ABSTRACT

Title of dissertation: TRANSCRIPT ASSEMBLY
AND ABUNDANCE ESTIMATION
WITH HIGH-THROUGHPUT RNA SEQUENCING

Bruce C. Trapnell, Jr., Doctor of Philosophy, 2010

Dissertation directed by: Professor Steven Salzberg
Department of Computer Science

We present algorithms and statistical methods for the reconstruction and abundance estimation of transcript sequences from high throughput RNA sequencing ("RNA-Seq"). We evaluate these approaches through large-scale experiments of a well studied model of muscle development.

We begin with an overview of sequencing assays and outline why the short read alignment problem is fundamental to the analysis of these assays. We then describe two approaches to the contiguous alignment problem, one of which uses massively parallel graphics hardware to accelerate alignment, and one of which exploits an indexing scheme based on the Burrows-Wheeler transform. We then turn to the spliced alignment problem, which is fundamental to RNA-Seq, and present an algorithm, TopHat. TopHat is the first algorithm that can align the reads from an entire RNA-Seq experiment to a large genome without the aid of reference gene models.

In the second part of the thesis, we present the first comparative RNA-Seq as-

sembly algorithm, Cufflinks, which is adapted from a constructive proof of Dilworth's Theorem, a classic result in combinatorics. We evaluate Cufflinks by assembling the transcriptome from a time course RNA-Seq experiment of developing skeletal muscle cells. The assembly contains 13,689 known transcripts and 3,724 novel ones. Of the novel transcripts, 62% were strongly supported by earlier sequencing experiments or by homologous transcripts in other organisms. We further validated interesting genes with isoform-specific RT-PCR.

We then present a statistical model for RNA-Seq included in Cufflinks and with which we estimate abundances of transcripts from RNA-seq data. Simulation studies demonstrate that the model is highly accurate. We apply this model to the muscle data, and track the abundances of individual isoforms over development.

Finally, we present significance tests for changes in relative and absolute abundances between time points, which we employ to uncover differential expression and differential regulation. By testing for relative abundance changes within and between transcripts sharing a transcription start site, we find significant shifts in the rates of alternative splicing and promoter preference in hundreds of genes, including those believed to regulate muscle development.

TRANSCRIPT ASSEMBLY AND ABUNDANCE ESTIMATION WITH HIGH-THROUGHPUT RNA SEQUENCING

by

Bruce C. Trapnell, Jr.

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Advisory Committee:
Professor Steven L. Salzberg Chair/Advisor
Professor Lior Pachter, Co-Advisor
Professor Mihai Pop
Professor Carl Kingsford
Professor Steven Mount

Dedication

To those that first led me to science: Bruce Trapnell, Sr., Herbert Wood, and Donal Sullivan.

Acknowledgments

I performed the work in this thesis under the guidance of Steven Salzberg and Lior Pachter. Their training, advice, and technical contributions made the work possible, and I am in their debt.

The content of this thesis has either appeared in published articles or is currently in press, and I am grateful to a large group of talented collaborators for allowing me to include our joint work.

Chapter 1 overviews short read sequencing assays, a methodology I expect to transform molecular biology by greatly amplifying the need for principled and well engineered computation in routine experiments. I am far from alone in holding this view. My perspective was developed through conversations with the collaborators listed here, but particularly with Robert Bradley, Ali Mortazavi, Michael Schatz, Ben Langmead, and of course my advisors Steven Salzberg and Lior Pachter.

Chapter 2 describes three alignment programs for short reads: MUMmerGPU, Bowtie, and TopHat. MUMmerGPU is joint work with Mike Schatz, Amitabh Varshney, and Art Delcher. Mike and I wrote MUMmerGPU and describe it in two co-first author papers, and it was my first bioinformatics project. His patience, mentoring, and technical advice then and throughout my graduate work were invaluable. Bowtie is joint work with Ben Langmead, Mihai Pop, and Steven Salzberg. Ben is the primary author of Bowtie and extending the search algorithm of Paolo Ferragina and Giovanni Manzini to handle mismatches in short read alignments was his idea. Ben was generous to allow me to help him execute Bowtie, and his creativity and

engineering skills resulted in a program on which I was able to build the stack of RNA-Seq software presented here. I wrote TopHat, but the idea was suggested by Lior, and Steven, Lior, and I designed the experiments for evaluating it and wrote the paper together.

Chapters 3, 4, and 5 describe the algorithms and mathematics in Cufflinks, which Lior, and I have spent most of our time and energy on for over a year. Lior has been a constant source of good ideas, arguments against bad ones, moral support, and rescue from failure. I implemented the Cufflinks assembler and the associated statistical methods, but there is a third component to the package that has proved invaluable. Geo Pertea wrote "cuffcompare", which solves the subtle and difficult problem of comparing two transcriptome assemblies, or comparing one assembly to a set of known transcripts. Geo has endured numerous feature requests, specification changes, crisis bug reports that turned out to be my fault, and weekend work for the sake of my PhD project. He is a generous man with his time. Jeltje van Baren generated a number of bug reports for both TopHat and Cufflinks through months of diligent testing, and both programs are robust because of her work and frustration.

The paper describing Cufflinks is joint work with Steven, Geo, Jeltje, and our collaborators from Caltech: Ali Mortazavi, Brian Williams, Gordon Kwan, and Barbara Wold. When I first joined CBCB as a student, I started reading papers in bioinformatics, and the first I read was Ali's paper with Barbara, Rick Myers, and David Johnson describing ChIP-Seq. While Ben and I were writing Bowtie, Ali and Brian and their colleagues described RNA-Seq in a paper which convinced me that sequencing assays will revolutionize molecular biology. Through a truly

auspicious set of events, we came to work together to try and "solve RNA-Seq", as Ali puts it. It has been my great honor to work with them. Ali has patiently mentored me and inspired me to try and become a computational biologist as well as a computer scientist. Brian Williams, with some help from Gordon, generated nearly all of the wet data discussed in this thesis. This is an enormous contribution of both time and creativity - the RNA-Seq protocol used in the experiment in this thesis is substantially improved over the one in his original paper. It produces paired end RNA-Seq reads, which dramatically improve the accuracy of Cufflinks' transcript abundance estimates. When Lior and I first discussed the idea of TopHat, we knew ultimately we wanted a program like Cufflinks. We had a vague idea that with the abundances of individual transcripts, one could discover some interesting biology, but the idea of inferring regulatory changes from abundance changes began with Barbara. She has shaped the biological analysis and interpretation throughout Cufflinks' development. Barbara also generously donated to the paper not only an enormous RNA-Seq dataset, but also the data from a set of ChIP-Seq experiments that greatly strengthen the assembly validation.

I am grateful to all of the users of Bowtie, TopHat, and Cufflinks, who have spent their own time and in many cases donated their data to help test and improve the software. I am especially grateful (in no particular order) to Mitchell Guttman, John Rinn, Kat Chang, Bob Schmitz, Karen Power, Natalie Twine, Irina Khrebtukova, Gary Schroth, Diane Trout, Kasper Hansen, Angela Brooks, Steffen Durinck, Richard McCombie, Todd Wylie, Elaine Mardis, and Paul Flicek. I particularly value the feedback of the modENCODE RNA group, including: Jane

Landolin, Dave Sturgill, Mike Duff, Sue Celniker, Brian Oliver, and Brenton Graveley.

I was fortunate to help Rob Bradley with his work on a ChIP-Seq project with Mike Eisen. I learned a great deal from both of them during that project. I'm also thankful for helpful discussions with Kiril Datchev, Art Delcher, Mihai Pop, Steve Mount, and Bernd Sturmfels.

I would not be in graduate school if it were not for the influence and efforts of two incredible mentors: Herbert Wood and Jim Reggia. Dr. Wood got me excited about being a scientist, and Dr. Reggia made sure I was given the chance become one.

My earliest teachers of course are my parents and grandparents. My mother Siobhan Moose, my stepfather Ernest Moose, my stepmother Victoria Trapnell, and my father Bruce Trapnell have supported me throughout my education in many ways, and I would not have completed this work without that support. I am especially grateful to my mother and Ernie for giving me a place to live while working on my PhD. It's hard to overestimate the impact of fresh laundry and hot meals cooked by mom on one's scientific output. My uncle Michael Fritz was my first and best programming instructor - by insisting on lean, fast code at all times, and by giving me truly tough tasks in our effort to start a software company, he prepared me to work on the software in this thesis.

My grandfather, Donal Sullivan, set an example that when engineering something, there is such a thing as the The Right Way to Do It. My father shares that belief, and would have made almost as good an engineer as he is a scientist. The

numerous science projects I worked on with both of them as a child explain perhaps more than anything else why I have pursued a PhD. Woodshop projects, model rockets, plots of coin flip distributions, dry ice, agar plates, and balsa glider construction feature in my most vivid childhood memories. I have executed this work according to their principles to the best of my ability.

Above all others, I am grateful to my wife, Bianca Viray.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Algorithms and statistics for sequencing assays

Often, the phrase "DNA sequencing" conjures images of the race to determine the nucleotide sequence of the human genome, the storage medium for our genetic information. Within the last 15 years, the genomes of hundreds of organisms, ranging in size from bacteria to large mammals have been sequenced, most using fully-automated DNA sequencers. The complete sequence of the human genome is expected to greatly assist in the effort to understand our molecular biology, evolutionary history, and physiological diversity. However, the task of unraveling and understanding the enormously complex biological program stored in our DNA has barely begun.

Around the time scientists were starting to sequence whole genomes, others started using the sequencing technology to discover genes and determine the conditions in which they are expressed. Adams *et al* used sequencing technology to determine the sequence of the signature of an expressed gene[1]. *Expressed sequence tag* (EST) experiments uncovered first hundreds and then thousands of genes, many before the human genome was fully sequenced. As sequencing technology became less expensive and more reliable, researchers have continued the tradition of using sequencing technology to take measurements of the molecular activities of the cell.

Coupling bench techniques such as polymerase chain reaction (PCR), reverse transcription of RNA, and chromatin immunoprecipitation (to cite just a few of many examples) with high-throughput sequencing has revealed not only new protein coding genes, but non-coding RNA, DNA-protein interactions, regulatory sites in the genome, and other features central to our biology. Many of these assays go beyond discovery to measure the abundance of RNAs in a tissue sample or cell or the strength of an interaction, allowing scientists to design more powerful and sensitive experiments.

Until recently, assays such as EST sequencing produced modest amounts of raw sequence data compared to whole genome sequencing projects. Recently however, advances in reversible-terminator chemistry, optics, and robotics have enabled commercial sequencing technologies that produce a staggering amount of data from each experiment. For example, using a machine from Illumina, we describe here an experiment that produced over 30 gigabases of sequencing reads, or roughly the number of nucleotides as were stored in the entire GenBank database, a repository of all publicly available sequences, as of 2003. We expect this experiment to be considered relatively small-scale within a few years, as the throughput and quality of sequencers is rapidly improving. With this amount of data comes not only the great challenge of simply storing, analyzing, and summarizing it all, but exploiting the accompanied improvements in sensitivity and resolution to gain new biological insights. This thesis describes algorithms to meet the computational and statistical challenges associated with recent ultra high-throughput sequencing assays.

This chapter first highlights fundamental principles of sequencing assay design,

and makes clear why certain computational tasks, such as short read alignment, are at the core of most sequencing assay analyses. Next, two of the first (and still most common) sequencing assays are discussed. We used both high-throughput transcriptome sequencing[46, 47, 9] ("RNA-Seq"), and chromatin-immunoprecipitation sequencing [29, 55, 44] ("ChIP-Seq") to investigate gene expression dynamics in developing embryonic muscle cells. The promise of RNA-Seq, which is to provide a precise measurement of the abundance of every RNA in the transcriptome of a tissue or cell, has not yet been realized, primarily due to computational challenges.

RNA-Seq, ChIP-Seq, and many other recent assays measure the state of a cell or tissue by examining the distribution of alignments of sequenced fragments across a population of reference sequences. In RNA-Seq, genes covered by aligned fragments are inferred to be undergoing active transcription. For ChIP-Seq, "peaks" of piled-up reads reveal locations where proteins are binding to DNA - a critical piece of information for the study of gene regulation. We describe several algorithms to compute these alignments in Chapter 2, beginning with a hardware-accelerated approach that explored the use of commodity graphics processing units (GPUs) in short read alignment. While GPUs yielded a several-fold speedup over a CPU implementation of a classic sequence alignment algorithm, the sheer volume of short read data produced from a single experiment called for faster algorithms. Through the use of Burrows-Wheeler indexing, Langmead *et al* achieved a dramatically faster short read alignment algorithm called *Bowtie*[31], which we briefly review. We built the RNA-Seq read alignment program *TopHat*[64] around Bowtie, allowing reads to be aligned to the transcriptome in the absence of gene annotations and enabling the

discovery of novel splicing events.

In Chapter 3 we turn to the estimation of the abundances of a set of transcripts in a given sample. We briefly review the method of Jiang and Wong[28] for estimating isoform abundances with short (36bp) single-read RNA-Seq before describing our model, which accommodates arbitrarily long paired-end reads. The model is linear, which means that its likelihood function has a unique maximum and can be found with numerical means. However, this model is still subject to limitations described by Jiang and Wong of their model, so we adopt their importance sampling techniques to make reliable abundance estimates even near the boundaries of the model's parameter space. This importance sampling procedure allows us to estimate a variance-covariance matrix, which we will use to provide confidence intervals and integrate in statistical tests described in Chapter 5.

In Chapter 4, we address the problem of assembling full length transcript sequences from the alignment of RNA-Seq reads. The *Cufflinks* assembler produces a minimal set of transcript sequences necessary to explain the alignments. The assembler implements a constructive proof of a classic theorem in combinatorics. Given a directed acyclic graph (DAG), Dilworth's Theorem states that a minimum cover of the vertices of the DAG by paths has cardinality equal to the largest subset of vertices with the property that none can be reached from any other[14]. Reducing the problem of finding the cardinality of the cover to finding a maximum matching in a bipartite graph produces the cover itself. Cufflinks stores RNA-Seq fragment alignments in a directed acyclic graph, finds the minimum path cover via maximum matching, and converts cover elements into transcripts, thus producing a parsimo-

nious assembly that explains all of the fragment alignments.

A principle aim of many RNA-Seq experiments is not just to quantify RNA abundance in a sample, but to identify transcripts that are significantly more or less abundant between a pair of samples. In Chapter 5, we develop a set of statistical tests for RNA-Seq experiments, and describe a novel approach to testing that reveals changes not just in expression, but in gene regulation. Using the information-theoretic *Jensen-Shannon* divergence, we describe tests for significance of changes in the relative abundance of transcripts that discriminates transcriptional and post-transcriptional effects (see sections 1.2 and 1.3).

Before proceeding to the algorithms and mathematics, some biological background is necessary.

1.2 Transcription of RNA

Proteins and functional nucleic acids are molecules that make up cells and participate in their biomolecular interactions. These molecules are synthesized in cells by a complex machinery composed itself of proteins and nucleic acids, and the rate of production, or *expression* determines the extent of their impact on the cell's activities. The information necessary to construct a protein or non-coding RNA is stored in a *gene*, which is a subsequence of DNA in the genome. The synthesis of these molecules begins with the direct copying or *transcription* of the gene sequence, into an precursor RNA. This precursor RNA (pre-mRNA) is further processed in subsequent steps that will determine the functional properties of the molecule, as

described in the next section.

