurately calculate the gene expression abundance scores for customized genomic intervals using short reads originating from both exons and exon-exon junctions. Furthermore, RNA CoMPASS can calculate a whole-genome signal map at base-wise resolution in a short time allowing researchers to solve an array of bioinformatics problems. Finally, RNA CoMPASS can export both wiggle files for alignment visualization in the UCSC genome browser and an alignment

statistics report. The biological impact of these features has been already demonstrated via several case studies that predict miRNA targets using short read alignment information files.

With just a few mouse clicks, RNA CoMPASS will provide biomedical researchers all-in-one functionality including human transcriptome quantification and the typical endogenous RNA-Sequencing analysis along with the investigation of exogenous sequences. RNA CoMPASS is deployable on either a local cluster or a grid environment managed by PBS submission. Our software is constantly updated and will greatly facilitate the downstream analysis of NGS data. Both the source code and the executable files are freely available under the GNU General Public License at <http://rnacompass.sourceforge.net>.

## Appendix A

### Tables

The following tables are raw data generated by RNA CoMPASS for performance comparisons between single machine and local cluster (or grid system). These tables record the start time, end time and duration of each module during the whole test processing.

	SRR032238_2GB.fastq					
	Single Machine			Local Cluster		
	Start	End	Duration	Start	End	Duration
<b>Novoalign</b>	9:59:46	11:16:06	<b>1:16:20</b>	12:11:06	12:49:46	<b>0:38:40</b>
<b>TopHat</b>	11:17:29	12:58:10	<b>1:40:41</b>	12:51:12	14:26:58	<b>1:35:46</b>
<b>SAMtools</b>	12:58:10	12:58:41	<b>0:00:31</b>	14:26:58	14:27:35	<b>0:00:37</b>
<b>BLAST</b>	12:59:43	15:22:14	<b>2:22:31</b>	14:28:38	15:17:50	<b>0:49:12</b>
<b>Add Gids</b>	15:22:15	15:27:57	<b>0:05:42</b>	15:17:50	15:23:25	<b>0:05:35</b>
<b>MEGAN</b>	15:27:58	15:31:20	<b>0:03:22</b>	15:23:25	15:26:48	<b>0:03:23</b>

	SRR032238_4GB.fastq					
	Single Machine			Local Cluster		
	Start	End	Duration	Start	End	Duration
<b>Novoalign</b>	15:31:20	18:08:52	<b>2:37:32</b>	15:26:49	16:36:58	<b>1:10:09</b>
<b>TopHat</b>	18:12:08	21:29:45	<b>3:17:37</b>	16:41:46	19:52:12	<b>3:10:26</b>
<b>SAMtools</b>	21:29:45	21:30:52	<b>0:01:07</b>	19:52:12	19:53:42	<b>0:01:30</b>
<b>BLAST</b>	21:32:58	2:18:12	<b>4:45:14</b>	19:55:57	21:24:33	<b>1:28:36</b>
<b>Add Gids</b>	2:18:13	2:24:22	<b>0:06:09</b>	21:24:34	21:30:28	<b>0:05:54</b>
<b>MEGAN</b>	2:24:22	2:30:01	<b>0:05:39</b>	21:30:29	21:37:08	<b>0:06:39</b>

	SRR032238_5GB.fastq					
	Single Machine			Local Cluster		
	Start	End	Duration	Start	End	Duration
<b>Novoalign</b>	2:30:02	5:38:38	<b>3:08:36</b>	21:37:09	23:01:49	<b>1:24:40</b>
<b>TopHat</b>	5:42:24	9:58:00	<b>4:15:36</b>	23:06:13	3:02:15	<b>3:56:02</b>
<b>SAMtools</b>	9:58:00	9:59:22	<b>0:01:22</b>	3:02:15	3:04:13	<b>0:01:58</b>
<b>BLAST</b>	10:01:54	15:29:25	<b>5:27:31</b>	3:06:57	4:51:51	<b>1:44:54</b>
<b>Add Gids</b>	15:29:27	15:35:55	<b>0:06:28</b>	4:51:51	4:58:07	<b>0:06:16</b>
<b>MEGAN</b>	15:35:55	15:42:41	<b>0:06:46</b>	4:58:07	5:05:59	<b>0:07:52</b>

	SRR032246_2GB.fastq					
	Single Machine			Local Cluster		
	Start	End	Duration	Start	End	Duration
<b>Novoalign</b>	15:42:43	16:42:31	<b>0:59:48</b>	5:06:00	5:37:43	<b>0:31:43</b>
<b>TopHat</b>	16:43:50	17:53:58	<b>1:10:08</b>	5:39:01	6:46:10	<b>1:07:09</b>
<b>SAMtools</b>	17:53:58	17:54:20	<b>0:00:22</b>	6:46:10	6:46:40	<b>0:00:30</b>
<b>BLAST</b>	17:55:18	20:12:07	<b>2:16:49</b>	6:47:40	7:35:51	<b>0:48:11</b>
<b>Add Gids</b>	20:12:07	20:18:07	<b>0:06:00</b>	7:35:52	7:41:12	<b>0:05:20</b>
<b>MEGAN</b>	20:18:07	20:20:51	<b>0:02:44</b>	7:41:12	7:44:20	<b>0:03:08</b>

	SRR032246_4GB.fastq					
	Single Machine			Local Cluster		
	Start	End	Duration	Start	End	Duration
<b>Novoalign</b>	20:20:52	22:23:32	<b>2:02:40</b>	7:44:20	8:41:26	<b>0:57:06</b>
<b>TopHat</b>	22:26:23	0:58:38	<b>2:32:15</b>	8:44:44	10:55:49	<b>2:11:05</b>
<b>SAMtools</b>	0:58:38	0:59:46	<b>0:01:08</b>	10:55:49	10:56:51	<b>0:01:02</b>
<b>BLAST</b>	1:01:44	5:43:41	<b>4:41:57</b>	10:58:44	12:31:47	<b>1:33:03</b>
<b>Add Gids</b>	5:43:41	5:50:08	<b>0:06:27</b>	12:31:47	12:38:02	<b>0:06:15</b>
<b>MEGAN</b>	5:50:08	5:55:35	<b>0:05:27</b>	12:38:02	12:43:23	<b>0:05:21</b>

	SRR032246_5GB.fastq					
	Single Machine			Local Cluster		
	Start	End	Duration	Start	End	Duration
<b>Novoalign</b>	5:55:36	8:28:33	<b>2:32:57</b>	12:43:24	13:49:16	<b>1:05:52</b>
<b>TopHat</b>	8:31:58	11:41:56	<b>3:09:58</b>	13:53:40	16:41:06	<b>2:47:26</b>
<b>SAMtools</b>	11:41:56	11:43:22	<b>0:01:26</b>	16:41:06	16:42:47	<b>0:01:41</b>
<b>BLAST</b>	11:45:47	17:27:28	<b>5:41:41</b>	16:45:16	18:32:37	<b>1:47:21</b>
<b>Add Gids</b>	17:27:30	17:34:13	<b>0:06:43</b>	18:32:38	18:39:13	<b>0:06:35</b>
<b>MEGAN</b>	17:34:13	17:40:45	<b>0:06:32</b>	18:39:13	18:45:52	<b>0:06:39</b>