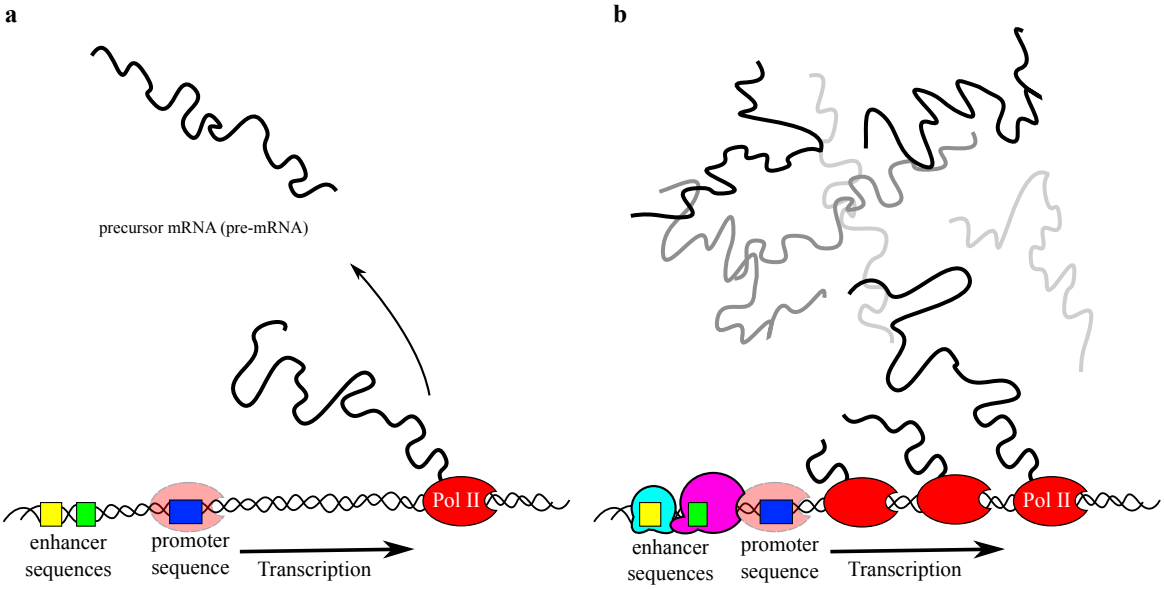


Figure 1.1: Synthesis of RNA, or *transcription*, is catalyzed by RNA polymerase. Messenger RNA, which forms templates for translation of genes into protein, is synthesized by RNA polymerase II (Pol II). (a) Pol II binds to a recognition sequence, or promoter, upstream of a gene, and then proceeds along the template strand of the DNA, producing a precursor mRNA molecule. (b) Proteins called *transcription factors* can bind to PolII or to proteins bound to PolII, as well as to other recognition sequences called *enhancers* near or within the gene to increase the rate of transcription. By stabilizing PolII and increasing its promoter-binding efficiency, or by making the locus more accessible to the transcriptional machinery, transcription factors may specifically target a gene for an increase in mRNA synthesis and thus protein production.

Transcription is a chemical reaction catalyzed by RNA polymerase, an enzyme that adds nucleotides ('A', 'C', 'G', and 'T/U') to a growing chain, forming the pre-mRNA. RNA polymerase binds to the double stranded DNA at one side of the gene (the "5'-end") and proceeds along it, simultaneously separating the strands and copying one of them into a growing pre-mRNA molecule. RNA polymerase binds to a specific site "upstream" of the gene, called the *promoter*, which contains

a short string of nucleotides that are chemically recognized by the enzyme. Once it is securely bound, the reaction begins and it starts to move toward the "3'-end" of the gene. For secure binding to occur, other proteins called *transcription factors* must also bind near the promoter to help stabilize RNA polymerase and help initiate transcription. Where nearly all genes are transcribed by a subtype of polymerase called RNA Polymerase II (polII) and several other generic transcription factors, most genes also require one or more transcription factors that bind only to the promoters of a subset of an organism's genes. These factors are *specific* for their target genes, and the targets will only be expressed if some or all of their specific transcription factors are also expressed.

Exactly what makes particular transcription factor specific for their targets is an area of intense research, but there is general consensus that:

1. Organisms have many transcription factors

2. Transcription factors help initiate, amplify, dampen, or entirely inhibit the transcription of their targets by either binding directly to DNA or interacting with other proteins that are bound to DNA in a complex.

3. A target gene may be transcribed starting at several different sites, with different promoters, giving rise to multiple pre-mRNAs.

4. A target gene may have multiple specific transcription factors whose interactions may determine its expression in a complex way.

These principles imply that a complex network of regulatory relationships

exists among the genes of an organism, and *transcriptional regulation* is believed to be a central strategy that has evolved to direct the cell's activities and determine its function within the organism. In other words, a liver cell differs from a muscle cell in large part due to differences in how these cells' genes (which are identical) are regulated. Cataloging the transcription factors, identifying their targets, and determining how they regulate the expression of these targets is believed to be central for a complete understanding of the molecular biology of our cells.

1.3 Alternative splicing

Pre-mRNA is often modified before being exported to the cytoplasm and used to synthesize proteins. There are several types of post-transcriptional pre-mRNA modifications, but we consider here only *splicing*, one of the most common and well understood. Splicing is the removal of subsequences of a pre-mRNA, followed by the joining of the remainder into a contiguous piece of RNA. There is mounting evidence that a majority of pre-mRNAs can be spliced in more than one way. *Alternative splicing* results from the differential use of *splice sites*, the positions at which the pre-mRNA is cut and ligated. Because alternative splicing can alter the nucleotide sequence of the final mRNA molecule, multiple proteins can be synthesized from a single pre-mRNA. Further, some splice variants are aggressively targeted by the cell's RNA degradation machinery before they are translated into proteins. The cell can thus use alternative splicing not just as a means to select which proteins are produced from an actively transcribed gene, but also how much total protein may

be produced from each pre-mRNA.

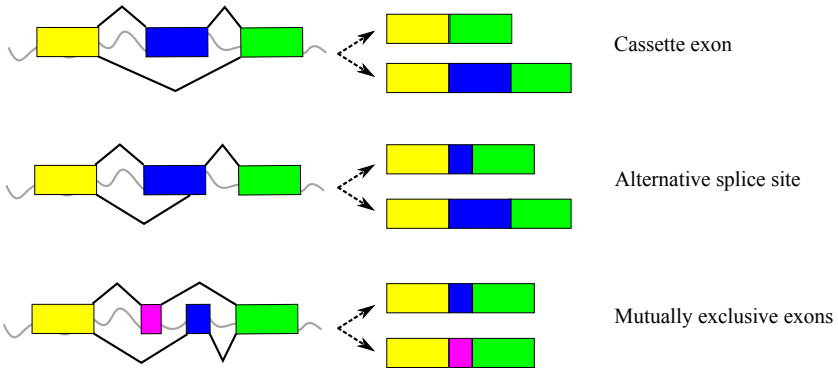


Figure 1.2: Before pre-mRNA are translated into protein, they are processed and exported from the nucleus to the cytoplasm. One type of post-transcriptional processing is called *splicing*, where sections of the mRNA (introns) are removed, and the remaining subsequences (exons) are concatenated together into the mature mRNA. A single pre-mRNA can be processed in more than one way, giving rise to multiple mRNAs. Three common types of alternative splicing events are shown above.

1.4 Biological inferences through fragment sequencing

Sequencing assays observe the state of cells and tissues and measure their activity at the molecular level with the following (very general) workflow:

1. Pick a cellular state or process that one wishes to observe.

2. Construct or capture a pool of nucleic acid sequences which, if known, would be informative about the sample.

3. Sequence that pool, which often requires fragmentation.

4. Reconstruct the sequences present in the pool (if they were fragmented) and

estimate their relative or absolute abundances through computational analysis of the sequencing reads, reference sequences, and other experimental data.

5. From the pool sequences and their abundances, make biological inferences, ideally in a statistically rigorous and principled way.

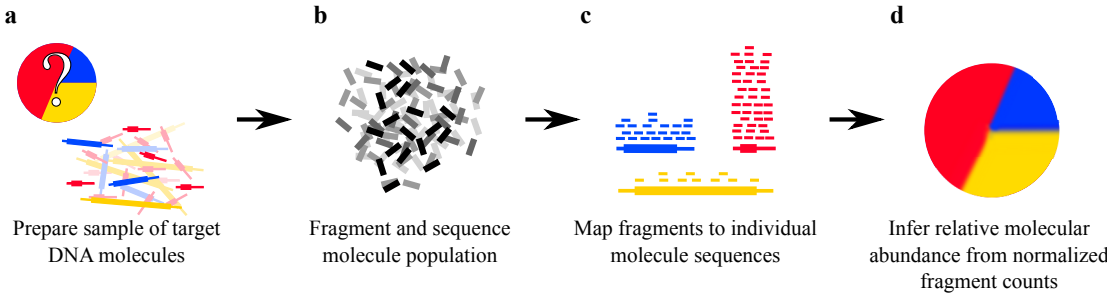


Figure 1.3: Quantitative sequencing assays aim to identify the sequences present in a sample along with their relative abundances. (a) A sample of DNA molecules is first prepared in which the relative abundance of each sequence (illustrated as a pie chart) is unknown. (b) This sample is sequenced, producing a set of randomly selected substrings of each molecular sequence. The number of fragments produced from each sequence in the sample is a function of that sequence's abundance (and other properties for some assays). (c) These fragments are mapped back to the sample sequences from which they originated. (d) The sample sequence abundances are inferred using the fragment mapping and a statistical model of how the fragment abundances relate to the sample sequences. Because there may be uncertainty in the mapping of fragments to sample sequences, and because the fragments are generated by the stochastic process of DNA sequencing, there may be uncertainty in the inferred abundances for the sample molecules (shown as a pie chart with "fuzzy" edges).

Depending on the assay, the computational analysis needed to make reliable inferences ranges from trivial to enormously challenging. Some types of small RNA sequencing require essentially no reconstruction of the underlying pool, because

each molecule is short enough to be entirely sequenced with a single read - and abundance estimation can reduce to counting sequences. Other assays, such as surveying complex structural variants with genomic resequencing, can in some cases amount to *de novo* whole genome shotgun assembly, a notoriously hard problem. Further, estimating the abundances of sequences in the assayed pool may be difficult, because the dynamic range of abundances may be many orders of magnitude (as in transcriptome sequencing), and can be complicated by issues of sampling, sequencing bias, and difficulties in protocol modeling.

Analysis strategies for many protocols have relied on aligning fragments to an available reference genome as a means of assembling the pool sequences and identifying which pool sequence each fragment came from. Aligning the sequenced fragments projects the pool sequences into the genomic coordinate space, which can make the task of inferring the underlying pool sequences either trivial or much easier. Often, finding the locations in the genome to which fragments align is a major goal of the assay (see section 1.6). Assembling pool sequences *de novo* with a short read assembler such as Velvet can produce a highly fragmented set of sequences (due to repeats in the pool) and often demands large amounts of memory. Aligning fragments to the genome reduces the computational load by eliminating the need for a general sequence overlap graph or other data structure used in assembly, and also can minimize the problems caused by repeats in the pool sequences.

Many sequencing assays are quantitative - the number of fragments originating from a given pool sequence are proportional its abundance in the pool. Depending on the assay protocol, the number of fragments generated from each pool sequence

may also depend on other properties such as the sequence's length. By calculating the number of alignments mapped to each pool sequence (and normalizing for its length), and dividing by the total number of fragments mapped in the assay, one can estimate pool sequence abundances. While some protocols introduce sampling bias that makes the correlation between fragment abundance and the abundance of length-normalized pool sequences less than perfect, many quantitative sequencing assays (e.g. RNA-Seq and ChIP-Seq, summarized below) have been demonstrated to be far more accurate and sensitive than previous technologies.

The favored units for reporting sample sequence abundances in many quantitative assays to date is not using the abundances directly, but rather using a measure abbreviated as FPKM, which means "expected number of fragments per kilobase of sample sequence per million fragments mapped". These units are equivalent to measuring sample sequence abundances (multiplied by a scalar). The computational advantage of FPKM, is that the normalization constants conveniently simplify some of the formulas for the variances of abundance estimates.

1.5 RNA-Seq

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)[1] or full-length complementary DNA (cDNA) sequences using conventional Sanger sequencing technology. Recently a new experimental method, RNA-Seq, has emerged that has a number of advantages over conventional EST

sequencing: by direct, high-throughput sequencing of a tissue or single-cell transcriptome, it avoids the need for bacterial cloning of cDNA and it generates data that can be used as a measure of the level of gene expression. Thus RNA-Seq experiments not only discover novel transcripts, they can replace conventional microarray experiments for measuring expression. Compared to microarray technology, RNA-Seq experiments provide much higher-resolution measurements of expression at comparable cost and reproducibility[43].

The major drawback of RNA-Seq over conventional EST sequencing is that the sequences themselves are much shorter. When first described, RNA-Seq produced millions of 25-36bp reads from each experiment. Recent improvements to the protocol and sequencing technology extend reads to 75-125bp, but remain short relative to the reads produced with Sanger sequencers. There are several variants of RNA-Seq, but we limit our discussion to the now widely-adopted described in Mortazavi *et al* and its extensions. RNA is first isolated from the sample, and is generally enriched for polyadenylated transcripts. Because single-stranded molecules may be self-complementary, they can fold into secondary structures, which may interfere with subsequent steps of the protocol. Thus, single stranded RNA is fragmented by a chemical reaction or by physical means (e.g. sonication). From these fragments, a randomly-primed cDNA library is built, size selected using an agarose gel, and loaded on to a sequencer such as a the Illumina Genome Analyzer. Each fragment, which in most protocols is 100-300bp long, is sequenced from one or both ends, producing a 25-125bp read from each end. The number of fragments produced by a transcript is proportial to its relative abundance in the transcriptome, after dividing

by it's length. That is, the longer of two equally abundant transcripts will produce more fragments.

Because RNA-Seq experiments generate fragments in proportion to the underlying abundance of transcripts, the first application of the assay was the estimation of gene expression. Directly measuring the relative abundance of all proteins in a sample is not currently feasible. However, measuring the relative abundance of the mRNAs giving rise to each protein is believed to be a good proxy for protein abundance. In RNA-Seq, "gene expression" refers to the fraction of the transcriptome occupied by the transcripts for each gene. A naïve, yet popular, current approach to expression estimation is to sum the fragments mapping to a gene (where the sum is taken across all exons appearing in all possible isoforms), and then to normalize the count by either the total number of exonic bases, or by the average length of the transcripts. We call the former method the "projective normalization" method, and the latter the "average length" method.

Proposition 1. *If a gene has two or more isoforms in the sample the expression of that gene is underestimated by projective normalization.*

Proof: Suppose gene g has k isoforms, and from isoform i of length l_i, the sequencing experiment produced f_i fragments. Then if the fraction of the transcriptome occupied by isoform i is $\rho_i = \frac{f_i}{l_i}$, the fraction of the transcriptome occupied by g is

$$\rho_g = \sum_{i=0}^{k} \frac{f_i}{l_i} \tag{1.1}$$

Let the length of the projective normalization of g be denoted $l_{P(g)}$. Note that

for each isoform i, $l_i \leq l_{P(g)}$. Thus, the projectively normalized expression of g, computed by

$$\rho_{g_P} = \sum_{i=0}^{k} \frac{f_i}{l_{P(g)}} \tag{1.2}$$

is always less than the true gene expression ρ_g. □

Stated differently, the projective normalization method has the problem that it produces numbers that are not proportional to the abundances of the gene when the sample contains multiple isoforms for that gene. Further, expression values computed in this way are not additive, severely limiting the use of RNA-Seq in systems biology analyses and other settings. The average length method is flawed for the same reason. In some cases the method might produce the correct answer (for the wrong reasons), but it is bound to be incorrect on many examples, especially in genes with transcripts of variable lengths and non-uniform abundances. In RNA-Seq, the expression of a gene should simply be the sum of its individual transcript values. Even if one is interested only in the expression of whole genes rather than individual transcripts, the abundances for those transcripts must be computed. However, computing those values is computationally difficult.

The central computational challenge of analyzing RNA-Seq experiments lies in assigning fragments to transcripts. However, because the transcriptomes are incomplete even for well-studied species such as human and mouse, it is generally necessary to discover or assemble transcript sequences before assigning reads. This can be done in two steps: (1) aligning fragments to the genome as a proxy for aligning them to the transcriptome and (2) inferring full-length transcript sequences from

these alignments. In higher eukaryotes such as vertebrates, alternative splicing is common, which adds a further layer of complexity to the analysis. Even with a complete transcriptome, reads may not be uniquely assigned to a single alternative splice variant of a gene because that gene's isoforms share many exons.

1.6 ChIP-Seq

ChIP-Seq is a quantitative sequencing assay that aims to identify sites where a specific transcription factor is binding to genomic DNA and quantify the strength of binding activity at each site. ChIP-Seq can also be used to identify the locations and modification states of *histones*, protein complexes around which genomic DNA is wrapped, and which are believed to greatly influence the transcription and possibly even splicing of genes. ChIP-Seq begins with a "cross-linking" step in which proteins bound to DNA are treated with formaldehyde, fixing the proteins to their binding sites with strong chemical bonds. The DNA is then sheared by sonication or chemically via nuclease, resulting in DNA fragments typically 200-1000bp long. Fragments with bound proteins are then enriched by adding an antibody that binds specifically to the protein of interest. The cross-links between the protein of interest and the bound fragments are then reversed. As a control to establish the rate of background (i.e. fragments not bound to protein) precipitation, the sample is also immunoenriched with a non-specific antibody.

The target and control IP fragments are then size selected via methods similar to for RNA-Seq, and a pair of sequencing libraries are built. These libraries

are sequenced, and the reads are aligned and mapped to the genome. Locations where reads from the target library represent possible binding sites for the target protein. The more reads that pile up in a given spot, the more immunoenriched fragments originated from that location in the genome, and thus the greater the binding strength. However, antibodies for different proteins have different affinities, which means that they are not equally effective at pulling down bound fragments. Moreover, even IPs performed with high-affinity antibodies will also pull down some unbound fragments. The depth of coverage of the genome by the control sample is thus crucial for establishing what constitutes a genuine binding site.

1.7 A case study: sequencing the myogenic transcriptome

Expression analysis is a central technique for identifying important genes in many biological settings, but it is particularly common in developmental studies. In the development of the fruit fly embyro, concentration patterns and gradients of several master regulatory proteins are responsible for establishing the "patterning" of the organism. The development of the major anatomic structures of the adult fly, such as the head, thorax, wings, and abdomen is specified by the differential expression of genes at different positions within the embryo[33]. A similar, but less well understood system of coordinated differential expression also drives vertebrate development[20]. While vertebrates are anatomically diverse and diverge in their developmental programs, the development of some parts of the vertebrate body are believed to be driven by essentially the same program of gene expression. Striated

muscle, which includes skeletal and cardiac muscle in vertebrates and arthropods, is believed to have evolved more than 700 million years ago, before vertebrates and arthropods diverged from a common ancestor[49]. Both *in vivo* gene expression studies of and *in vitro* models have revealed much about the regulation of muscle development, or *myogenesis*. Beyond understanding the evolution of muscle, unraveling the dynamics of protein, DNA, and RNA interactions that relate its development would greatly increase our understanding of wound healing along with the pathology of a wide array of human muscle diseases.

As a demonstration of the computational methods described in this thesis, we performed a timecourse of paired-end 75bp RNA-Seq on a well-studied model of skeletal muscle development, the C2C12 mouse myoblast cell line. Regulated RNA expression of key transcription factors drives myogenesis and the execution of the differentiation process involves changes in expression of hundreds of genes[71, 62]. Prior studies have not measured global transcript isoform expression, though there are well-documented expression changes at the whole gene level for a set of marker genes in this system. We aimed to establish the prevalence of differential promoter use and differential splicing, because such data could reveal much about the models regulatory behavior. A gene with isoforms that code for the same protein may be subject to complex regulation in order to maintain a certain level of output in the face of changes in expression of its transcription factors. Alternatively, genes with isoforms that code for different proteins could be functionally specialized for different cell types or states. By analyzing changes in relative abundances of transcripts produced by the alternative splicing of a single primary transcript, we hoped to

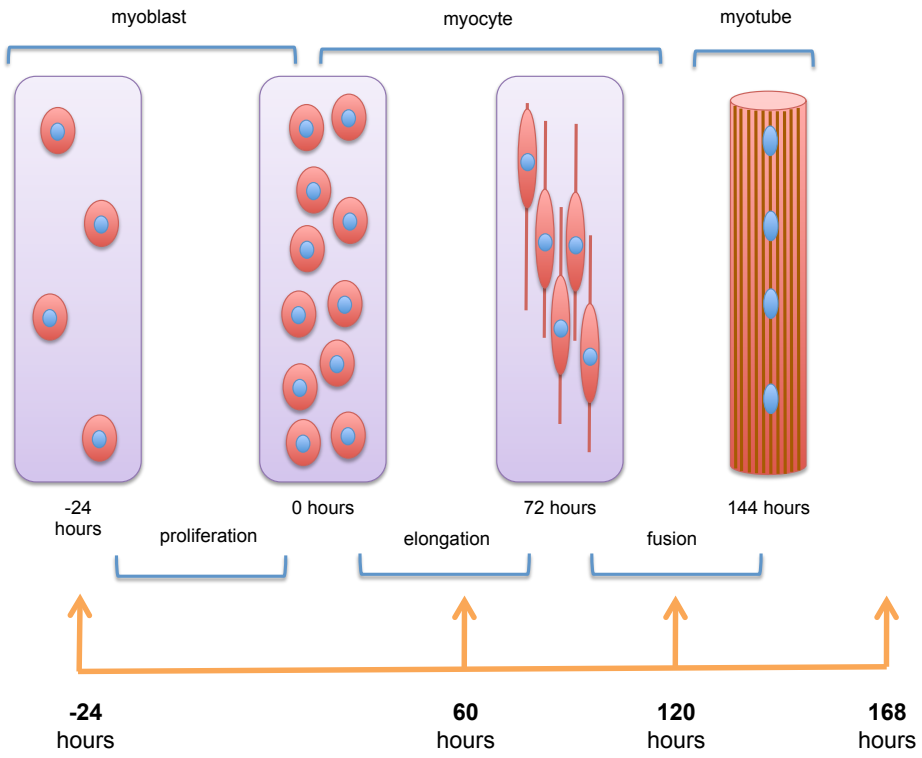
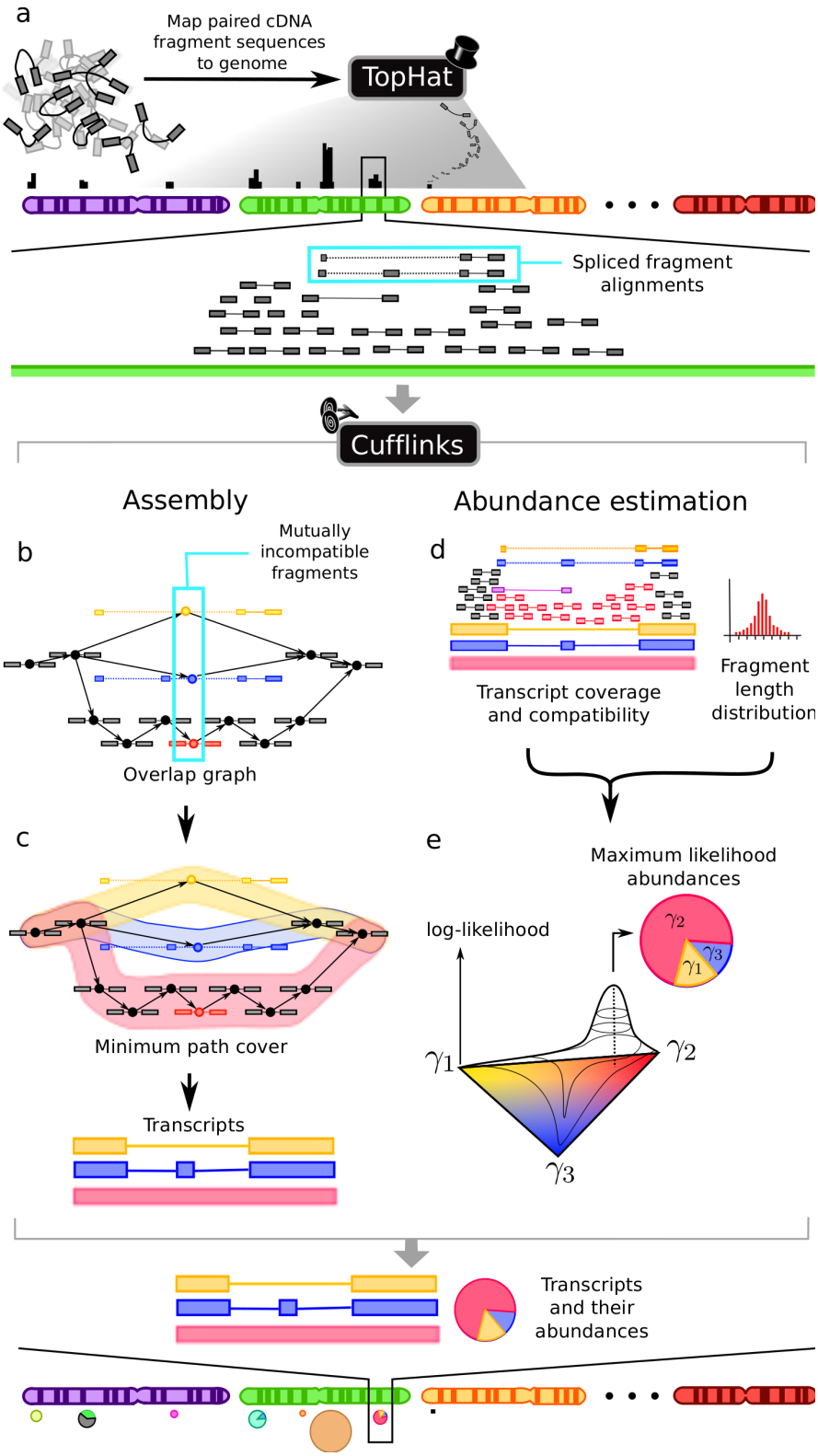


Figure 1.4: RNA-Seq samples taken at strategic time points in C2C12 development. Illustration after Ohtake *et al*[48]

Figure 1.5 ***(following page)*:** Overview of Cufflinks. The algorithm takes as input cDNA fragment sequences that have been (a) aligned to the genome by software capable of producing spliced alignments, such as TopHat. With paired-end RNA-Seq, Cufflinks treats each pair of fragment reads as a single alignment. The algorithm assembles overlapping bundles of fragment alignments (b-c) separately, which reduces running time and memory use because each bundle typically contains the fragments from no more than a few genes. Cufflinks then estimates the abundances of the assembled transcripts (d-e).


a
Map paired cDNA fragment sequences to genome
TopHat
Spliced fragment alignments
Cufflinks
Assembly
Abundance estimation
b
Mutually incompatible fragments
Overlap graph
d
Transcript coverage and compatibility
Fragment length distribution
c
Minimum path cover
Transcripts
e
Maximum likelihood abundances
log-likelihood
γ_1
γ_2
γ_3
Transcripts and their abundances

infer the impact of post-transcriptional processing (e.g. splicing) on RNA output separately from rates of primary transcription. Such analysis could identify genes with a role in the system and suggest experiments to establish precisely how they are regulated.

Total RNA was extracted from developing C2C12 cells, and subsequently mRNA was isolated at four different time points (-24 hours, 60 hours, 120 hours, 168 hours). cDNA was prepared following a similar procedure to the one described in[46], with modifictions described in Appendix C. Fragmentation of the mRNA followed by size selection resulted in fragment lengths 200nt long for all of the time-points. The timepoint sequences totaled 430,467,018 paired 75bp reads sequenced from the transcriptome of mouse skeletal muscle C2C12 cells induced to undergo myogenic differentiation.

We first mapped these fragments to the mouse genome using TopHat (see section 2.4. We then used Cufflinks to assemble the alignments into transcripts and estimate their abundances (see Chapters 4 and 3). Figure 1.5. gives an overview of Cufflinks. After the fragments have been mapped with TopHat (a), Cufflinks assembles the transcripts from the alignments. (b) The first step in fragment assembly is to identify pairs of incompatible fragments that must have originated from distinct spliced mRNA isoforms. Fragments are connected in an overlap graph when they are compatible and their alignments overlap in the genome. Each fragment has one node in the graph, and an edge, directed from left to right along the genome, is placed between each pair of compatible fragments. In this example, the yellow, blue, and red fragments must have originated from separate isoforms, but any other

fragment could have come from the same transcript as one of these three. (c) Assembling isoforms from the overlap graph. Paths through the graph correspond to sets of mutually compatible fragments that could be merged into complete isoforms. The overlap graph here can be minimally covered by three paths, each representing a different isoform. Dilworths Theorem states that the number of mutually incompatible reads is the same as the minimum number of transcripts needed to explain all the fragments. Cufflinks implements a proof of Dilworths Theorem that produces a minimal set of paths that cover all the fragments in the overlap graph by finding the largest set of reads with the property that no two could have originated from the same isoform. (d) Estimating transcript abundance. Fragments are matched (denoted here using color) to the transcripts from which they could have originated. The violet fragment could have originated from the blue or red isoform. Gray fragments could have come from any of the three shown. Cufflinks estimates transcript abundances using a statistical model in which the probability of observing each fragment is a linear function of the abundances of the transcripts from which it could have originated. Because only the ends of each fragment are sequenced, the length of each may be unknown. Assigning a fragment to different isoforms often implies a different length for it. Cufflinks can incorporate the distribution of fragment lengths to help assign fragments to isoforms. For example, the violet fragment would be much longer, and very improbable according to Cufflinks model, if it were to come from the red isoform instead of the blue isoform. (e) The program then numerically maximizes a function that assigns a likelihood to all possible sets of relative abundances of the yellow, red and blue isoforms $(\gamma_1, \gamma_2, \gamma_3)$, producing the abundances

that best explain the observed fragments, shown as a pie chart.

As discussed in Chapter 4 , we analyzed the transcripts at each point during C2C12 development by comparing them to databases of known mouse RNAs and also by performed wet validation experiments. Because we aimed to identify promoter switching and dynamics, we performed ChIP-Seq experiments targeting RNA Polymerase II and TAF1, a general transcription factor that marks active promoters. Determining the fraction of transcripts that had a polII or TAF1 peak immediately upstream of the 5' ends allowed us to validate novel transcription start sites (and thus, novel promoters) using independent experimental means.

We then analyzed the expression dynamics of the myogenic transcriptome using the statistical model detailed in Chapter 3. The mathematical background needed and the simulation experiments we performed to validate the model are also discussed. Cufflinks also includes software that performs statistical significance testing for changes between pairs of RNA-Seq samples. We discuss these tests in Chapter 5, along with the results of testing conducted on the myogenic transcriptome. We tracked changes in more than 10,000 genes, and uncovered not only widespread expression changes at the level of individual transcripts, but also changes in the transcriptional and post-transcriptional regulation of hundreds of genes.

Chapter 2

Short read alignment

2.1 Overview

Current sequencing assays measure molecular biological activity by sequencing nucleic acid fragments and mapping them to reference molecules (e.g. a reference genome). The positions of these alignments, their density in certain loci, and the differences (mismatches, insertions, and deletions or 'indels') between the fragment sequences and the reference are all informative. Thus, computing the alignments between sequenced fragments and a set of potentially very long reference sequences is a core computational step in a sequencing assay. Current sequencing machines from Illumina, Life Technologies, and Helicos produce tens to hundreds of millions of sequencing reads per run. Each run takes a few days to over one week to complete, and multiple assays can be prepped and processed in a single machine run. Thus, a single lab could produce billions of basepairs of raw sequencing data in a short time.

The individual sequencing reads from the above technologies are short - typically a string 25bp to 125bp long. The reference sequences to which these reads

This chapter discusses three programs: MUMmerGPU, Bowtie, and TopHat. MUMmerGPU[58, 65] is joint work with Michael Schatz, Arthur Delcher, and Amitabh Varshney. Bowtie[31] is joint work with Ben Langmead, Mihai Pop, and Steven Salzberg, and was primarily written by Ben. TopHat[64] is joint work with Lior Pachter and Steven Salzberg.

must be aligned are typically chromosomes from a single genome, which combined form a string billions of basepairs long. The task of aligning even one small read to a string the size of a genome is challenging, especially when mismatches and indels must be allowed. To extract meaningful biological insights from a sequencing assay, it is generally necessary to map all of the millions of reads, making assay analysis computationally very demanding. However, in many assays, the reads may be aligned independently, making short read alignment an "embarassingly parallel" problem. As data from the Illumina Genome Analyzer began to be made publicly available, we turned to another rapidly evolving technology - commodity graphics processing units (GPUs) - for a solution to the short read alignment problem.