	SRR006514_400MB.fastq					
	Single Machine			Local Cluster - 6		
	Start	End	Duration	Start	End	Duration
<b>Novoalign</b>	9:20:39	10:15:53	<b>0:55:14</b>	9:19:36	9:53:30	<b>0:33:54</b>
<b>BLAST</b>	10:16:14	12:47:10	<b>2:30:56</b>	9:53:51	11:07:24	<b>1:13:33</b>
<b>Add Gids</b>	12:47:10	12:51:15	<b>0:04:05</b>	11:07:24	11:11:30	<b>0:04:06</b>
<b>MEGAN</b>	12:51:15	12:52:55	<b>0:01:40</b>	11:11:30	11:13:11	<b>0:01:41</b>

	SRR006514_800MB.fastq					
	Single Machine			Local Cluster - 6		
	Start	End	Duration	Start	End	Duration
<b>Novoalign</b>	15:44:45	17:10:18	<b>1:25:33</b>	16:57:21	17:43:08	<b>0:45:47</b>
<b>BLAST</b>	17:10:59	22:22:06	<b>5:11:07</b>	17:43:51	19:17:34	<b>1:33:43</b>
<b>Add Gids</b>	22:22:06	22:26:26	<b>0:04:20</b>	19:17:34	19:21:55	<b>0:04:21</b>
<b>MEGAN</b>	22:26:26	22:29:29	<b>0:03:03</b>	19:21:55	19:25:01	<b>0:03:06</b>

	SRR006514_1600MB.fastq								
	Single Machine			Local Cluster - 6			Local Cluster - 24		
	Start	End	Duration	Start	End	Duration	Start	End	Duration
<b>Novoalign</b>	9:23:37	11:41:02	<b>2:17:25</b>	16:52:51	18:03:08	<b>1:10:17</b>	11:24:40	11:59:06	<b>0:34:26</b>
<b>BLAST</b>	11:42:29	17:13:42	<b>5:31:13</b>	18:04:37	20:27:06	<b>2:22:29</b>	12:00:50	13:05:33	<b>1:04:43</b>
<b>Add Gids</b>	17:13:42	17:18:03	<b>0:04:21</b>	20:27:06	20:31:35	<b>0:04:29</b>	13:05:33	13:10:13	<b>0:04:40</b>
<b>MEGAN</b>	17:18:03	17:21:15	<b>0:03:12</b>	20:31:35	20:36:52	<b>0:05:17</b>	13:10:13	13:16:19	<b>0:06:06</b>

## Glossary

Definitions are collected here for easy reference. In general, the accepted definitions for terms are used, although some terms are used in a more restricted sense than their usual interpretation.

**3'-UTR region** - sequences on the 3' end of mRNA but not translated into protein. 3' UTR may contain sequences that regulate translation efficiency, mRNA stability, and polyadenylation signals.

**7-mer seed** - mRNA bind to the 2-8 nucleotides of the 5'UTR of miRNA seed.

**8-mer seed** - mRNA bind to the 2-8 nucleotides of the 5'UTR of miRNA seed.

**ABI Solid** - ABI Solid Sequencing to profile stem cell transcriptomes.

**ABYSS** - assembly By Short Sequences - a de novo, parallel, paired-end sequence assembler.

**Alternative splicing** - is a process by which the exons of the RNA produced by transcription of a gene (a primary gene transcript or pre-mRNA) are reconnected in multiple ways during RNA splicing.

**Assembly** - a process to reconstruct a long DNA sequence from numerous fragments.

**BAM format** - a BAM file (.bam) is the binary version of a SAM file.

**BED format** - is a tab-delimited text file that defines a feature track.

**BLAST** - is an algorithm for comparing primary biological sequence information, such as the amino-acid sequences of different proteins or the nucleotides of DNA sequences.



**Bowtie** - is an ultrafast, memory-efficient short read aligner. It aligns short DNA sequences (reads) to the human genome at a rate of over 25 million 35-bp reads per hour.

**Bridge amplification** - is a Illumina's bridge amplification sequencing technology.

**Browser and Server (B/S) structure** - is the rise of a WEB Network Models, WEB Browser is the client the most important applications. This model unifies the client, the system functions to achieve the core focus to the server, simplifying the system development, maintenance and use.

**Burkitts lymphoma** - is a cancer of the lymphatic system (in particular, B lymphocytes).

**Burrows-Wheeler transformation (BWT)** - is an algorithm used in data compression techniques such as bzip2.

**cDNA** - complementary DNA; a form of DNA artificially synthesized from a messenger RNA template and used in genetic engineering to produce gene clones.

**Change Point Analysis** - it is an analytical method that attempts to find a point along a distribution of values where the characteristics of the values before and after the point are different.

**Chromosome** - is an organized structure of DNA and protein found in cells. It is a single piece of coiled DNA containing many genes, regulatory elements and other nucleotide sequences.

**Client and Server architecture** - is a computing model that acts as distributed application which partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

**Constrained convex quadratic** - is an optimization problem in which both the objective function and the constraints are quadratic functions.

**Contamination** - is the presence of a minor and unwanted constituent (contaminant) in material, physical body, natural environment, at a workplace, etc.

**Contig** - a series of overlapping clones or a sequence defining an uninterrupted section of a chromosome.

**De novo** - is a Latin expression meaning "from the beginning," "afresh," "anew," "beginning again."

**Differential Expression Analysis** - differential expression using the simplified exon union or exon intersection methods reports no changes between conditions while estimating read counts and expression for the individual isoforms detects both differential expression at the gene and isoform level.

**Dynamic programming** - is a method for solving complex problems by breaking them down into simpler subproblems. It is applicable to problems exhibiting the properties of overlapping subproblems which are only slightly smaller and optimal substructure.

**edgeR** - a Bioconductor package for differential expression analysis of digital gene expression data.

**EM algorithm** - in statistics, an expectation maximization (EM) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables.

**Embryonic** - is a multicellular diploid eukaryote in its earliest stage of development, from the time of first cell division until birth, hatching, or germination.

**Endogenous** - originating or produced within an organism, tissue, or cell

**Enzyme** - any of numerous proteins or conjugated proteins produced by living organisms and functioning as biochemical catalysts.

**Epstein-Barr virus (EBV)** - also called human herpesvirus 4 (HHV-4), is a virus of the herpes family and is one of the most common viruses in humans.

**ERANGE** - is a Python package for doing RNA-seq and ChIP-seq (hence the "dual-use"), and is a descendant of the ChIPSeq mini peak finder (Johnson, 2007).

**Exogenous** - Biology Derived or developed from outside the body; originating externally.

**Exon** - a sequence of DNA that codes information for protein synthesis that is transcribed to messenger RNA.

**Exon-exon junctions** - a sequence fragment cross two exons. One end is mapped on first exon and the other is mapped on the second exon.

**False Discovery Rate (FDR)** - a statistical method used in multiple hypothesis testing to correct for multiple comparisons.

**FASTA** - is a DNA and protein sequence alignment software package first described (as FASTP).

**FASTQ** - is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores.

**Fluorescence** - is the emission of light by a substance that has absorbed light or other electromagnetic radiation.

**FPKM** - Fragments Per Kilobase of exon model per Million mapped Fragments.

**Fragment** - refers to long DNA strands break or separate (something) into fragments.

**Gene annotations** - is the process of identifying the locations of genes and all of the coding regions in a genome and determining what those genes do.

**Gene expression** - is the process by which information from a gene is used in the synthesis of a functional gene product.

**Grid network** - is a kind of computer network consisting of a number of (computer) systems connected in a grid topology.

**High-throughput sequencing (HTS)** - is a technology that can parallelize the sequencing process, producing thousands or millions of sequences at once.

**Hit table format** - a file format that is used for BLAST stores output results.

**Hybridization probe** - is a fragment of DNA or RNA of variable length (usually 100-1000 bases long), which is used in DNA or RNA samples to detect the presence of nucleotide sequences (the DNA target) that are complementary to the sequence in the probe.