2.2 Hardware-accelerated read mapping

Our early efforts to provide efficient tools for mapping short reads to large genomes resulted MUMmerGPU, an open-source high-throughput parallel pairwise local sequence alignment program that runs on commodity Graphics Processing Units (GPUs) in common workstations. MUMmerGPU uses the new Compute Unified Device Architecture (CUDA) from nVidia to align multiple query sequences against a single reference sequence stored as a suffix tree. The program is a adaptation of the popular MUMmer suffix-tree-based alignment program[13]. By processing the queries in parallel on the highly parallel graphics card, MUMmerGPU achieves more than a 10-fold speedup over a serial CPU version of the sequence alignment kernel, and outperforms the exact alignment component of MUMmer on a high end

CPU by 3.5-fold in total application time when aligning reads from recent sequencing projects using Solexa/Illumina, 454, and Sanger sequencing technologies.

Most personal computer workstations today contain hardware for 3D graphics acceleration called graphics processing units. Recently, GPUs have been harnessed for non-graphical, general purpose (GPGPU) applications. GPUs feature hardware optimized for simultaneously performing many independent floating-point arithmetic operations for displaying 3D models and other graphics tasks. Thus, GPGPU programming has been successful primarily in the scientific computing disciplines which involve a high level of numeric computation. However, other applications could be successful, provided those applications feature significant parallelism.

As the GPU has become increasingly more powerful and ubiquitous, researchers have begun exploring ways to tap its power for non-graphics, or general-purpose (GPGPU) applications[50]. This has proven challenging for a variety of reasons. Traditionally, GPUs have been highly specialized with two distinct classes of graphics stream processors: vertex processors, which compute geometric transformations on meshes, and fragment processors, which shade and illuminate the rasterized products of the vertex processors. The GPUs are organized in a streaming, data-parallel model in which the processors execute the same instructions on multiple data streams simultaneously. Modern GPUs include several (tens to hundreds) of each type of stream processor, so both graphical and GPGPU applications are faced with parallelization challenges[21]. Furthermore, on-chip caches for the processing units on GPUs are very small (often limited to what is needed for texture filtering operations) compared to general purpose processors, which feature caches measured

in megabytes. Thus, read and write operations can have very high latency relative to the same operations when performed by a CPU in main memory.

Most GPGPU successes stem from scientific computing or other areas with a homogeneous numerical computational component[24]. These applications are well suited for running on graphics hardware because they have high arithmetic intensity the ratio of time spent performing arithmetic to the time spent transferring data to and from memory[12]. In general, the applications that have performed well as a GPGPU application are those that can decompose their problems into highly independent components each having high arithmetic intensity. Some bioinformatics applications with these properties have been successfully ported to graphics hardware. Liu et al. implemented the Smith-Waterman local sequence alignment algorithm to run on the nVidia GeForce 6800 GTO and GeForce 7800 GTX, and reported an approximate 16 speedup by computing the alignment score of multiple cells simultaneously[40]. Charalambous et al. ported an expensive loop from RAxML, an application for phylogenetic tree construction, and achieved a 1.2 speedup on the nVidia GeForce 5700 LE[8].

nVidia's new G80 architecture radically departs from the traditional vertex+fragment processor pipeline. It features a set of multiprocessors that each contain a number of stream processors. Graphics applications can use these as either vertex or fragment processors, and GPGPU applications can program them for general computation. All processors on a single multiprocessor simultaneously execute the same instruction, but different multiprocessors can execute different instructions. nVidia anticipated the benefits of such a unified architecture for GPGPU

computing, and released the Compute Unified Device Architecture (CUDA) SDK to assist developers in creating non-graphics applications that run on the G80 and future GPUs. CUDA offers improved flexibility over previous GPGPU programming tools, and does not require application writers to recast operations in terms of geometric primitives, as was required by earlier GPGPU environments.

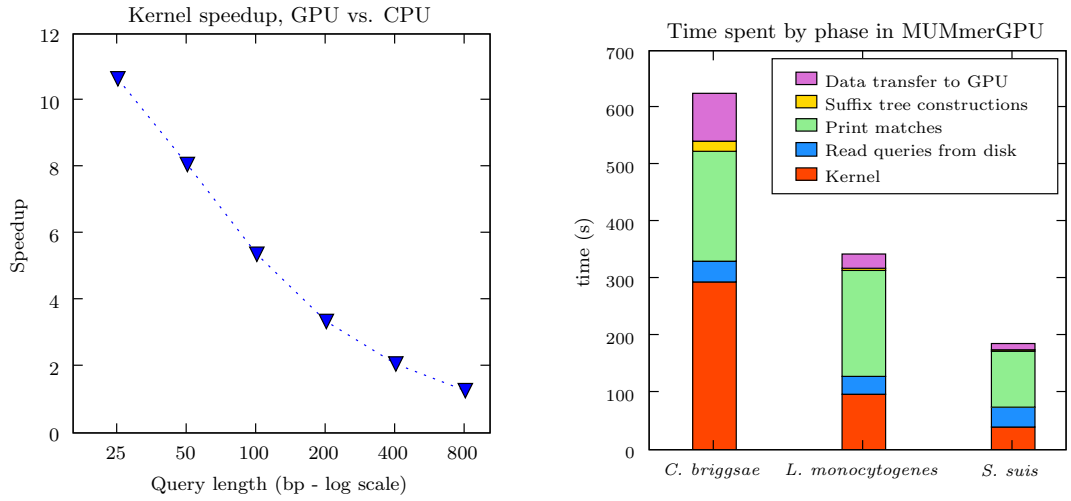
The MUMmerGPU algorithm performs parallelized exact string alignment on the GPU. First a suffix tree of the reference sequence is constructed on the CPU using Ukkonen's algorithm[67] and transfered to the GPU. Then the query sequences are transfered to the GPU, and are aligned to the tree on the GPU using the alignment algorithm described above. Alignment results are temporarily written to the GPU's memory, and then transfered in bulk to host RAM once the alignment kernel is complete for all queries. Finally, all maximal alignments longer than a user-supplied value are reported by post-processing the raw alignment results on the CPU.

Operations on the suffix tree have extremely low arithmetic intensity—they consist mostly of following a series of pointers. Thus, sequence alignment with a suffix tree might be expected to be a poor candidate for a parallel GPGPU application. However, our results show that a significant speedup, as much as a 10-fold speedup (Figure 1(a)), can be achieved through the use of cached texture memory and data reordering to improve access locality. Even though MUMmerGPU is a low arithmetic memory intensive program, and the stream processor cache on a typical GPUs is limited, MUMmerGPU achieved a significant speedup, in part, by reordering the nodes to match the access patterns and fully use the cache. We therefore

expect with careful analysis of the access pattern, essentially any highly parallel algorithm to perform extremely well on a relatively inexpensive GPU, and anticipate widespread use of GPGPU and other highly parallel multicore technologies in the near future.

An update to MUMmerGPU aimed to eliminate bottlenecks in the computation by accelerating output of results and reducing latency in the alignmnent kernel. MUMmerGPU 2.0 features a new stackless depth-first-search print kernel and is 13x faster than the serial CPU version of the alignment code and nearly 4x faster in total computation time than MUMmerGPU 1.0. We exhaustively examined 128 GPU data layout configurations to improve register footprint and running time and conclude higher occupancy has greater impact than reduced latency. MUMmerGPU 2.0 uses the same suffix tree based match kernel as described in the original version of MUMmerGPU, but we have added several significant improvements to increase performance and capabilities for the overall application. First, we implemented a new query streaming model in which reads are streamed past overlapping segments of the reference, allowing us to compute alignments to Mammalian-sized reference genomes. Second, we implemented a new GPU-based print-kernel that post-processes the results from the match kernel into alignments suitable for printing. This computation had previously been the limiting factor in end-to-end application time for commonly used parameters (Figure 1(b)).

The print kernel performs the computation via an iterative depth-first-search on the suffix tree using a constant amount of memory and no stack. This non-traditional implementation is required to meet the severe restrictions on kernel code,



(a) Speedup of MUMmerGPU on the GPU over the CPU.

(b) Breakdown of MUMmerGPU processing time.

Figure 2.1: (a) Speedup of MUMmerGPU on the GPU over the CPU. The decrease in speedup when processing error-free synthetic reads as read length increases is due to a combination of thread divergence and poor cache hit rate.(b) Breakdown of MUMmerGPU processing time. The stacked bar charts indicate the amount of time spent in each phase of the MUMmerGPU for the three test sets. Given a sufficiently large number of sequencing reads, the time spent building the suffix tree is small compared to time spent aligning queries.

but is between 1.5- and 4-fold faster than the previous (CPU-based) version of the routine. Popov et al recently reported a different algorithm for traversing trees in a CUDA kernel[53] which requires additional pointers between the leaf nodes in a kd-tree, but our technique is applicable to any tree without additional pointers. Finally, we optimized performance for both kernels by identifying the best organization of the DNA sequencing reads and suffix tree in GPU memory. We explored 128 variations of the data layout policy, and quantify the tradeoffs involved for kernel complexity, cache use, and data placement. We find that optimizing these choices can greatly accelerate performance, and mistuned choices have an equal but negative effect on performance compared to the nave version. For example, storing the suffix array as a one-dimensional array proved to be faster than storing it as a two-dimensional array, despite the fact that GPUs are typically cached with two-dimensional access locality in mind (Figure 2.2). Processor occupancy determined performance for our data-intensive application, but techniques that reduce GPU memory latency without compromising occupancy were also generally beneficial. We describe several techniques to reduce kernel register footprint and thus improve occupancy that are widely applicable to GPGPU programs. Overall, MUMmerGPU 2.0 is nearly 4x faster in total computation time than the originally published version of the code for the most commonly encountered workloads.

MUMmerGPU demonstrated that a surprisingly large speedup was possible for applications with essentially no arithmetically intense component. In absolute performance terms though, it has proved unable to align reads at the throughput required by recent sequencing experiments. In a recent effort to find driving muta-

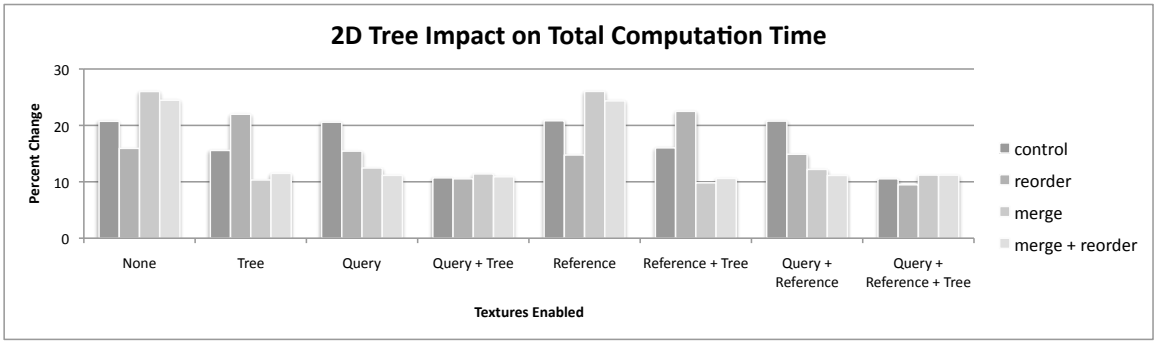


Figure 2.2: Performance impact of MUMmerGPU data layout policy. Storing the MUMmerGPU suffix array as a 1D array instead of a 2D texture accelerates MUMmerGPU.

tions in acute myeloid leukemia, Ley *et al* sequenced the tumor and normal genomes from a single individual with Illumina, producing 140 gigabases of raw sequencing reads[34]. To process this data with, Maq[36] or SOAP[38], two of the fastest available alignment programs at the time would have taken more than 5 months of CPU time[31].

2.3 Ultra-high throughput mapping with Burrows-Wheeler indexing

Maq and SOAP take the same basic algorithmic approach as other recent read mapping tools such as RMAP[59], ZOOM[39], and SHRiMP[56]. Each tool builds a hash table of short oligomers present in either the reads (SHRiMP, Maq, RMAP, and ZOOM) or the reference (SOAP). Some employ recent theoretical advances to align reads quickly without sacrificing sensitivity. For example, ZOOM uses 'spaced seeds' to significantly outperform RMAP, which is based on a simpler algorithm developed by Baeza-Yaetes and Perleberg[3]. Spaced seeds have been shown to yield higher sensitivity than contiguous seeds of the same length[6, 42]. SHRiMP employs

a combination of spaced seeds and the Smith-Waterman[60] algorithm to align reads with high sensitivity at the expense of speed. Eland is a commercial alignment program available from Illumina that uses a hash-based algorithm to align reads.

Bowtie uses a different and novel indexing strategy to create an ultrafast, memory-efficient short read aligner geared toward mammalian re-sequencing. In our experiments using reads from the 1,000 Genomes project, Bowtie aligns 35-base pair (bp) reads at a rate of more than 25 million reads per CPU-hour, which is more than 35 times faster than Maq and 300 times faster than SOAP under the same conditions (see Tables 1 and 2). Bowtie employs a Burrows-Wheeler index based on the full-text minute-space (FM) index, which has a memory footprint of only about 1.3 gigabytes (GB) for the human genome. The small footprint allows Bowtie to run on a typical desktop computer with 2 GB of RAM. The index is small enough to be distributed over the internet and to be stored on disk and re-used. Multiple processor cores can be used simultaneously to achieve even greater alignment speed. We used Bowtie to align 14.3 coverage worth of human Illumina reads from the 1,000 Genomes project in about 14 hours on a single desktop computer with four processor cores.[31]

Bowtie indexes the reference genome using a scheme based on the Burrows-Wheeler transform (BWT)[7] and the FM index[17, 18]. A Bowtie index for the human genome fits in 2.2 GB on disk and has a memory footprint of as little as 1.3 GB at alignment time, allowing it to be queried on a workstation with under 2 GB of RAM. The common method for searching in an FM index is the exact-matching algorithm of Ferragina and Manzini, illustrated in figure 2.3. Bowtie does not simply

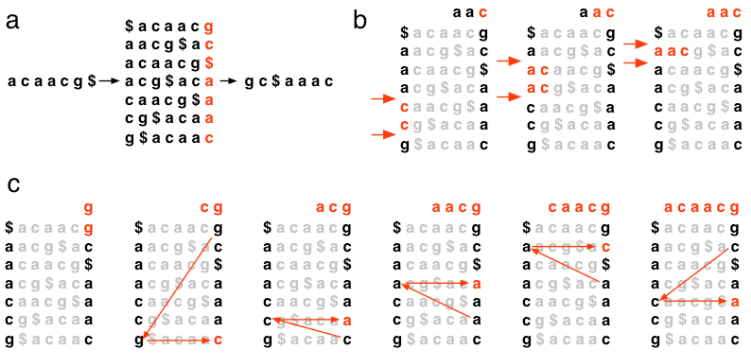


Figure 2.3: Exact string matching with a Burrows-Wheeler index

adopt this algorithm because exact matching does not allow for sequencing errors or genetic variations. We introduced two novel extensions that make the technique applicable to short read alignment: a quality-aware backtracking algorithm that allows mismatches and favors high-quality alignments; and 'double indexing', a strategy to avoid excessive backtracking. The Bowtie aligner follows a policy similar to Maq's, in that it allows a small number of mismatches within the high-quality end of each read, and it places an upper limit on the sum of the quality values at mismatched alignment positions.

2.4 TopHat: Alignment of RNA-Seq reads

MUMmerGPU and Bowtie both align reads to a reference genome, and Bowtie allows for mismatches in order to tolerate sequencing errors in the reads and discover single base differences between the donor and the reference. In principle, an algorithm that infers individual transcript abundances by measuring the fraction of fragments originating from each of a set of known transcripts would begin by com-

puting alignments between fragments and the set of known transcripts that may be contained in the sample using a tool like Bowtie. However, because the transcriptome for mouse is incompletely annotated, such an analysis requires mapping of fragments to the genome as a proxy for mapping directly to transcripts, so that new transcript structures can be discovered and so alignments will not be missed. This means that alignments of short sequencing reads must be allowed to span exon-exon splice junction in genomic coordinate space. We previously developed a program called TopHat to map RNA-Seq reads to the genome. TopHat does not require a reference transcriptome and can therefore be used to discover novel splice junctions.[64]

TopHat finds junctions by mapping reads to the reference in two phases. In the first phase, the pipeline maps all reads to the reference genome using Bowtie. All reads that don't map to the genome are set aside as "initially unmapped reads," or IUM reads. Bowtie reports, for each read, one or more alignments containing no more than a few mismatches (two, by default) in the 5'-most s bases of the read. The remaining portion of the read on the 3' end may have additional mismatches, provided that the Phred-quality-weighted Hamming distance is less than a specified threshold (70 by default). This policy is based on the empirical observation that the 5' end of a read contains fewer sequencing errors than the 3' end[26]. TopHat allows Bowtie to report more than one alignment for a read (default = 10), and suppresses all alignments for reads that have more than this number. This policy allows so called "multireads" from genes with multiple copies to be reported, but excludes alignments to low-complexity sequence, to which failed reads often align.

Low complexity reads are not included in the set of IUM reads; they are simply discarded.

When TopHat was first released, RNA-Seq experiments used a single-end sequencing protocol - cDNA fragments were sequenced only from one end. These reads were also no longer than 36bp. TopHat's algorithm for detecting novel splice junctions was designed specifically to work with these reads, but as the sequencing technology improved, TopHat evolved to exploit features of "second-generation" RNA-Seq. In the next section, the original, first-generation TopHat algorithms are described and evaluated. Improvements to TopHat made since its initial release are outlined briefly below.

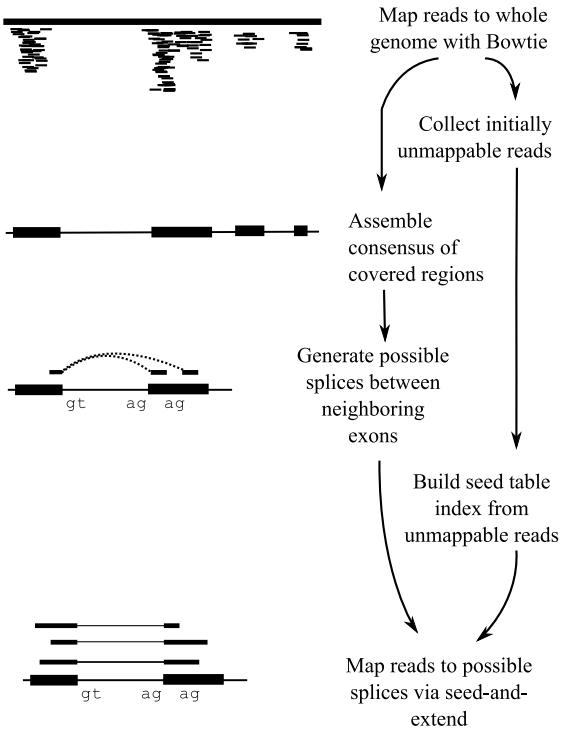


Figure 2.4: The TopHat pipeline for first-generation RNA-Seq.

2.4.1 Junction discovery with short, unpaired reads

Fragment alignments in the initial map are typically clustered together, where each cluster or "island" of coverage coincides with the core of an exon. To map reads to splice junction between exons, TopHat first enumerates all donor and acceptor dinucleotides (e.g. 'GT' and 'AG') within the or near each island to the genome. Next, it considers all pairings of these dinucleotides that could form canonical (GT-AG) introns between neighboring (but not necessarily adjacent) islands of map coverage. Each possible intron is checked against the IUM reads for reads that span the splice junction, as described below. By default, TopHat only examines potential introns longer than 70bp and shorter than 20000bp when working with first-generation reads, but these default minimum and maximum intron lengths can be adjusted by the user. These values describe the vast majority of known eukaryotic introns. For example, more than 93% of mouse introns in the UCSC known gene set fall within this range. However, users willing to make a small sacrifice in sensitivity will see substantially lower running time by reducing the maximum intron length.

To improve running times and avoid reporting false positives, the program excludes donor-acceptor pairs that fall entirely within a single island, unless the island is very deeply sequenced. An example of a "single island" junction is illustrated in Figure 2.5. The gene shown has two alternate transcripts, one of which has an intron that coincides with the UTR of the other transcript. The figure shows the normalized coverage of the intron and its flanking exons by uniquely-mappable reads as reported by Mortazavi *et al.* Both transcripts are clearly present in the RNA-

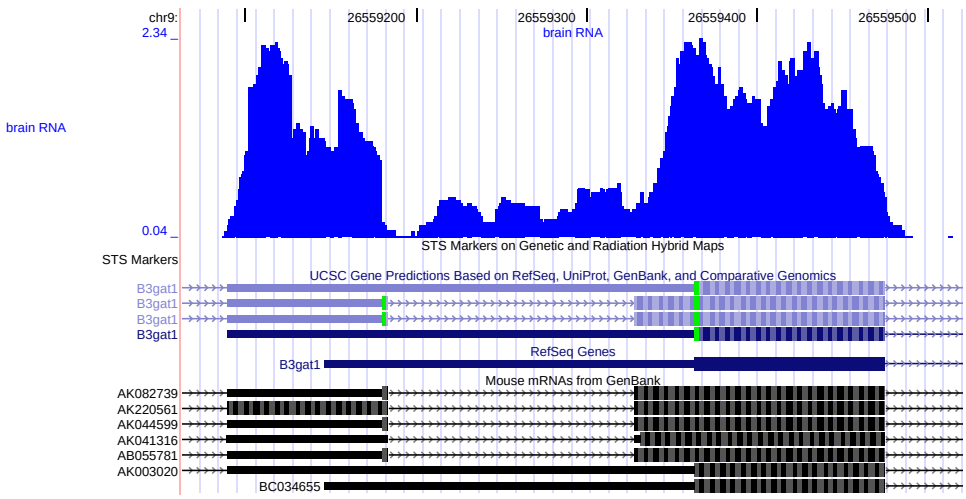


Figure 2.5: An intron entirely overlapped by the 5' UTR of another transcript. Both isoforms are present in the brain tissue RNA sample. The top track is the normalized uniquely-mappable read coverage reported by ERANGE for this region[46]. The lack of a large coverage gap causes TopHat to report a single island containing both exons. TopHat looks for introns within single islands in order to detect this junction.

Seq sample, and TopHat reports the entire region as a single island. In order to detect such junctions without sacrificing performance and specificity, TopHat looks for introns within islands that are deeply sequenced. During the island extraction phase of the pipeline, the algorithm computes the following statistic for each island spanning coordinates i to j in the map:

$$D_{ij} = \frac{\sum_{m=i}^{j} d_m}{j - i} \cdot \frac{1}{\sum_{m=0}^{n} d_m} \tag{2.1}$$

where d_m is the depth of coverage at coordinate m in the Bowtie map, and n is the length of the reference genome. When scaled to range [0,1000], this value represents the normalized depth of coverage for an island. We observed that single-

island junctions tend to fall within islands with high D (data not shown). TopHat thus looks for junctions contained in islands with $D \geq 300$, though this parameter can be changed by the user. A high D value will prevent TopHat from looking for junctions within single islands, which will improve running time. A low D value will force TopHat to look within many islands, slowing the pipeline, but potentially finding more junctions.

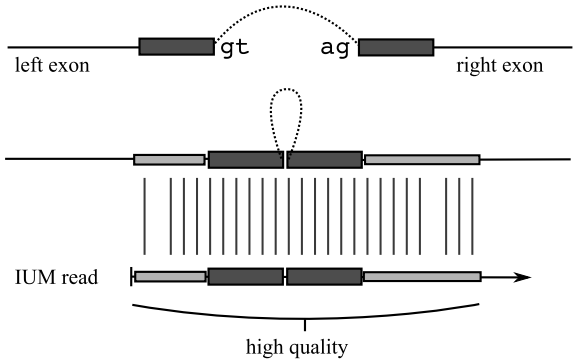


Figure 2.6: The seed and extend alignment used to match reads to possible splice sites. For each possible splice site, a seed is formed by combining a small amount of sequence upstream of the donor and downstream of the acceptor. This seed, shown in dark gray, is used to query the index of reads that were not initially mapped by Bowtie. Any read containing the seed is checked for a complete alignment to the exons on either side of the possible splice. In the light gray portion of the alignment, TopHat allows a user specified number of mismatches. Because reads typically contain low-quality base calls on their 3' ends, TopHat only examines the first 28 base pairs on the 5' end of each read by default.

For each splice junction, Tophat searches the IUM reads in order to find reads that span junctions using a seed-and-extend strategy. The pipeline indexes the IUM reads using a simple lookup table to amortize the cost of searching for a

spliced alignment over many reads. As illustrated in Figure 2.6, TopHat finds any reads that span splice junctions by at least k bases on each side (where k = 5bp by default), so the table is keyed by $2k$-mers, where each $2k$-mer is associated with reads that contain that $2k$-mer. For each read, the table contains $(s-2k+1)$ entries corresponding to possible positions where a splice may fall within a read, where s is the length of the high-quality region on the 5' end (default = 28bp). Users with longer reads may wish to increase s to improve sensitivity. Lowering s will improve running time, but may reduce sensitivity. Increasing k will improve running time, but may limit TopHat to finding junctions only in highly expressed (and thus deeply covered) genes. Reducing it will dramatically increase running time, and while sensitivity will improve, the program may report more false positives. Next TopHat takes each possible splice junction and makes a $2k$-mer "seed" for it by concatenating the k bases downstream of the acceptor to the k bases upstream of the donor. The IUM read index is then queried with this $2k$-mer to find all reads which contain the seed. This exact $2k$-mer match is extended to find all reads that span the splice junction. To extend the exact match for the seed region, TopHat aligns the portions of the read to the left and right of the seed with the left island and right island, respectively, allowing a user-specified number of mismatches. TopHat will miss spliced alignments to reads with mismatches in the seed region of the splice junction, but we expect this tradeoff between speed and sensitivity will be favorable for most users.

The algorithm reports all of the spliced alignments it finds, and then builds a set of non-redundant splice junctions using these alignments. However, some spliced

alignments are discarded prior to reporting junctions in order to avoid reporting false junctions. In their large-scale RNA-Seq study, Wang *et al* reported millions of alternative splicing events in humans and observed that 86% of the minor isoforms were expressed at at least 15% of the level of the major isoform[68]. TopHat's heuristic filter for spliced alignments is based on this observation. For each junction, the average depth of read coverage is computed for the left and right flanking regions of the junction separately. The number of alignments crossing the junction is divided by the coverage of the more deeply covered side to obtain an estimate of the minor isoform frequency. If TopHat estimates that the splice junction occurs at less than 15% of the depth of coverage of the exons flanking it, the junction is not reported. The minimum minor isoform frequency parameter is adjustable by the user, and may be entirely disabled. While the default value in TopHat reflects a result from a human RNA-Seq study, we expect that minor isoforms are expressed at similar frequencies in other mammals, and that the value will be suitable when the software is used to process reads from other mammals.

We compared TopHat with ERANGE on a set of 47,781,892 reads, each 25 bp long, from a recent RNA-Seq study using *Mus musculus* brain tissue [46]. To align reads across splice junctions, ERANGE appends to the reference genome a set of spanning sequences that contain all annotated splice sites. For each splice site, a sequence of length $L - 4$ (for reads of length L) is extracted from the exons flanking that site, and these are concatenated to create a spanning sequence. This constituted a total of 205,151 junctions for *M. musculus*. Mortazavi *et al* trimmed reads to 25bp, so we chose $s = 25$ and $k = 5$, which caused TopHat to report

junctions spanned by the 25 bp on the 5' end of a read, with at least 5 bp on each side of the junction. We also required reads to match the exon sequence on each side of the junction exactly.

For each gene, ERANGE reports the number of mapped reads per kilobase of exon per million mapped reads (RPKM), a measure of transcription activity. The authors characterize 15.0 and 25.0 as moderate and high levels of transcription, respectively. ERANGE reported 108,674 splice junctions in genes with positive RPKM, and 37,675 junctions in genes with RPKM $\geq$ 15.0. TopHat reported 81.9% of the ERANGE junctions in genes above 15.0 RPKM, and 72.2% of all ERANGE junctions. Figure 2.7 shows how TopHat's sensitivity in detecting junctions varies with the RPKM of the genes. An example of TopHat's ability to detect junctions even in genes with very low RPKM is illustrated in Figure 9(a). Of the 30,121 junctions reported by ERANGE and not reported by TopHat, 15,689 (52%) fell within genes expressed below 5 RPKM and were likely missed due to lack of coverage. A further 3,209 (10%) of the missed junctions had RPKM $\geq$ 5.0 but had endpoints more than 20,000bp apart. Filtering based on minor isoform fraction excluded 4,560 (15%). TopHat detected several thousand known splice junctions that ERANGE excluded, presumably during its multiread 'rescue' phase, where it randomly assigns each spliced multiread to matched genes according to their relative expression levels. Of the 104,711 junctions reported by TopHat, 84,988 are listed among the UCSC gene models for *M. musculus*, or 81.1%. The remaining 19,722 may represent novel junctions.

To assess TopHat's ability to identify true junctions without reporting false

Table 2.6: TopHat junction finding under simulated sequencing of transcripts. The simulation sampled a set of transcripts with 9,879 true splice junctions.

Depth of sequence coverage	True Positives	(% total)	False Positives	(% of reported)
1	1744	17	114	6
5	7666	77	585	7
10	8737	88	428	4
25	9275	93	267	2
50	9351	94	235	2

positives, we simulated the results of Illumina short-read sequencing of alternatively spliced genes at several depths. The EMBL-EBI Alternative Splicing Transcript Database (ASTD)[63] contains 1,295 transcripts from mouse chromosome 7. These were generated by the short read simulator from Maq. The simulator computes an empirical distribution of read quality scores and uses these to generate sequencing errors in the reads it produces. We trained the simulator using the reads from the Mortazavi *et al* study, so the sequencing error profile on simulated reads should be similar to the real reads. We generated simulated sequence from the ASTD transcripts, which contained 9,879 splice junctions, at 1-, 5-, 10-, 25-, and 50-fold coverage. TopHat's junction predictions at each coverage level are summarized in Table 2.6. TopHat captures up to 94% of the 9,879 ASTD splice junctions on mouse chromosome 7. Sensitivity suffers when transcripts are sequenced at less than five-fold coverage. TopHat reports few false positives even in deeply sequenced transcripts.

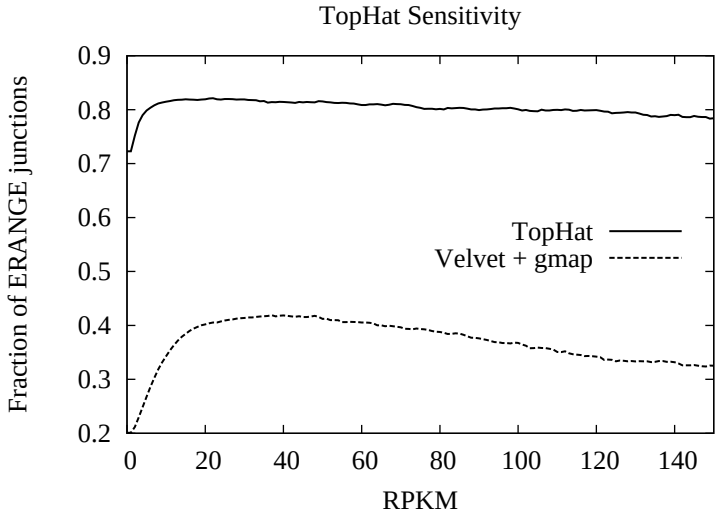


Figure 2.7: TopHat sensitivity as RPKM varies. For genes transcribed above 15.0 RPKM, TopHat detects more than 80% reported by ERANGE in the *M. musculus* brain tissue study. TopHat detects more than 72% of all junctions observed by ERANGE, including those in genes expressed at only a single transcript per cell. A *de novo* assembly of the RNA-Seq reads, followed by spliced alignment of the assembled transcripts produces markedly poorer sensitivity, detecting around 40% of junctions in genes transcribed above 25.0 RPKM, but comparatively few junctions in more highly transcribed genes

The UCSC gene models are relatively conservative, so we searched the GenBank mouse EST database using BLAT[30] for the previously unreported junctions. We also searched the database for known junctions and randomly generated junctions as positive and negative controls, respectively. The positive control group was drawn from the 205,151 junction sequences constructed by Mortazavi *et al* as part of the ERANGE study. The second set consisted of previously unreported junction sequences reported by TopHat. The negative control consisted of random pairings of the left and right halves of junction sequences from the second group. All sequences

in each of the three groups were 42bp long, and each group contained 1,000 sequences chosen randomly. Figure 2.8 shows the distribution of E-values for each sequence's best BLAST hit against the GenBank mouse EST database. As expected, nearly all of the known junctions are confirmed by high-quality hits to ESTs. Also expected is the lack of high-quality hits for sequences in the "random-pairing" negative control. More than 11% of the 1,000 TopHat junctions we searched for actually have high-quality hits to mouse ESTs. In total, 2,543 of the 19,722 junctions not in UCSC gene models had hits to mouse ESTs with E-value $< 1 \times 10^{-6}$.