**Hybridize** - to form base pairs between complementary regions of (two strands of DNA that were not originally paired).

**Illumina Genome Analyzer platform** - is a platform of Illumina (company), the recent applications include sequencing mammalian transcriptomes.

**Initially Unmapped Reads (IUM's)** - the short sequences which are not able to aligned to reference genome using aligners.

**Intergenic** - is a stretch of DNA sequences located between clusters of genes that contain few or no genes.

**Intron** - a segment of a gene situated between exons that is removed before translation of messenger RNA and does not function in coding for protein synthesis.

**Isoform** - is any of several different forms of the same protein. Different forms of a protein may be produced from related genes, or may arise from the same gene by alternative splicing.

**Java Parallel Processing Framework (JPPF)** - JPPF enables applications with large processing power requirements to be run on any number of computers, in order to dramatically reduce their processing time.

**Life Science's 454** - Life Science's 454 Sequencing to discover SNPs in maize.

**Lowest common ancestor** - is a concept in graph theory and computer science. Let  $T$  be a rooted tree with  $n$  nodes. The lowest common ancestor is defined between two nodes  $v$  and  $w$  as the lowest node in  $T$  that has both  $v$  and  $w$  as descendants (where we allow a node to be a descendant of itself).

**Mammalian** - any of various warm-blooded vertebrate animals of the class Mammalia, including humans, characterized by a covering of hair on the skin and, in the female, milk-producing mammary glands for nourishing the young.

**MAQ** - a Bioinformatics Tool for Automatic Macroarray Analysis.

**MEGAN** - (MEtaGenome ANalyzer) is a computer program that allows optimized analysis of large metagenomic datasets.

**Metagenomic** - is the study of metagenomes, genetic material recovered directly from environmental samples.

**Microarray** - a multiplex lab-on-a-chip. It is a 2D array on a solid substrate (usually a glass slide or silicon thin-film cell) that assays large amounts of biological material using high-throughput screening methods.

**miRNA-155** - a short RNA molecule that plays a crucial role in various physiological and pathological processes.

**Model-View-Controller (MVC)** - is a design pattern for computer user interfaces that divides an application into three areas of responsibility.

**Molecular** - an electrically neutral group of two or more atoms held together by covalent chemical bonds. Molecules are distinguished from ions by their electrical charge.

**mRNA** - is a molecule of RNA that encodes a chemical "blueprint" for a protein product. mRNA is transcribed from a DNA template, and carries coding information to the sites of protein synthesis, the ribosomes.

**Mutation** - in molecular biology and genetics, mutations are changes in a genomic sequence: the DNA sequence of a cell's genome or the DNA or RNA sequence of a virus.

**Mutu I cell line** - is derived from Burkitt's lymphoma (BL) and retain the in vivo phenotype of Epstein Barr virus (EBV) expression that is characterized by expression of EBV-determined nuclear antigen 1 (EBNA1), EBV-encoded RNAs (EBERs) and transcripts from the BamHI A region (BARF0).

**Needleman-Wunsch algorithm** - the Needleman Wunsch algorithm performs a global alignment on two sequences (called A and B here). It is commonly used in bioinformatics to align protein or nucleotide sequences.

**Next-Generation Sequencing** - refers to a group of new DNA sequencing technologies that can rapidly sequence DNA on the gigabase scale.

**non-coding RNA** - is a functional RNA molecule that is not translated into a protein. Less-frequently used synonyms are non-protein-coding RNA (npcRNA), non-messenger RNA (nmRNA), small non-messenger RNA (snmRNA) and functional RNA (fRNA).

**Novoalign** - gapped alignment of single end and paired end Illumina GA I & II reads and reads from the new Helicos Heliscope Genome Analyzer. High sensitivity and specificity, using base qualities at all steps in the alignment. Includes adapter trimming, base quality calibration, Bi-Seq alignment, and option to report multiple alignments per read.

**Nucleic acid** - any of a group of complex compounds found in all living cells and viruses, composed of purines, pyrimidines, carbohydrates, and phosphoric acid. Nucleic acids in the form of DNA and RNA control cellular function and heredity.

**Nucleotide** - any of various compounds consisting of a nucleoside combined with a phosphate group and forming the basic constituent of DNA and RNA.

**One-color design** - refers one sample is used in a microarray technology experiment.

**OpenMP** - is an application programming interface (API) that supports multi-platform shared memory multiprocessing programming in C, C++, and FORTRAN.

**Organism** - in biology, an organism is any contiguous living system (such as animal, plant, fungus, or micro-organism).

**Parallelization** - is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel").

**Phylogeny** - the evolutionary development and history of a species or higher taxonomic grouping of organisms.

**Pipeline** - a chain of data-processing stages.

**Portable Batch System (PBS)** - is the name of computer software that performs job scheduling. Its primary task is to allocate computational tasks, i.e., batch jobs, among the available computing resources.

**Probe-target hybridization** - is usually detected by optically labeled targets, which determines the relative abundance of each target in the sample.

**Protein** - is organic compounds made of amino acids arranged in a linear chain and folded into a globular or fibrous form.



**Reference genome** - is a digital nucleic acid sequence database, assembled by scientists as a representative example of a species' set of genes.

**Regulation of gene expression (or gene regulation)** - the processes that cells and viruses use to regulate the way that the information in genes is turned into gene products.

**Relationship matrix** - a matrix that constructs from gene annotations to reflect the relationship between isoforms and exons.

**Repeat mapped reads** - reads which are mapped on multiple locations against reference genome using aligners.

**Ribonucleic acid (RNA)** - is part of a group of molecules known as the nucleic acids, which are one of the four major macromolecules (along with lipids, carbohydrates and proteins) essential for all known forms of life. Like DNA, RNA is made up of a long chain of components called nucleotides.

**RNA-seq** - also called "Whole Transcriptome Shotgun Sequencing" ("WTSS") and dubbed "a revolutionary tool for transcriptomics", refers to the use of high-throughput sequencing technologies to sequence cDNA in order to get information about a sample's RNA content, a technique that is quickly becoming valuable in the study of diseases like cancer.

**RPKM** - Reads Per Kilobase of exon model per Million mapped reads.

**rRNA** - is the RNA component of the ribosome, the protein manufacturing organelle of all living cells.

**SAM format** - SAM (Sequence Alignment/Map) format is a generic format for storing large nucleotide sequence alignments.

**SAMMate** - a Graphical User Interface (GUI) RNA-seq analysis pipeline, allows biomedical researchers to quickly process Fasta/Fastq and SAM/BAM files, and is compatible with both single-end and paired-end sequencing technologies. SAMMate automates some of more standard procedures in RNA-seq analysis.

**SAMtools** - is a set of utilities for interacting with and post-processing short reads alignments in the SAM/BAM format.

**SEED algorithm** - is an algorithm for choosing the initial values (or "seeds") for searching or mapping sequences.

**Sequence alignment** - in bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences.

**Serial Pipeline** - a number of software are executed in sequential order.

**Shotgun Sequencing** - is a method used for sequencing long DNA strands. It is named by analogy with the rapidly-expanding, quasi-random firing pattern of a shotgun.

**Smith-Waterman algorithm** - the Smith-Waterman algorithm is a well-known algorithm for performing local sequence alignment; that is, for determining similar regions between two nucleotide or protein sequences. Instead of looking at the total sequence, the Smith Waterman algorithm compares segments of all possible lengths and optimizes the similarity measure.

**SOAP** - is a bioinformatics package used for the assembly and analysis of DNA sequences. SOAP is particularly well-suited for Illumina next generation sequences. There are 5 members in this package.

**Strand** - a term commonly used to describe one of the two complementary polynucleotide chains found in double-stranded DNA.

**Target** - a cell or organ that is affected by a particular agent, e.g., a hormone or drug.

**Taxon** - is a group of (one or more) organisms, which a taxonomist adjudges to be a unit.

**Taxonomical analysis** - a Taxonomic Analysis is a search for the way that the cultural domains are organized. It usually involves drawing a graphical interpretation of the ways in which the individual participants' moves, form groups and patterns that structure the conversation.

**Taxonomy** - the scientific classification of organisms into specially named groups based either on shared characteristics or on evolutionary relationships as inferred from the fossil record or established by genetic analysis.

**Tomcat web server** - is an open source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Oracle Corporation, and provides a "pure Java" HTTP web server environment for Java code to run.

**TopHat** - is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner Bowtie, and then analyzes the mapping results to identify splice junctions between exons.

**Transcript** - is the process of creating an equivalent RNA copy of a sequence of DNA.

**Transcript quantification** - refers to quantify the transcript abundance using RNA-seq data.

**Transcriptome** - the transcriptome is the set of all RNA molecules, including mRNA, rRNA, tRNA, and other non-coding RNA produced in one or a population of cells.

**tRNA** - is a small RNA molecule (usually about 73-95 nucleotides) that transfers a specific active amino acid to a growing polypeptide chain at the ribosomal site of protein synthesis during translation.

**Two-color design** - refers two independent samples are used in a microarray technology experiment.

**UCSC Genome Browser** - is an on-line genome browser hosted by the University of California, Santa Cruz (UCSC). It is an interactive website offering access to genome sequence data from a variety of vertebrate and invertebrate species and major model organisms, integrated with a large collection of aligned annotations.

**Uniquely mapped reads** - reads which are uniquely mapped on reference genome using aligners.

**Unmapped reads** - Reads which are not mapped any location on reference genome using aligners.

**Wiggle format** - is for display of dense, continuous data such as GC percent, probability scores, and transcriptome data.

## Appendix B

### Important codes

Example: parallelized Novoalign module for JPPF framework:

```
public void submitJob(String jobName, int nbTasks, String filePath, String fileName, String indexFileName)
{
    JPPFJob job = null;
    try {
        job = new JPPFJob();
        job.setId(jobName);

        for (int i = 0; i < nbTasks; i++) {
            NovoAlignTask task = new NovoAlignTask();
            task.setHostName(m_FTPCredential.m_hostName);
            task.setFTPUserName(m_FTPCredential.m_FTPUserName);
            task.setFTPPassword(m_FTPCredential.m_FTPPassword);
            task.setPort(m_FTPCredential.m_port);
            task.setHomeDirectory(m_FTPCredential.m_homeDirectory);
            task.setId("NovoAlign_Task_" + i);
            task.setFileName(fileName + "." + i);

            if (!m_mateFileName.equalsIgnoreCase("")) {
                task.setMateFileName(m_mateFileName + "." + i);
            }

            task.setIndexFileName(indexFileName);
            task.setFASTQFormat(m_fastqFormat);
            job.addTask(task);
        }

        m_logWriter.info("Submitting Novoalign jobs to machines.", false);

        List<JPPFTask> results = getClient().submit(job);
        int totalofMean = 0;
        for (JPPFTask result : results) {
            File file = new File(filePath + fileName + ".sam." + indexFileName);
            FileWriter filewriter = new FileWriter(file, true);
            String resultString = (String) result.getResult();
            int position = resultString.indexOf(", MeanValue:");
            String novoFileName = resultString.substring(9, position);
            int mean = Integer.parseInt(resultString.substring(position + 12));
            totalofMean += mean;
            m_logWriter.info("Mean:" + mean, false);

            String stringLine;
            BufferedReader in = new BufferedReader(new FileReader(filePath + novoFileName));

            while ((stringLine = in.readLine()) != null) {
                if (!stringLine.startsWith("#")) {
                    filewriter.write(stringLine + "\n");
                }
            }
        }
    }
}
```

```

    }

    in.close();
    filewriter.close();
}

m_meanofHumanPairedEnd = (int)((double)totalofMean/(double)results.size());
m_logWriter.info("m_meanofHumanPairedEnd:" + m_meanofHumanPairedEnd, false);

} catch (Exception e) {
    System.err.println(e);
}
}

```

Example: calculate gene RPKM expression:

```

public void calculate(HashMap<String, HashMap<String, Gene>> chromosome, HashMap<String, Vector<String>>
geneRPKMs, double countedReadsNum)
{
    Set<String> keySet = chromosome.keySet();
    Iterator<String> itChromosome = keySet.iterator();

    while (itChromosome.hasNext())
    {
        String chromName = itChromosome.next();
        HashMap<String, Gene> genes = chromosome.get(chromName);

        Set<String> keys = genes.keySet();
        Iterator<String> itGene = keys.iterator();
        while (itGene.hasNext())
        {
            String geneID = itGene.next();
            Gene gene = genes.get(geneID);
            gene.m_expressionValue = 1000000000 *
(double)gene.m_copies/(double)(gene.getTotalExonLength() * countedReadsNum);

            if (countedReadsNum == 0)
                gene.m_expressionValue = 0;

            if (!geneRPKMs.containsKey(chromName + geneID))
            {
                Vector<String> geneRPKM = new Vector<String>();
                geneRPKM.add(chromName);
                geneRPKM.add(gene.m_geneName);
                geneRPKM.add(gene.m_geneID);
                geneRPKM.add(String.valueOf(gene.getTotalExonLength()));
                geneRPKM.add(String.valueOf(gene.m_copies));
                geneRPKM.add(String.valueOf(gene.m_expressionValue));
                geneRPKMs.put(chromName + geneID, geneRPKM);
            }
            else
            {
                Vector<String> geneRPKM = geneRPKMs.get(chromName + geneID);
                geneRPKM.add(String.valueOf(gene.m_copies));
                geneRPKM.add(String.valueOf(gene.m_expressionValue));
            }
        }
    }
}

```

Example: calculate transcript expression using iterative procedure of iQuant algorithm:

```

/**
 * The following method is to estimate proportions using iterative procedure
 * with known gene expression score.
 * @param E
 * @param W
 * @param gene
 * @return proportions
 */
private Matrix calculate(Matrix E, Matrix W, double score, Gene gene)
{
    long sampleNum = E.getColumnCount();
    long lengthExons = E.getRowCount();
    long isoNum = W.getColumnCount();

    Matrix S = W.clone();
    Matrix A = DenseMatrix.factory.ones(1,isoNum);
    int b = 1;
    Matrix z = E.toRowVector(Ret.NEW);
    Matrix isoProps = DenseMatrix.factory.ones(1,isoNum);
    Matrix newIsoProps = DenseMatrix.factory.zeros(1,isoNum);
    int iter = 0;

    try
    {
        while(isoProps.minus(newIsoProps).normF() > m_epsl && iter < m_maximalSteps)
        {
            isoProps = newIsoProps.clone();
            Matrix rs = DenseMatrix.factory.zeros(1,sampleNum);

            for (int i = 0; i < sampleNum; i++)
            {
                Matrix zero = DenseMatrix.factory.ones(isoNum,1);
                Matrix one = (Matrix)E.selectColumns(Ret.NEW,
i.transpose().mtimes(W).mtimes(zero);
                Matrix two = DenseMatrix.factory.ones(isoNum,1);
                double three = (W.mtimes(two)).normF();
                Matrix value = one.divide(three*three);

                if (iter == 0)
                    rs.setAsDouble(score, 0, i);
                else
                    rs.setAsDouble(value.doubleValue(), 0, i);
            }

            Matrix H = DenseMatrix.factory.zeros(0,0);

            for (int i = 0; i < sampleNum; i++)
                H = H.appendVertically(S.times(rs.getAsDouble(0,i)));
        }
    }
}