We examined the previously unreported junctions that lacked high quality hits to mouse EST by dividing them into three categories: junctions between two known exons, junctions between a known exon and a novel one, and junctions between two novel exons. Of the 17,719 junctions without EST hits, 10,499 joined novel exons, 6,077 joined a novel exon with a known one, and 603 joined a pair of known exons. One example of a junction from the second category is occurred in the ADP-ribosylation factor *Arfgef1*, which is important in vesicular trafficking[45]. The junction in figure 9(b) skips two of the gene's 38 exons. TopHat reported several junctions in *Arfgef1* that were previously unknown and indicates that *Arfgef1* is alternatively spliced.

We also compared TopHat to a simple strategy based on *de novo* assembly of RNA-Seq reads. The advantage of such a strategy is that, like TopHat, no known junctions or gene models are needed. We ran the Velvet short read assembler[72] (version 0.7.11, -k=21) on our RNA-Seq reads to produce 149,628 transcript contigs with N50=131. We then aligned these contigs back to the mouse reference genome using

the spliced alignment program GMAP[69], one of the leading methods for alignment of ESTs and full-length cDNAs to genomic DNA. The sensitivity of the Velvet+GMAP method is shown in Figure 2.7. The method detects around 20% of all junctions reported by ERANGE. While the method detects around 40% of junction in genes transcribed above an RPKM value of 25.0, its detection rate decreases as RPKM further increases. We speculate that many of these highly transcribed genes have several alternate isoforms, and that junctions in these genes may cause Velvet to break contigs at the transcript junctions shared by multiple isoforms.

The entire TopHat run took 21 hours, 50 minutes on a 3.0Ghz Intel Xeon 5160 processor, using less than 4GB of RAM, a throughput of nearly 2.2 million reads per CPU hour.

2.4.2 Improved junction discovery with second-generation RNA-Seq

We extended our previous algorithms described to exploit longer paired reads enabled by improvements to the sequencing technology and the RNA-Seq protocol. The original TopHat program used a seed-and-extend alignment strategy to find spliced alignments of unpaired RNA-Seq experiments. However, due to computational limitations, our original method reported only alignments across GT-AG introns shorter than 20Kb by default. This strategy also could not align reads that spanned multiple splice junctions. However, as sequencing technology has improved and longer (paired end) reads have become available, we have modified the software to employ new strategies to align reads across splice junctions. TopHat version 1.0.7

and later splits a read 75bp or longer in three or more segments of approximately equal size (25bp), and maps them independently. Reads with segments that can be mapped to the genome only non-contiguously are marked as possible intron-spanning reads. These "contiguously unmappable" reads are used to build a set of possible introns in the transcriptome. With reads 75bp or longer, TopHat no longer depends on coverage islands in an initial mapping to find junctions. This allows the program to discover junctions within islands at no additional computational cost.

Suppose read S is a read of length l that crosses a splice junction. TopHat splits S into $n = \lfloor k/l \rfloor$ segments, each k bases long, where $k = 25$bp default. At most one of these segments must cross the splice junction. TopHat maps the segments $s_1, ..., s_n$ with Bowtie to the genome, and checks for internal segments $s_2, ..., s_{n-1}$ that do not map anywhere to the genome, as well as for pairs of successive segments s_i, s_{i+1} that both align to the genome, but not adjacently. When a segment s_i fails to align because it crosses a splice junction, but s_{i-1} and s_{i+1} are aligned (say at starting at positions x and y, respectively), TopHat looks for the donor and acceptor sites for the junction near x and y. Assuming the transcript is on Crick strand of the genome (without loss of generality) the donor must fall within k bases upstream of position $x + k$, and the acceptor must be within k bases downstream of y, a total of k possible exon-exon splice junctions. Similarly, when successive segments s_i and s_{i+1} align to the genome non-adjacently at positions x and y, the junction spanned by the read must be from positions $x + k$ to y in the genome. The original TopHat algorithm only discovered introns with canonical (GT-AG) dinucleotides, in order to keep running time low. A single read, with segments aligned on each side

of a potential junction, is sufficiently strong evidence that TopHat does not need to require that the junction be canonical. Thus, TopHat searches for GC-AG and AT-AC introns when aligning reads 50bp or longer (by default).

While early versions of TopHat used a seed-and-extent strategy to align spliced reads, versions since 1.0.7 construct a Bowtie index of splice sequences on the fly. The advantage of this approach is that junction discovery is separated from spliced read alignment, and spliced alignments are not more constrained in terms of allowable mismatches than contiguous genomic alignment. Moreover, user-supplied junctions or those from annotation can be mixed into the junction database along with newly discovered ones. This allows users to exploit other sources of intron evidence, such as spliced EST alignments, homologous gene structures from related species, and computationally predicted genes from software such as Glimmer[57] or Augustus[61]. For each junction the program concatenates $k$bp upstream of the donor to $k$bp downstream of the acceptor to form a synthetic spliced sequence around the junction. The segments of the contiguously unmappable reads are then aligned against these synthetic sequences with Bowtie. The resulting contiguous and spliced segment alignments for these reads are merged to form complete alignments to the genome, each spanning one or more splice junctions.

2.4.3 Resolving multiple alignments for fragments

The alignments for both reads from a mate pair are examined together to produce a set of alignments for the corresponding library fragment as a whole,

reported in SAM format[37]. These fragment alignments are ranked heuristically, and only highest ranking alignments are reported. The ranks are designed to incorporate very loose assumptions on intron and gene length, namely that introns longer than 20kb are rare. Let x and y be fragment alignments. Then $x < y$ if *any* of the following (applied in order) are true:

1. x is a singleton, and y has both ends mapped
2. x cross more splice junctions than y
3. The reads for x map significantly farther apart in the genome than expected according to the library's fragment length distribution ($\geq$ 3 s.d.), and y's are not.
4. The reads for x are significantly closer together than expected according to the library's fragment length distribution, and y's are not.
5. x's reads map more than 100bp farther apart than y's
6. x and y both span an intron, and x spans a longer one.
7. x has more mismatches than y to the genome.

Fragments that have multiple equally good alignments according to the above rules are ambiguously mapped, and so all of the equally good alignments are reported. If there are n alignments for a fragment, each has a probability of only $1/n$ of being correct. The SAM format encodes this probability in the mapping quality field, which is later used by Cufflinks to reduce the contribution of multiply mapping

fragments (to $1/n$ of a uniquely mappable read) in FPKM calculations (FPKM is a measurement of expression, and is formally defined in Chapter 3).

Using first generation RNA-Seq reads, TopHat reported more than 72% of all exon splice junctions captured by the ERANGE annotation-based analysis pipeline, including junctions from genes transcribed at around one transcript per cell. TopHat captured around 80% of splice junctions in more actively transcribed genes. More significant is its ability to detect novel splice junctions. While it is difficult to assess how many of TopHat's 19,722 newly discovered junctions are genuine, TopHat's alignment parameters for this run were quite strict: only exact matches were reported for splice junctions, and reads were required to have relatively long anchors on each side of the splice site. Close inspection of junctions strengthened the case that many are true splices. The TopHat pipeline processed an entire RNA-Seq run in less than a day on a single processor of a standard workstation. ERANGE is appropriate for high-quality measurement of gene expression in mammalian RNA-Seq projects, provided that a reliable annotation of exon-exon junctions is available. QPALMA can accurately align short reads across junctions without an annotation, but makes such substantial sacrifices in speed that it may not be practical for large mammalian projects. TopHat thus represents a significant advance over previous RNA-Seq splice detection methods, both in its performance and its ability to find junctions *de novo*.

The TopHat pipeline and its default parameter values are designed for detecting junctions even in genes transcribed at very low levels. However, the system may fail to detect junctions for a variety of reasons. The most common reason for missing

a junction is that the transcript has very low sequencing coverage, in which case there might be no read that straddles the junction with sufficient sequence on each side. With first-generation reads, junctions spanning very long introns or introns with non-canonical donor and acceptor sites (such as GC-AG introns) will also be missed. New RNA-Seq protocols that produce long, paired-end reads have made TopHat's task easier.

2.5 Mapping of reads from the myogenesis case study

The exact distribution of the C2C12 fragment lengths is shown in Figure 2.10 (in Chapters 3 and 5 this distribution of fragment lengths is referred to as F). These estimates are based on alignments of the spiked-in sequences using Bowtie 0.12[31] (see Chapter 2).

Fragments were mapped to build 37.1 of the mouse genome with TopHat version 1.0.13.

Sample	**Sequenced fragments**	**Aligned fragments**	**Singleton fragments**	**Spliced fragments**	**Multi-mapping fragments**	**Total alignments**
-24 hours	42,184,539	35,852,366	11,031,886	8,824,825	1,768,041	41,663,170
60 hours	70,192,031	57,071,494	18,104,211	15,778,114	2,265,378	64,637,511
120 hours	41,069,106	27,914,989	14,431,734	7,711,026	1,881,772	33,929,133
168 hours	61,787,833	50,705,080	20,396,250	14,585,287	2,458,292	58,797,912
Total	215,233,509	171,543,929	63,964,081	46,899,252	8,373,483	199,027,726

Table 2.10: Number of fragments sequenced, aligned and mapped with TopHat.

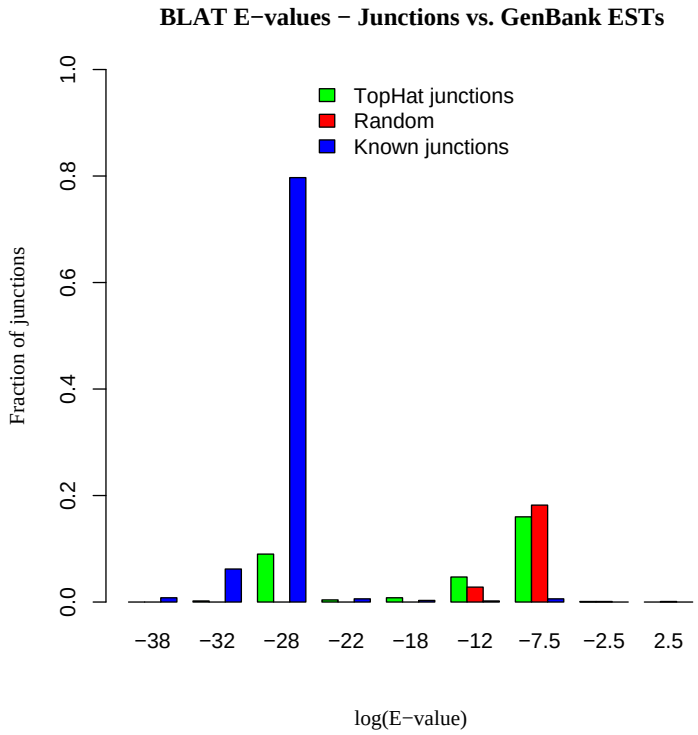
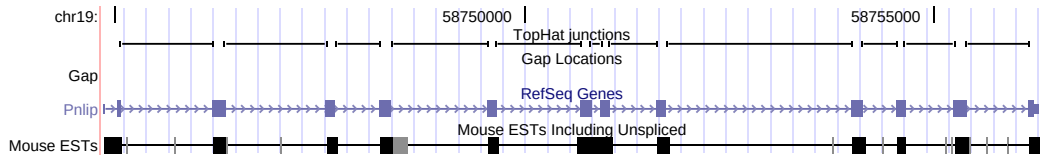
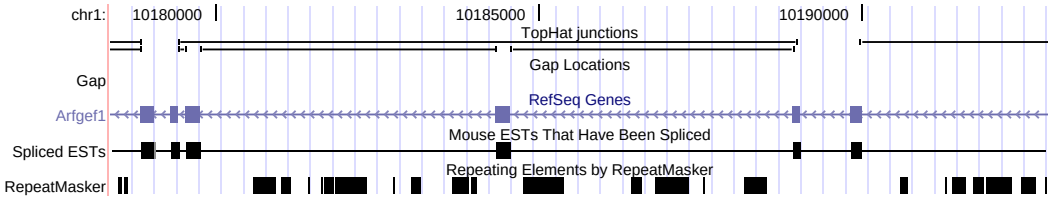


Figure 2.8: The BLAT E-value distribution of known, previously unreported, and randomly generated splice junction sequences when searched against GenBank mouse ESTs. As expected, known junctions have high-quality BLAT hits to the EST database. Randomly-generated junction sequences do not. High-quality BLAT hits for more than 11% of the junctions identified by TopHat suggest that the UCSC gene models for mouse are incomplete. These junctions are almost certainly genuine, and because the mouse EST database is not complete, 11% is only a lower bound on the specificity of Tophat



(a) Junctions detected in genes transcribed at low levels.



(b) A novel junction in *Arfgef1*

Figure 2.9: (a) TopHat detects junctions in genes transcribed at very low levels. The gene *Pnlip* was transcribed at only 7.88 RPKM in the brain tissue according to ERANGE, and yet TopHat reports the complete known gene model. (b) A previously unreported splice junction detected by TopHat is shown as the topmost horizontal line. This junction skips two exons in the ADP-ribosylation gene *Arfgef1*.

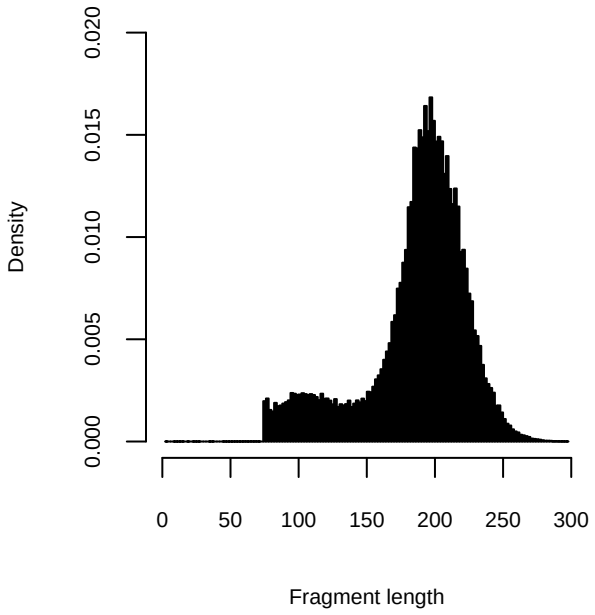


Figure 2.10: Length distribution of C2C12 RNA-Seq fragments.

Chapter 3

Estimating transcript abundances

For the purposes of estimating transcript abundances, we developed a statistical model parameterized by the abundances of these transcript sequences. Cufflinks model allows for the probabilistic deconvolution of RNA-Seq fragment densities to account for cases where genome alignments of fragments do not uniquely correspond to source transcripts. The model incorporates minimal assumptions[5] about the sequencing experiment, and extends the single read sequencing model of Jiang and Wong[28] to the paired-end case. Despite the added complexity, the likelihood function remains concave, allowing us to find the maximum likelihood estimates of abundances numerically. Abundances were reported in Fragments Per Kilobase of transcript per Million fragments mapped (FPKM). Confidence intervals for estimates were obtained using a Bayesian inference method based on importance sampling from the posterior distribution. Abundances of spiked control sequences and benchmarks with simulated data revealed that Cufflinks abundance estimates are highly accurate. The inclusion of novel isoforms of known genes during abundance estimation had a dramatic impact on the estimates of known isoforms in many genes, highlighting the importance of coupling transcript discovery together with

This chapter discusses a statistical model of RNA-Seq experiments, and is joint work with Lior Pachter. The validation of this model is joint work with Brian Williams, Ali Mortazavi, Gordon Kwan, and Barbara Wold.

abundance estimation.

3.1 Definitions

A *transcript* is an RNA molecule that has been transcribed from DNA. A *primary transcript* is an RNA molecule that has yet to undergo modification. The *genomic location* of a primary transcript consists of a pair of coordinates in the genome representing the $5'$ transcription start site and the $3'$ polyadenylation cleavage site. We denote the set of all transcripts in a transcriptome by T. We partition transcripts into *transcription loci* (for simplicity we refer to these as loci) so that every locus contains a set of transcripts all of whose genomic locations do not overlap the genomic location of any transcript in any other locus. Formally, we consider a maximal partition of transcripts into loci, a partition denoted by G, where the genomic location of a transcript $t \in g \in G$ does not overlap the genomic location of any transcript u where $u \in h \in G$ and $h \neq g$. We emphasize that the definition of a transcription locus is not biological; transcripts in the same locus may be regulated via different promoters, and may differ completely in sequence (for example if one transcript is in the intron of another) or have different functions. The reason for defining loci is that they are computationally convenient.

We assume that at the time of an experiment, a transcriptome consists of an ensemble of transcripts T where the proportion of transcript $t \in T$ is ρ_t, so that $\sum_{t \in T} \rho_t = 1$ and $0 \leq \rho_t \leq 1$ for all $t \in T$. Formally, a *transcriptome* is a set of transcripts T together with the abundances $\rho = \{\rho_t\}_{t \in T}$. For convenience

we also introduce notation for the proportion of transcripts in each locus. We let $\sigma_g = \sum_{t \in g} \rho_t$. Similarly, within a locus g, we denote the proportion of each transcript $t \in g$ by $\tau_t = \frac{\rho_t}{\sigma_g}$. We refer to ρ, σ and τ as *transcript abundances*.

Transcripts have lengths, which we denote by $l(t)$. For a collection of transcripts $S \subset T$ in a transcriptome, we define the length of S using the weighted mean:

$$l(S) = \frac{\sum_{t \in S} \rho_t l(t)}{\sum_{t \in S} \rho_t}. \tag{3.1}$$

It is important to note that the length of a set of transcripts depends on their relative abundances; the reason for this will be clear later.

One grouping of transcripts that we will focus on is the set of transcripts within a locus that share the same transcription start site (TSS). Unlike the concept of a locus, grouping by TSS has a biological basis. Transcripts within such a group are by definition alternatively spliced, and if they have different expression levels, this is most likely due to the spliceosome and not due to differences in transcriptional regulation.

3.2 A statistical model for RNA-Seq

In order to analyze expression levels of transcripts with RNA-Seq data, it is necessary to have a model for the (stochastic) process of sequencing. A *sequencing experiment* consists of selecting a total of M fragments of transcripts uniformly at random from the transcriptome. Each fragment is identified by sequencing from its ends, resulting in two reads called *mate pairs*. The length of a fragment is a

random variable, with a distribution we will denote by F. That is, the probability that a fragment has length i is $F(i)$ and $\sum_{i=1}^{\infty} F(i) = 1$. In this paper we assume that F is normal, however in principle F can be estimated using data from the experiment (e.g. spike-in sequences). We decided to use the normal approximation to F (allowing for user specified parameters of the normal distribution) in order to simplify the requirements for running Cufflinks at this time.

The assumption of random fragment selection is known to oversimplify the complexities of a sequencing experiment, however without rigorous ways to normalize we decided to work with the uniform at random assumption. It is easy to adapt the model to include more complex models that address sequencing bias as RNA-Seq experiments mature and the technologies are better understood.

The transcript abundance estimation problem in paired-end RNA-Seq is to estimate ρ given a set of transcripts T and a set of reads sequenced from the ends of fragments. In Cufflinks, the transcripts T can be specified by the user, or alternatively T can be estimated directly from the reads. The latter problem is the transcript assembly problem which we discuss in Chapter 4.

The fact that fragments have different lengths has bearing on the calculation of the probability of selecting a fragment from a transcript. Consider a transcript t with length $l(t)$. The probability of selecting a fragment of length k from t at one of the positions in t assuming that it is selected uniformly at random, is $\frac{1}{l(t)-k}$. For this reason, we will define an adjusted length for transcripts as

$$\tilde{l}(t) = \sum_{i=1}^{l(t)} F(i)(l(t) - i + 1). \tag{3.2}$$

We also revisit the definition of length for a group of transcripts, and define

$$\tilde{l}(S) = \frac{\sum_{t \in S} \rho_t \tilde{l}(t)}{\sum_{t \in S} \rho_t}. \tag{3.3}$$

It is important to note that given a read it may not be obvious from which transcript the fragment it was sequenced from originated. The consistency of fragments with transcripts is important and we define the *fragment-transcript matrix* $A_{R,T}$ to be the $M \times |T|$ matrix with $A(r,t) = 1$ if the fragment alignment r is completely contained in the genomic interval spanned by t, and all the implied introns in r match introns in t (in order), and with $A(r,t) = 0$ otherwise. Note that the reads in Figure 1.5c are colored according to the matrix $A_{R,T}$, with each column of the matrix corresponding to one of the three colors (yellow, blue, red) and reads colored according to the mixture of colors corresponding to the transcripts their fragments are contained in.

Even given the read alignment to a reference genome, it may not be obvious what the length of the fragment was. Formally, in the case that $A_{R,T}(r,t) = 1$ we denote by $I_t(r)$ the fragment length from within a transcript t implied by the (presumably unique) sequences corresponding to the mate pairs of a fragment r. If $A_{R,T}(r,t) = 0$ then $I_t(r)$ is set to be infinite and $F(I_t(r)) = 0$.

Given a set of reads, we assume that we can identify for each of them the set of transcripts with which the fragments the reads belonged to are consistent. The rationale for this assumption is the following: we map the reads to a reference

genome, and we assume that the read lengths are sufficiently long so that mate-pairs can be aligned to the genome. We refer to the alignment of a pair of mated reads to the genome as a single *fragment alignment*. We also assume that we know all the possible transcripts and their alignments to the genome. Therefore, we can identify for each read the possible transcripts from which the fragment it belonged to originated.

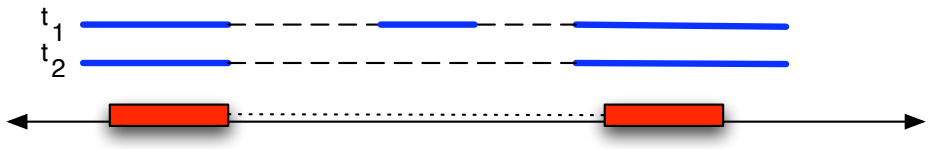


Figure 3.1: Alignments of reads to the genome (rectangles) may be consistent with multiple transcripts (in this case both t_1 and t_2). The transcripts t_1 and t_2 differ by an internal exon; introns are indicated by long dashed lines. If we denote the fragment alignment by r, this means that $A_{R,T}(r, t_1) = 1$ and $A_{R,T}(r, t_2) = 1$. It is apparent that the implied length $I_{t_1}(r) > I_{t_2}(r)$ due to the presence of the extra internal exon in t_1.

We are now ready to write down the likelihood equation for the model. We will write $L(\rho|R)$ for the likelihood of a set of fragment alignments R constructed from M reads. The notation $Pr(trans. = t)$ means "the probability that a fragment selected at random originates from transcript t".

$$L(\rho|R) = \prod_{r\in R} Pr(rd.\ aln. = r) \tag{3.4}$$

$$= \prod_{r\in R}\sum_{t\in T} Pr(rd.\ aln. = r|trans. = t)Pr(trans. = t) \tag{3.5}$$

$$= \prod_{r\in R}\sum_{t\in T} \frac{\rho_t\tilde{l}(t)}{\sum_{u\in T}\rho_u\tilde{l}(u)} Pr(rd.\ aln. = r|trans. = t) \tag{3.6}$$

$$= \prod_{r\in R}\sum_{t\in T} \frac{\rho_t\tilde{l}(t)}{\sum_{u\in T}\rho_u\tilde{l}(u)} \left(\frac{F(I_t(r))}{l(t) - I_t(r) + 1}\right) \tag{3.7}$$

$$= \prod_{r\in R}\sum_{t\in T} \alpha_t \left(\frac{F(I_t(r))}{l(t) - I_t(r) + 1}\right), \tag{3.8}$$

where

$$\alpha_t = \frac{\rho_t\tilde{l}(t)}{\sum_{u\in T}\rho_u\tilde{l}(u)}. \tag{3.9}$$

Observe that α_t is exactly the probability that a fragment selected at random comes from transcript t, and we have that $\sum_{t\in T}\alpha_t = 1$. In light of the probabilistic meaning of the $\alpha = \{\alpha_t\}_{t\in T}$, we refer to them as *fragment abundances*.

It is evident that the likelihood function is that of a linear model and that the likelihood function is concave (Proposition 15) so a numerical method can be used to find the α. It is then possible, in principle, to recover the ρ using Lemma 14. However the number of parameters is in the tens of thousands, and in practice this form of the likelihood function is unwieldy. Instead, we re-write the likelihood utilizing the fact that transcripts in distinct loci do not overlap in genomic location.

We first calculate the probability that a fragment originates from a transcript

within a given locus g:

$$\begin{aligned}
\beta_g &:= \sum_{t \in g} \alpha_t && (3.10)\\
&= \frac{\sum_{t \in g} \rho_t \tilde{l}(t)}{\sum_{u \in T} \rho_u \tilde{l}(u)} && (3.11)\\
&= \frac{\sum_{t \in g} \sigma_g \tau_t \tilde{l}(t)}{\sum_{h \in G} \sum_{u \in h} \sigma_h \tau_u \tilde{l}(u)} && (3.12)\\
&= \frac{\sigma_g \sum_{t \in g} \tau_t \tilde{l}(t)}{\sum_{h \in G} \sigma_h \sum_{u \in h} \tau_u \tilde{l}(u)} && (3.13)\\
&= \frac{\sigma_g \tilde{l}(g)}{\sum_{h \in G} \sigma_h \tilde{l}(h)}. && (3.14)
\end{aligned}$$

Recall that $\sigma_g = \sum_{t \in g} \rho_t$ and that $\tau_t = \frac{\rho_t}{\sigma_g}$ for a locus g.

Similarly, the probability of selecting a fragment from a single transcript t conditioned on selecting a transcript from the locus g in which t is contained is

$$\gamma_t = \frac{\tau_t \tilde{l}(t)}{\sum_{u \in g} \tau_u \tilde{l}(u)}. \tag{3.15}$$

The parameters $\gamma = \{\gamma_t\}_{t \in g}$ are conditional fragment abundances, and they are the parameters we estimate from the data in the next Section. Note that for a transcript $t \in g$, $\alpha_t = \beta_g \cdot \gamma_t$ and it is easy to convert between fragment abundances and transcript abundances using Lemma 14.

We denote the fragment counts by X; specifically, we denote the number of alignments in locus g by X_g. Note that $\sum_{g \in G} X_g = M$. We also use the notation g_r to denote the (unique) locus from which a read alignment r can be obtained.

The likelihood function is given by

$$L(\rho|R) = \prod_{r\in R} Pr(aln. = r) \tag{3.16}$$

$$= \prod_{r\in R}\sum_{g\in G} Pr(aln. = r|loc. = g)Pr(loc. = g) \tag{3.17}$$

$$= \prod_{r\in R} \frac{\sigma_{g_r}\tilde{l}(g_r)}{\sum_{g\in G}\sigma_g\tilde{l}(g)} Pr(aln. = r|loc. = g_r) \tag{3.18}$$

$$= \prod_{r\in R} \beta_{g_r} \sum_{t\in g_r} Pr(aln. = r|loc. = g_r, trans. = t)Pr(trans. = t|loc. = g_r) \tag{3.19}$$

$$= \prod_{r\in R} \beta_{g_r} \sum_{t\in g_r} \frac{\tau_t\tilde{l}(t)}{\sum_{u\in g_r}\tau_u\tilde{l}(u)} Pr(aln. = r|loc. = g_r, trans. = t) \tag{3.20}$$

$$= \left(\prod_{r\in R}\beta_{g_r}\right)\left(\prod_{r\in R}\sum_{t\in g}\gamma_t \cdot Pr(aln. = r|loc. = g_r, trans. = t)\right) \tag{3.21}$$

$$= \left(\prod_{r\in R}\beta_{g_r}\right)\left(\prod_{r\in R}\sum_{t\in g}\gamma_t \cdot \frac{F(I_t(r))}{l(t) - I_t(r) + 1}\right) \tag{3.22}$$

$$= \left(\prod_{g\in G}\beta_g^{X_g}\right)\left(\prod_{g\in G}\left(\prod_{r\in R: r\in g}\sum_{t\in g}\gamma_t \cdot \frac{F(I_t(r))}{l(t) - I_t(r) + 1}\right)\right). \tag{3.23}$$

Explicitly, in terms of the parameters ρ, Equation (3.23) simplifies to Equation (3.8) but we will see in the next section how the maximum likelihood estimates $\hat{\rho}$ are most conveniently obtained by first finding $\hat{\beta}$ and $\hat{\gamma}$ using Equation (3.23).

We note that it is biologically meaningful to include prior distributions on σ and τ that reflect the inherent stochasticity and resulting variability of transcription in a cell. This will be an interesting direction for further research as more RNA-Seq data (with replicates) becomes available allowing for the determination of biologically meaningful priors. In particular, it seems plausible that specific isoform abundances may vary considerably and randomly within cells from a single tissue

and that this may be important in studying differential splicing. We mention to this to clarify that in this paper, the confidence intervals we report represent the variability in the maximum likelihood estimates $\hat{\sigma}_j$ and $\hat{\tau}_j^k$, and are not the variances of prior distributions.

3.3 Estimation of parameters

We begin with a discussion of identifiability of our model. Identifiability refers to the injectivity of the model, i.e.,

$$\text{if } Prob(\rho_1|r) = Prob(\rho_2|r), \forall r, \text{ then } \rho_1 = \rho_2. \tag{3.24}$$

The identifiability of RNA-Seq models was discussed in[25], where a standard analysis for linear models is applied to RNA-Seq (for another related biological example, see[52]which discusses identifiability of haplotypes in mixed populations from genotype data). The results in these papers apply to our model. For completeness we review the conditions for identifiability. Recall that $A_{R,T}$ is the fragment-transcript matrix that specifies which transcripts each fragment is compatible with. The following theorem provides a simple characterization of identifiability:

Theorem 2. *The RNA-Seq model is identifiable iff $A_{R,T}$ is full rank.*

Therefore, for a given set of transcripts and a read set R, we can test whether the model is identifiable using elementary linear algebra. For the results in this paper, when estimating expression with given annotations, when the model was not identifiable we picked *a* maximum likelihood solution, although in principle it

is possible to bound the total expression of the locus and/or report identifiability problems to the user.

Returning to the likelihood function

$$\left(\prod_{g\in G}\beta_g^{X_g}\right)\left(\prod_{g\in G}\left(\prod_{r\in R:r\in g}\sum_{t\in g}\gamma_t\cdot\frac{F(I_t(r))}{l(t)-I_t(r)+1}\right)\right), \tag{3.25}$$

we note that both the β and γ parameters depend on the ρ parameters. However, we will see that if we maximize the β separately from the γ, and also each of the sets $\{\gamma_t : t \in g\}$ separately, then it is always possible to find ρ that match both the maximal β and γ. In other words, the problem of finding $\hat{\rho}$ is equivalent to finding $\hat{\beta}$ that maximizes $\prod_{g\in G}\beta_g^{X_g}$ and separately, for each locus g, the $\hat{\gamma}_t$ that maximize

$$\prod_{r\in R:r\in g}\sum_{t\in g}\gamma_t\frac{F(I_t(r))}{l(t)-I_t(r)+1}. \tag{3.26}$$

We begin by solving for the $\hat{\beta}$ and $\hat{\gamma}$ and the variances of the maximum likelihood estimates, and then explain how these are used to report expression levels.