```

```

        Matrix temp0 = (H.transpose().mtimes(H));
        Matrix xHatLS = null;

        if (!temp0.isSingular())
            xHatLS = temp0.inv().mtimes(H.transpose()).mtimes(z);
        else
            return null;

        Matrix temp5 = (H.transpose().mtimes(H));
        if (temp5.isSingular())
            return null;

        Matrix B = temp5.inv();
        Matrix temp4 = (A.mtimes(B).mtimes(A.transpose()));

        if (temp4.isSingular())
            return null;

        Matrix part1 = B.mtimes(A.transpose()).mtimes(temp4.inv());

        Matrix temp1 = DenseMatrix.factory.eye(isoNum,isoNum);
        Matrix temp2 = temp1.minus(part1.mtimes(A)).mtimes(xHatLS);
        Matrix xHatCLS = temp2.plus(part1.times(b));

        newIsoProps = xHatCLS.transpose();
        W = DenseMatrix.factory.zeros(lengthExons,isoNum);

        Matrix temp = DenseMatrix.factory.eye(isoNum,isoNum);
        for (int i = 0; i < isoNum; i++)
            temp.setAsDouble(newIsoProps.getAsDouble(0,i), i,i);
        W = S.mtimes(temp);

        iter++;

        gene.setPredictedExpValue(rs.getAsDouble(0,0));
    }
} catch (Exception e)
{
    System.out.println("The gene cannot be calculated: " + gene.m_geneID);
    System.err.println(e.getMessage());
    return null;
}

Matrix E_mean = DenseMatrix.factory.ones(lengthExons,1).times(E.getMeanValue());
Matrix rs_v = DenseMatrix.factory.ones(isoNum,1).times(gene.m_predictedExpValue);
Matrix dif1 = E.minus(W.mtimes(rs_v));
Matrix dif1_square = dif1.transpose().mtimes(dif1);

Matrix dif2 = E.minus(E_mean);
Matrix dif2_square = dif2.transpose().mtimes(dif2);
gene.m_Rsquare = 1 - dif1_square.doubleValue()/dif2_square.doubleValue();

return isoProps;
}

```



Example: Detects differentially expressed genes and isoforms with support for edgeR:

```
public String assembleCommands(String inputFile, String outputFile)
{
    inputFile = "D:/EdgeR Data/LungDataTransReadCounts.txt";
    outputFile = "D:/EdgeR Data/output.txt";
    inputFile = "" + inputFile + "";
    outputFile = "" + outputFile + "";

    StringBuffer commands = new StringBuffer();
    commands.append("library(edgeR);");
    commands.append("raw.data<-read.delim(" + inputFile + ");");
    commands.append("d<-raw.data[,2:" + (m_sampleNum + 1) + "];");
    commands.append("rownames(d)<-raw.data[,1];");
    commands.append("group<-c(rep('GroupA'," + m_groupANum + "),rep('GroupB'," + m_groupBNum +
    "));");
    commands.append("d<-DGEList(counts=d,group=group);");
    commands.append("d<-estimateCommonDisp(d);");
    commands.append("de.com<-exactTest(d);");
    commands.append("write.table(topTags(de.com, n=" + m_featureNum + ")$table," + outputFile +
    ",sep=\"\\t\\n\");");
    return commands.toString();
}
```

Example: Generates coverage file in wiggle format for visualization:

```
private void buildForwardWig(String outputFile)
{
    int maxReadLength = SystemProperties.getInstance().getProperty("maxReadLength", 100);
    m_logWriter.info("The maximal length of short read is: " + maxReadLength, false);

    try
    {
        Set<String> keySet = m_samCoordinates.keySet();
        Iterator<String> itChromosome = keySet.iterator();

        if (m_isBuildCoverage) {
            File file = new File(outputFile);
            String fileIndex = outputFile.substring(outputFile.lastIndexOf("\\") + 1, outputFile.length());

            m_filewriter = new FileWriter(file, false);
            m_filewriter.write("track type=bedGraph name=\"read coverage " + fileIndex + "\" + \"\\n\"");
        }

        while (itChromosome.hasNext()) {
            String chromName = itChromosome.next();
            int[][] samCoverages = m_samCoverages.get(chromName);
            int[][] bedCoverages = null;
            if (m_bedCoverages != null) {
                bedCoverages = m_bedCoverages.get(chromName);
            }

            CombinedCoordinate combinedCoordinate = null;
            if (m_bedCoordinates != null) {

```

```

        combinedCoordinate = new CombinedCoordinate(m_samCoordinates.get(chromName),
m_bedCoordinates.get(chromName));
    } else {
        combinedCoordinate = new CombinedCoordinate(m_samCoordinates.get(chromName), null);
    }

    int startPosition = 0;
    int endPosition = 0;
    ArrayList<Integer> positions = new ArrayList<Integer>();
    positions.add(startPosition);

    while (combinedCoordinate.hasNextPosition()) {
        int currentPosition = combinedCoordinate.getNextPosition();
        if (currentPosition < 0) {
            continue;
        }

        positions.add(currentPosition);

        int[][] read = m_util.getRead(samCoverages, currentPosition);
        if (read == null && bedCoverages != null) {
            read = m_util.getRead(bedCoverages, currentPosition);
        }

        if (currentPosition == combinedCoordinate.getLastPosition() && read != null) {
            positions.add(currentPosition + read[2][0]);
        }

        if (currentPosition - endPosition <= maxReadLength && currentPosition !=
combinedCoordinate.getLastPosition()) {
            endPosition = currentPosition;
        } else {
            // build coverage file from start position to end position;
            outputForwardWigFile(samCoverages, bedCoverages, chromName, positions, outputFile);
            positions = new ArrayList<Integer>();
            startPosition = currentPosition;
            endPosition = currentPosition;
            positions.add(startPosition);
        }

        if (currentPosition == combinedCoordinate.getLastPosition()) {
            outputForwardWigFile(samCoverages, bedCoverages, chromName, positions, outputFile);
        }
    }

    if (m_isBuildCoverage) {
        m_filewriter.close();
    }

    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

## References

- [1] Adams MD, Soares MB, Kerlavage AR, Fields C, Venter JC. (1993) Rapid cDNA sequencing (expressed sequence tags) from a directionally cloned human infant brain cDNA library. *Nature Genet.* 4, 373-380.
- [2] Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman D J. (1997) *Nucleic Acids Res.* 25, 3389-3402.
- [3] Arner E, Hayashizaki Y, Daub CO: NGSView: an extensible open source editor for next-generation sequencing data. *Bioinformatics* (Oxford, England). (2010) 26:125-126, [<http://dx.doi.org/10.1093/bioinformatics/btp611>].
- [4] Bao H, Guo H, Wang J, Zhou R, Lu X, Shi S. (2009) MapView: visualization of short reads alignment on a desktop computer. *Bioinformatics*, 25(12):1554-1555.
- [5] Barbazuk WB, Emrich SJ, Chen HD, Li L, Schnable PS. (2007) "SNP discovery via 454 transcriptome sequencing". *The Plant Journal* 51 (5): 910-918.
- [6] Benson DA, Karsh-mizrachi I, Lipman DJ, Ostell J & Sayers EW. (2010) GenBank. *Nucleic acids research* 38(s1): D46-D51.
- [7] Birol I, Jackman SD, Nielsen CB, Qian JQ, Varhol R, Stazyk G, Morin RD, Zhao Y, Hirst M, Schein JE, Horsman DE, Connors JM, Gascoyne RD, Marra MA, Jones SJ. (2009) De novo transcriptome assembly with ABySS. *Bioinformatics*, 25:2872-2877.
- [8] Bohnert R, Rätsch G. (2010) rQuant.web: a tool for RNA-Seq-based transcript quantitation, *Nucleic Acids Research*, 38(Suppl 2):W348-51.
- [9] Burrows M, Wheeler D. (1994) A block sorting lossless data compression algorithm, Technical Report 124, Digital Equipment Corporation

- [10] Cameron JE, Yin Q, Fewell C, Lacey M, McBride J, Wang X, Lin Z, Schaefer BC, Flemington EK. (2008) Epstein-Barr virus latent membrane protein 1 induces cellular MicroRNA miR-146a, a modulator of lymphocyte signaling pathways. *J Virol* 82: 1946–1958.
- [11] Chen J, Wang YP. (2009) A Statistical Change Point Model Approach for the Detection of DNA Copy Number Variations in Array CGH Data. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 6(4):529-541.
- [12] Chepelev I, Wei G, Tang Q, and Zhao K. (2009) “Detection of single nucleotide variations in expressed exons of the human genome using RNA-Seq,” *Nucleic Acids Research*, vol. 37, no.16, article e106.
- [13] Chi SW, Zang JB, Mele A, Darnell RB. (2009) Argonaute HITS-CLIP decodes microRNA–mRNA interaction maps. *Nature* 460: 479–486.
- [14] Cloonan N, Forrest AR, Kolle G, Gardiner BB, Faulkner GJ, Brown MK, Taylor DF, Steptoe AL, Wani S, Bethel G, Robertson AJ, Perkins AC, Bruce SJ, Lee CC, Ranade SS, Peckham HE, Manning JM, McKernan KJ, Grimmond SM. (2008). Stem cell transcriptome profiling via massive-scale mRNA sequencing, *Nat Methods*. 5(7):613-9
- [15] Clurman BE, Hayward WS. (1989) Multiple proto-oncogene activations in avian leukemia virus-induced lymphomas: Evidence for stage-specific events. *Mol Cell Biol* 9: 2657–2664.
- [16] Coco J, Flemington EK, and Taylor C. (2011) PARSES: A Pipeline for Analysis of RNA-Seq Exogenous Sequences. *ISCA 3rd International Conference on Bioinformatics and Computational Biology*, New Orleans, LA.
- [17] Costa V, Angelini C, De Feis I, and Ciccodicola A. (2010) Uncovering the complexity of transcriptomes with RNA-Seq. *J Biomed Biotechnol*, pp. 853916, ISSN 1110-7251

- [18] Costinean S, Zanesi N, Pekarsky Y, Tili E, Volinia S, Heerema N, Croce CM. (2006) Pre-B cell proliferation and lymphoblastic leukemia/high-grade lymphoma in Em-miR155 transgenic mice. *Proc Natl Acad Sci* 103: 7024–7029.
- [19] Dabney AR, Storey JD. (2007) Normalization of two-channel microarrays accounting for experimental design and intensity-dependent relationships. *Genome Biol* 8: R44. doi: 10.1186/gb-2007-8-3-r44.
- [20] Deng N, Puetter A, Zhang K, Johnson K, Zhao Z, Taylor C, Flemington E and Zhu D. (2011) Isoform-level microRNA-155 Target Prediction using RNA-seq. *Nuc. Acid Res.*, doi: 10.1093/nar/gkr042
- [21] Durbin R, Eddy S, Krogh A, and Mitchison G. (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.
- [22] Edwards RH, Marquitz AR, and Raab-Traub N. (2008) Epstein-Barr virus BART microRNAs are produced from a large intron prior to splicing. *J. Virol.* 82:9094-9106.
- [23] Fujita PA, Rhead B, Zweig AS, Hinrichs AS, Karolchik D, Cline MS, Goldman M, Barber GP, Clawson H, Coelho A, Diekhans M, Dreszer TR, Giardine BM, Harte RA, Hillman-Jackson J, Hsu F, Kirkup V, Kuhn RM, Learned K, Li CH, Meyer LR, Pohl A, Raney BJ, Rosenbloom KR, Smith KE, Haussler D, Kent WJ. (2010) The UCSC Genome Browser database: Update. *Nucleic Acids Res* 38:D613–619.
- [24] Garcia-Blanco MA, Baraniak AP, Lasda EL. (2004) Alternative splicing in disease and therapy. *Nat Biotechnol*; 22:535– 46.
- [25] Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JY, Zhang J. (2004) Bioconductor:

Open software development for computational biology and bioinformatics. *Genome Biol* 5: R80.  
doi: 10.1186/gb-2004-5-10-r80.

[26] Gottwein E, Mukherjee N, Sachse C, Frenzel C, Majoros WH, Chi JT, Braich R, Manoharan M, Soutschek J, Ohler U, Cullen BR. (2007) A viral microRNA functions as an orthologue of cellular miR-155. *Nature* 450: 1096–1099.

[27] Greenbaum D, Colangelo C, Williams K, Gerstein M. (2003) "Comparing protein abundance and mRNA expression levels on a genomic scale". *Genome Biology* 4 (9): 117.

[28] Grimson A, Farh KK, Johnston WK, Garrett-Engele P, Lim LP, Bartel DP. (2007) MicroRNA targeting specificity in mammals: Determinants beyond seed pairing. *Mol Cell* 27: 91–105.

[29] Grover CE, Salmon A, and Wendel JF. (2012). Targeted sequence capture as a powerful tool for evolutionary analysis. *American Journal of Botany* 99:312-319

[30] Hammell M. (2010) Computational methods to identify miRNA targets. *Semin Cell Dev Biol* doi: 10.1016/j.semcdb.2010.01.004.

[31] Huson DH, Auch AF, Qi J, Schuster SC. (2007) MEGAN analysis of metagenomic data. *Genome Res* 17:377–386.

[32] Iorio MV, Ferracin M, Liu CG, Veronese A, Spizzo R, Sabbioni S, Magri E, Pedriali M, Fabbri M, Campiglio M, Ménard S, Palazzo JP, Rosenberg A, Musiani P, Volinia S, Nenci I, Calin GA, Querzoli P, Negrini M, Croce CM. (2005) MicroRNA gene expression deregulation in human breast cancer. *Cancer Res* 65: 7065–7070.

[33] Jiang H, Wong WH. (2008) SeqMap : mapping massive amount of oligonucleotides to the genome. *Bioinformatics*, 24(20):btm429-2396.

- [34] Jiang H, Wong WH. (2008) SeqMap: Mapping massive amount of oligonucleotides to the genome. *Bioinformatics* 24: 2395–2396.
- [35] Jiang H, Wong WH. (2009) Statistical inferences for isoform expression in RNA-seq. *Bioinformatics*, 25(8):1026-1032, [<http://dx.doi.org/10.1093/bioinformatics/btp113>].
- [36] Jiang J, Lee EJ, Schmittgen TD. (2006) Increased expression of micro-RNA-155 in Epstein-Barr virus transformed lymphoblastoid cell lines. *Genes Chromosomes Cancer* 45: 103–106.
- [37] Kircher M, Heyn P, and Kelso J. (2011) Addressing challenges in the production and analysis of Illumina sequencing data. *BMC Genomics* 12:382.
- [38] Kluiver J, Haralambieva E, de Jong D, Blokzijl T, Jacobs S, Kroesen BJ, Poppema S, van den Berg A. (2006) Lack of BIC and microRNA miR-155 expression in primary cases of Burkitt lymphoma. *Genes Chromosomes Cancer* 45: 147–153.
- [39] Kluiver J, Poppema S, de Jong D, Blokzijl T, Harms G, Jacobs S, Kroesen BJ, van den Berg A. (2005) BIC and miR-155 are highly expressed in Hodgkin, primary mediastinal and diffuse large B cell lymphomas. *J Pathol* 207: 243–249.
- [40] Langmead B, Trapnell C, Pop M, Salzberg S. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25, [<http://genomebiology.com/2009/10/3/R25>].
- [41] Langmead B, Trapnell C, Pop M, Salzberg SL. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* 10:R25.
- [42] Li B, Ruotti V, Stewart RM, Thomson JA, Dewey CN. (2010) RNA-seq gene expression estimation with read mapping uncertainty. *Bioinformatics* 26: 493–500.

- [43] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis GR, Durbin R. (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078-2079.
- [44] Li H, Ruan J, Durbin R. (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res*, [<http://dx.doi.org/10.1101/gr.078212.108>].
- [45] Li L, Xu J, Yang D, Tan X, Wang H. (2010) Computational approaches for microRNA studies: A review. *Mamm Genome* 21:1–12.
- [46] Li P, Ponnala L, Gandotra N, Wang L, Si Y, Tausta SL, Kebrom TH, Provart N, Patel R, Myers CR, Reidel EJ, Turgeon R, Liu P, Sun Q, Nelson T, Brutnell TP. (2010) The developmental dynamics of the maize leaf transcriptome. *Nat. Genet.* 42, 1060–1067.
- [47] Li R, Li Y, Kristiansen K, Wang J. (2008) SOAP: short oligonucleotide alignment program. *Bioinformatics*, 24:713-714.
- [48] Lin Z, Puetter A, Coco J, Xu G, Strong MJ, Wang X, Fewell C, Baddoo M, Taylor C, Flemington EK. (2012) Detection of murine leukemia virus in the Epstein-Barr virus-positive human B-cell line JY, using a computational RNA-Seq-based exogenous agent detection pipeline, PARSES. *J Virol.* 86(6):2970-7
- [49] Lin Z, Xu G, Deng N, Taylor C, Zhu D, Flemington EK. (2010) Quantitative and qualitative RNA-Seq-based evaluation of Epstein-Barr virus transcription in type 1 latency Burkitt's lymphoma cells. *J. Virol.*; 84 (24);13053-8
- [50] Liu J, Jennings SF, Tong W, Hong H. (2011) Next generation sequencing for profiling expression of miRNAs: technical progress and applications in drug development, *J. Biomedical Science and Engineering*, 4, 666-676



- [51] Liu ZJ. (2007) Fish genomics and analytical genetic technologies, with examples of their potential applications in management of fish resources. In: Bartley DM, Harvey BJ, Pullin RSV (eds) FAO Fisheries Proceedings 5. Workshop on status and trends in aquatic genetic resources. FAO, Rome, p 145–179
- [52] Lu J, Getz G, Miska EA, Alvarez-Saavedra E, Lamb J, Peck D, Sweet-Cordero A, Ebert BL, Mak RH, Ferrando AA, Downing JR, Jacks T, Horvitz HR, Golub TR. (2005) MicroRNA expression profiles classify human cancers. *Nature* 435: 834–838.
- [53] Maher CA, Kumar-Sinha C, Cao X, Kalyana-Sundaram S, Han B, Jing X, Sam L, Barrette T, Palanisamy N, Chinnaiyan AM. (2009) "Transcriptome sequencing to detect gene fusions in cancer". *Nature* 458 (7234): 97-101.
- [54] Manske HM, Kwiatkowski DP. (2009) LookSeq: a browser-based viewer for deep sequencing data. *Genome Res*; 19(11):2125–32.
- [55] Marioni JC, Mason CE, Mane SM, Stephens M, Gilad Y. (2008) RNAseq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res* 18: 1509–1517.
- [56] Mayr C, Bartel DP. (2009) Widespread shortening of 3'UTRs by alternative cleavage and polyadenylation activates oncogenes in cancer cells. *Cell* 138: 673–684.
- [57] Metzker ML. (2010) Sequencing technologies – the next generation. *Nat Rev Genet*. 11:31–46.
- [58] Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B. (2008) "Mapping and quantifying mammalian transcriptomes by RNA-Seq". *Nature Methods* 5 (7): 621-628.

- [59] Needleman, Saul B.; and Wunsch, Christian D. (1970) "A general method applicable to the search for similarities in the amino acid sequence of two proteins". *Journal of Molecular Biology* 48 (3): 443–53. doi:10.1016/0022-2836(70)90057-4
- [60] Neff NF, Newberry KM, Garabedian MJ, Myers RM. (2009) "Genomic determination of the glucocorticoid response reveals unexpected mechanisms of gene regulation," *Genome Research*, vol. 19, no. 12, pp. 2163–2171
- [61] Nguyen T, Deng N, Xu G, Duan Z, Zhu D, (2011) "iQuant: A fast yet accurate GUI tool for transcript quantification," *bioinformatics*, pp.1048-1050, 2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops
- [62] O'Connell RM, Rao DS, Chaudhuri AA, Boldin MP, Taganov KD, Nicoll J, Paquette RL, Baltimore D. (2008) Sustained expression of microRNA-155 in hematopoietic stem cells causes a myeloproliferative disorder. *J Exp Med* 205: 585–594.
- [63] O'Connell RM, Taganov KD, Boldin MP, Cheng G, Baltimore D. (2007) MicroRNA-155 is induced during the macrophage inflammatory response. *Proc Natl Acad Sci* 104: 1604–1609.
- [64] Olden K, Freudenberg N, Dowd J, Shields AE. (2011) Discovering how environmental exposures alter genes could lead to new treatments for chronic illnesses. *Health Aff (Millwood)*. 30:833-41.
- [65] Pacheco TR, Moita LF, Gomes AQ, Hacohen N and Carmo-Fonseca M. (2006) RNA interference knockdown of hU2AF35 impairs cell cycle progression and modulates alternative splicing of Cdc25 transcripts. *Mol. Biol. Cell*, 17, 4187–4199
- [66] Paşaniuc B, Zaitlen N, Halperin E. (2010) Accurate estimation of expression levels of homologous genes in RNA-seq experiments. In *Research in Computational Molecular Biology*.

Edited by: Berger B. Berlin/Heidelberg: Springer; 397-409, [Lecture Notes in Computer Science, vol 6044].

[67] Patterson TA, Lobenhofer EK, Fulmer-Smentek SB, Collins PJ, Chu TM, Bao W, Fang H, Kawasaki ES, Hager J, Tikhonova IR, Walker SJ, Zhang L, Hurban P, de Longueville F, Fuscoe JC, Tong W, Shi L, Wolfinger RD. (2006) Performance comparison of one-color and two-color platforms within the MicroArray Quality Control (MAQC) project. *Nat. Biotechnol.* 24, 1140–1150

[68] Rodriguez A, Vigorito E, Clare S, Warren MV, Couttet P, Soond DR, van Dongen S, Grocock RJ, Das PP, Miska EA, Vetrie D, Okkenhaug K, Enright AJ, Dougan G, Turner M, Bradley A. (2007) Requirement of bic/microRNA-155 for normal immune function. *Science* 316: 608–611.

[69] Roy NC, Altermann E, Park ZA, and McNabb WC. (2011). A comparison of analog and Next-Generation transcriptomic tools for mammalian studies. *Brief Funct. Genomics* 10: 135–150.

[70] Russo D, Ambrosino A, Vittoria A, Cecio A. (2003) Signal amplification by combining two advanced immunohistochemical techniques. *Eur J Histochem* 47:379–384

[71] Russo G, Zegar C, Giordano A. (2003) Advantages and limitations of microarray technology in human cancer. *Oncogene*, 22:6497-6507

[72] Ryan D. Morin, Matthew Bainbridge, Anthony Fejes, Martin Hirst, Martin Krzywinski, Trevor J. Pugh, Helen McDonald, Richard Varhol, Steven J.M. Jones, and Marco A. Marra. (2008). "Profiling the HeLa S3 transcriptome using randomly primed cDNA and massively parallel short-read sequencing". *BioTechniques* 45 (1): 81-94.

- [73] Sandberg R, Neilson JR, Sarma A, Sharp PA, Burge CB. (2008) Proliferating cells express mRNAs with shortened 3' untranslated regions and fewer microRNA target sites. *Science* 320: 1643–1647.
- [74] Scholin C, Miller P, Buck K, Chavez F, Harris P, Haydock P, Howard J & Cangelosi G. (1997). Detection and quantification of *Pseudo-nitzschia australis* in cultured and natural populations using LSU rRNA-targeted probes. *Limnol. Oceanogr.*, 42: 1265-1272.
- [75] Sekirov I, Russell S L, Antunes L C, Finlay B B. (2010) Gut microbiota in health and disease. *Physiol. Rev.* 90, 859–904
- [76] Selbach M, Schwanhaussner B, Thierfelder N, Fang Z, Khanin R, Rajewsky N. (2008) Widespread changes in protein synthesis induced by microRNAs. *Nature* 455: 58–63.
- [77] Shalon D, Smith SJ, Brown PO. (1996) A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Res* 6 (7): 639–645. doi:10.1101/gr.6.7.639
- [78] Skalsky RL, Samols MA, Plaisance KB, Boss IW, Riva A, Lopez MC, Baker HV, Renne R. (2007) Kaposi's sarcoma-associated herpesvirus encodes an ortholog of miR-155. *J Virol* 81: 12836–12845.
- [79] Smith AD, Xuan Z, Zhang MQ. (2008) Using quality scores and longer reads improves accuracy of Solexa read mapping. *BMC Bioinformatics*, 9:128.
- [80] Smith, Temple F.; and Waterman, Michael S. (1981) "Identification of Common Molecular Subsequences". *Journal of Molecular Biology* 147: 195–197. doi:10.1016/0022-2836(81)90087-5

- [81] Smyth GK. (2005) Limma: Linear models for microarray data. In *Bioinformatics and computational biology solutions using R and Bioconductor* (ed. R Gentleman et al.), pp 397–420. Springer, New York.
- [82] Storey JD, Tibshirani R. (2003) Statistical significance for genomewide studies. *Proc Natl Acad Sci* 100: 9440–9445.
- [83] Tam W, Ben-Yehuda D, Hayward WS. (1997) bic, a novel gene activated by proviral insertions in avian leukosis virus-induced lymphomas, is likely to function through its noncoding RNA. *Mol Cell Biol* 17: 1490–1502.
- [84] The R Development Core Team. (2008) R: A language and environment for statistical computing. R Foundation for Statistical Computing.
- [85] Trapnell C, Pachter L, Salzberg SL (2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9):1105-1111, [<http://dx.doi.org/10.1093/bioinformatics/btp120>].
- [86] Tuller T, Kupiec M, Ruppín E. (2007) Determinants of protein abundance and translation efficiency in *S. cerevisiae*. *PLoS Comput Biol* 3:2510–2519.
- [87] Tusher VG, Tibshirani R, Chu G. (2001) Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci* 98: 5116–5121.
- [88] Valerio C, Claudia A, Italia DF, and Alfredo C, (2010) Uncovering the Complexity of Transcriptomes with RNA-Seq, *Journal of Biomedicine and Biotechnology* Volume, Article ID 853916
- [89] Vladimir L, (2005) “The Needleman–Wunsch Algorithm for Sequence Alignment”, <http://www.ludwig.edu.au/course/lectures2005/Likic.pdf>
- [90] Volinia S, Calin GA, Liu CG, Ambs S, Cimmino A, Petrocca F, Visone R, Iorio M, Roldo C, Ferracin M, Prueitt RL, Yanaihara N, Lanza G, Scarpa A, Vecchione A, Negrini M,

Harris CC, Croce CM. (2006) A microRNA expression signature of human solid tumors defines cancer gene targets. *Proc Natl Acad Sci* 103: 2257–2261.

[91] Wang Z, Gerstein M, Snyder M. (2009) RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10:57-63, [<http://dx.doi.org/10.1038/nrg2484>].

[92] Xu G, Deng N, Zhao Z, Zhang K, Judeh T, Flemington EK and Zhu D. (2011) SAMMate: A GUI tool for processing short read alignment information in SAM/BAM format. *Source Code for Biology and Medicine*, 6:2.

[93] Xu G, Deng N, Zhao Z, Zhang K, Judeh T, Zhu D, Flemington EK. (2010): Transcriptome and targetome analysis in miR-155 expressing cells using RNA-seq. *RNA*.

[94] Yin Q, McBride J, Fewell C, Lacey M, Wang X, Lin Z, Cameron J, Flemington EK. (2008) MicroRNA-155 is an Epstein-Barr virus-induced gene that modulates Epstein-Barr virus-regulated gene expression pathways. *J Virol* 82: 5295–5306.

[95] Yin Q, Wang X, Fewell C, Cameron J, Zhu H, Baddoo M, Lin Z, Flemington EK. (2010) MiR-155 inhibits bone morphogenetic protein (BMP) signaling and BMP mediated Epstein-Barr virus reactivation. *J Virol* 84: 6318–6327.

[96] Zhang L, Huang J, Yang N, Greshock J, Megraw MS, Giannakakis A, Liang S, Naylor TL, Barchetti A, Ward MR, Yao G, Medina A, O'Brien-Jenkins A, Katsaros D, Hatzigeorgiou A, Gimotty PA, Weber BL, Coukos G. (2006) microRNAs exhibit high frequency genomic alterations in human cancer. *Proc Natl Acad Sci* 103: 9136–9141.

[97] Zheng S, Chen L: A hierarchical Bayesian model for comparing transcriptomes at the individual transcript isoform level. *Nucl. Acids Res.* (2009) 37(10):e75+, [<http://dx.doi.org/10.1093/nar/gkp282>].

[98] Zheng ZB, Wu YD, Yu XL, and Shang SQ. (2008) DNA microarray technology for simultaneous detection and species identification of seven human herpes viruses. *J Med Virol* 80:1042-50.

## **Vita**

Guorong Xu was born in Hunan province, China. He received his first Bachelor degree in the department of Mathematics and Computer Science from Jishou University in 2001 and received the second Bachelor degree in the School of Software from Tsinghua University in 2003.

In 2008, he started his Ph.D. study in the Department of Computer Science at the University of New Orleans. He received a M.S. degree in Computer Science in 2011. After the completion of his Ph.D. study, he will work at Baylor Health Care System as a Research Associate.