We can solve for the $\hat{\gamma}$ using the fact that the model is linear. That is, the probability of each individual read is linear in the read abundances γ_t. It is a standard result in statistics (see, e.g., Proposition 1.4 in Pachter and Sturmfels[51]) that the log likelihood function of a linear model is concave. Thus, a hill climbing method can be used to find the $\hat{\gamma}$. We used the EM algorithm for this purpose.

Rather than using the direct ML estimates, we obtained a regularized estimate by importance sampling from the posterior distribution with a proposal distribution we explain below. The samples were also used to estimate variances for our estimates.

It follows from standard MLE asymptotic theory that the $\hat{\gamma}$ are asymptotically multivariate normal with variance-covariance matrix given by the inverse of the observed Fisher information matrix. This matrix is defined as follows:

Definition 3 (Observed Fisher information matrix)**.** The observed Fisher information matrix is the negative of the Hessian of the log likelihood function evaluated at the maximum likelihood estimate. That is, for parameters $\Theta = (\theta_1, \ldots, \theta_n)$, the $n \times n$ matrix is

$$\mathcal{F}_{k,l}(\hat{\Theta}) = -\frac{\partial^2 \log(\mathcal{L}(\Theta|R))}{\partial\theta_k\theta_l}|_{\theta=\hat{\theta}}. \tag{3.27}$$

In our case, considering a single locus g, the parameters are $\Theta = (\gamma_{t_1}, \ldots, \gamma_{t_{|g|}})$, and as expected from Proposition 15:

$$\mathcal{F}_{t_k,t_l}(\hat{\Theta}) = \sum_{r\in R: r\in g} \left[\frac{1}{\left(\sum_{h\in g} \hat{\gamma}_h \frac{F(I_h(r))}{l(h)-I_h(r)+1}\right)^2} \frac{F(I_{t_k}(r))F(I_{t_l}(r))}{(l(t_k)-I_{t_k}+1)(l(t_l)-I_{t_l}+1)} \right]. \tag{3.28}$$

Because some of the transcript abundances may be close to zero, we adopted the Bayesian approach of[28] and instead sampled from the joint posterior distribution of Θ using the proposal distribution consisting of the multivariate normal with mean given by the MLE, and variance-covariance matrix given by the inverse of (3.28). If the Observed Fisher Information Matrix is singular then the user is warned and the confidence intervals of all transcripts are set to $[0, 1]$ (meaning that there is no information about relative abundances).

The method used for sampling was importance sampling. The samples were used to obtain a maximum-a-posterior estimate for $\hat{\gamma}_t$ for each t and for the variance-

covariance matrix which we denote by Ψ^g (where $g \in G$ denotes the locus). Note that Ψ^g is a $|g| \times |g|$ matrix. The covariance between $\hat{\gamma}_{t_k}$ and $\hat{\gamma}_{t_l}$ for $t_k, t_l \in g$ is given by $\Psi^g_{t_k,t_l}$.

Turning to the maximum likelihood estimates $\hat{\beta}$, we use the fact that the model is the log-linear. Therefore,

$$\hat{\beta}_g = \frac{X_g}{M}. \tag{3.29}$$

Viewed as a random variable, the counts X_g are approximately Poisson and therefore the variance of the MLE $\hat{\beta}_g$ is approximately X_g. We note that for the tests in this paper we directly used the total counts M and the proportional counts X_g, however it is easy to incorporate recent suggestions for total count normalization, such as quantile normalization[5] into Cufflinks.

The abundance of a transcript $t \in g$ in FPKM units is

$$\frac{10^6 \cdot 10^3 \cdot \alpha_t}{\tilde{l}(t)} = \frac{10^6 \cdot 10^3 \cdot \beta_g \cdot \gamma_t}{\tilde{l}(t)}. \tag{3.30}$$

Equation (3.30) makes it clear that although the abundance of each transcript $t \in g$ in FPKM units is proportional to the transcript abundance ρ_t it is given in terms of the read abundances β_g and γ_t which are the parameters estimated from the likelihood function.

The maximum likelihood estimates of β_g and γ_t are random variables, and we denote their scaled product (in FPKM units) by A_t. That is $Pr(A_t = a)$ is the probability that for a random set of fragment alignments from a sequencing experiment, the maximum likelihood estimate of the transcript abundance for t in FPKM units is a.

Using the fact that the expectation of a product of independent random variables is the product of the expectations, for a transcript $t \in g$ we have

$$E[A_t] = \frac{10^9 X_g \hat{\gamma}_t}{\tilde{l}(t)M}. \tag{3.31}$$

Given the variance estimates for the $\hat{\gamma}_t$ we turn to the problem of estimating $Var[A_t]$ for a transcript $t \in g$. We use Lemma 13 to obtain

$$\begin{aligned} Var[A_t] &= \left(\frac{10^9}{\tilde{l}(t)M}\right)^2 \left(\Psi^g_{t,t} X_g + \Psi^g_{t,t} X_g^2 + (\hat{\gamma}_t)^2 X_g\right) && (3.32) \\ &= X_g \left(\frac{10^9}{\tilde{l}(t)M)}\right)^2 \left(\Psi^g_{t,t}(1 + X_g) + (\hat{\gamma}_t)^2\right). && (3.33) \end{aligned}$$

This variance calculation can be used to estimate a confidence interval by utilizing the fact[2] that when the expectation divided by the standard deviation of at least one of two random variables is large, their product is approximately normal.

Next we turn to the problem of estimating expression levels (and variances of these estimates) for groups of transcripts. Let $S \subset T$ be a group of transcripts located in a single locus g, e.g. a collection of transcripts sharing a common TSS.

The analogy of Equation (3.30) for the FPKM of the group is

$$\begin{aligned} & \frac{10^6 \cdot 10^3 \cdot \beta_g \cdot \left(\sum_{t \in S} \gamma_t\right)}{\tilde{l}(S)} && (3.34) \\ = \; & 10^6 \cdot 10^3 \cdot \beta_g \cdot \sum_{t \in S} \frac{\gamma_t}{\tilde{l}(t)}. && (3.35) \end{aligned}$$

As before, we denote by B_S the random variables for which $Pr(B_S = b)$ is the probability that for a random set of fragment alignments from a sequencing experiment, the maximum likelihood estimate of the transcript abundance for all the transcripts in S in FPKM units is b. We note that the B_S are products and sums of

random variables (Equation (3.35)). This makes Equation (3.35) significantly more useful than the equivalent unsimplified Equation (3.34), especially because $\tilde{l}(S)$ is, in general, a ratio of two random variables.

We again use the fact that the expectation of independent random variables is the product of the expectation, in addition to the fact that expectation is a linear operator to conclude that for a group of transcripts S,

$$E[B_S] = \frac{10^9 \cdot X_g \cdot \sum_{t \in S} \frac{\hat{\gamma}_t}{\tilde{l}(t)}}{M}. \tag{3.36}$$

In order to compute the variance of B_S, we first note that

$$Var\left[\sum_{t \in S} \frac{\hat{\gamma}_t}{\tilde{l}(t)}\right] = \sum_{t \in S} \frac{1}{\tilde{l}(t)^2} \Psi^g_{t,t} + \sum_{t,u \in S} \frac{1}{\tilde{l}(t)\tilde{l}(u)} \Psi^g_{t,u}. \tag{3.37}$$

Therefore,

$$\begin{aligned} &Var[B_S] \quad = \\ &X_g \left(\frac{10^9}{M}\right)^2 \left((1 + X_g)\left(\sum_{t \in S} \frac{1}{\tilde{l}(t)^2} \Psi^g_{t,t} + \sum_{t,u \in S} \frac{1}{\tilde{l}(t)\tilde{l}(u)} \Psi^g_{t,u}\right) + \left(\sum_{t \in S} \frac{\hat{\gamma}_t}{\tilde{l}(t)}\right)^2\right). \end{aligned} \tag{3.38}$$

We can again estimate a confidence interval by utilizing the fact that B_S is approximately normal[2].

3.4 Assessment of abundance estimation

We evaluated the accuracy of Cufflinks' transcript abundance estimates by first comparing the estimated FPKM values for the spiked-in sequences in each sample against their intended concentrations (see C.2). Spike FPKMs were highly

correlated across a 5-log dynamic range in all four samples (Figure 3.2). However, because sequenced spike fragments were unambiguously mappable, we performed additional simulation to measure the accuracy of the software in alternatively spliced loci.

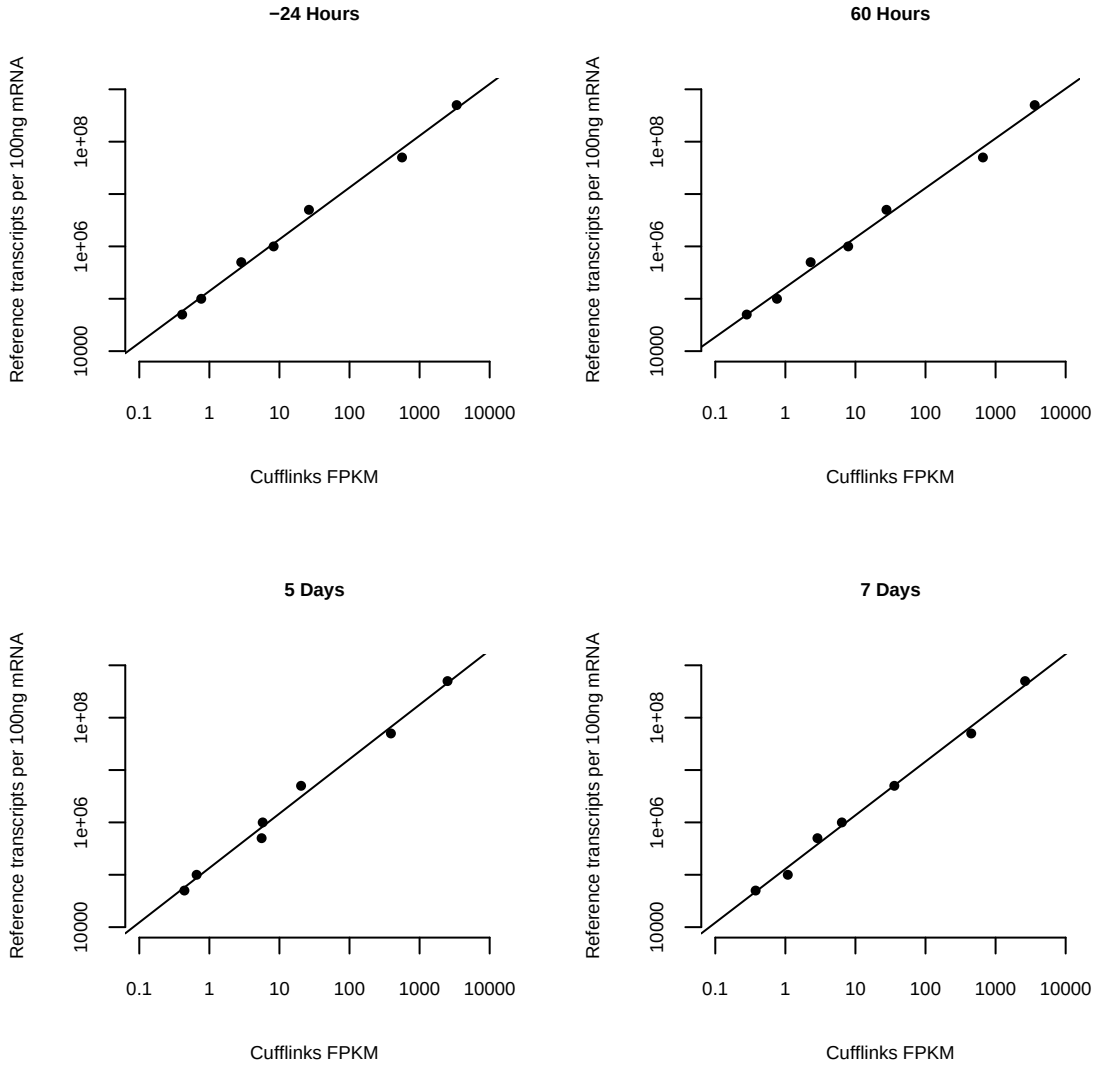


Figure 3.2: Cufflinks' abundance estimates of spiked-in sequences.

To assess the accuracy of Cufflinks' estimates, we simulated an RNA-Seq

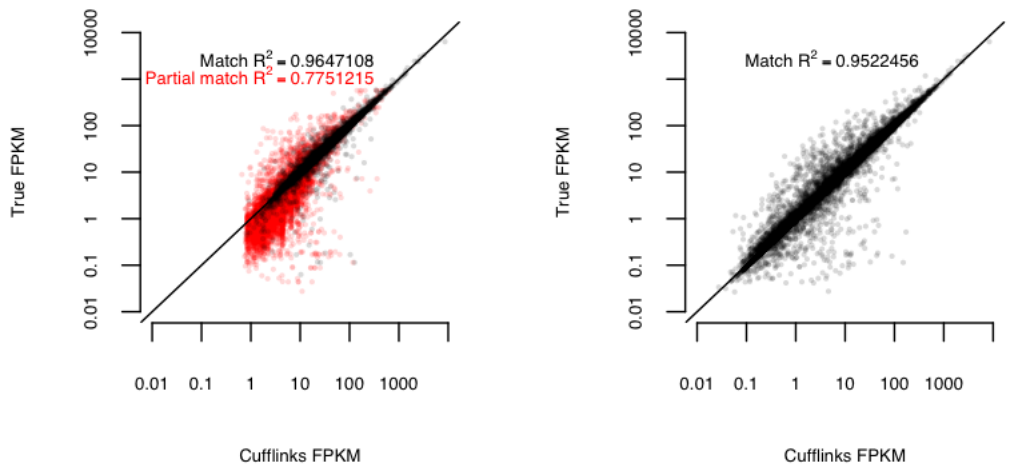


Figure 3.3: *In silico* assessment of the accuracy of Cufflinks abundance estimation when provided with a perfect assembly (a) and after *de novo* comparative assembly (b). Red points indicate in silico transcripts that were only partially recovered, where black points were fully reconstructed by Cufflinks. Simulated reads were aligned with TopHat and the alignments were provided to Cufflinks along with the structures of the transcripts in the simulated sample.

experiment using the FluxSimulator (http://flux.sammeth.net), a freely available software package that models whole-transcriptome sequencing experiments with the Illumina Genome Analyzer. The software works by first randomly assigning expression values to the transcripts provided by the user, constructing an amplified, size-selected library, and sequencing it. Mouse UCSC transcripts were supplied to the software, along with build 37.1 of the genome. FluxSimulator then randomly assigned expression ranks to 18,935 transcripts, with the expression value y computed from the rank x according to the formula

$$y = \left(\frac{x}{5.0 \times 10^7}\right)^{-0.6} e^{-\left(\frac{x}{9.5\times10^3}\right)-\left(\frac{x}{9.5\times10^3}\right)^2}. \quad (3.39)$$

From these relative expression levels, the software constructed an *in silico* RNA sample, with each transcript assigned a number of molecules according to its abundances. The software modeled the polyadenylation of each transcript by adding a poly-A tail (of mean length 125nt) after the terminal exon. FluxSimulator then simulated reverse transcription of *in silico* mRNAs by random hexamer priming, followed by size selection of RT products to between 175 and 225 nt. The resulting "library" of 6,601,805 cDNA fragments was then sampled uniformly at random for simulated sequencing, where the initial and terminal 75bp of each selected fragment were reported as reads. FluxSimulator does not allow precise control over the number of reads generated (Michael Sammeth, personal communication), but nevertheless generated 13,203,516 75nt paired-end RNA-Seq reads. These reads included sequencing errors; FluxSimulator includes a position-specific sequencing error model.

Fragments were mapped with TopHat to the mouse genome using identical parameters to those used to map the C2C12 reads, mapping a total of 6,176,961 (93% of the library). These alignments were supplied along with the exact set of expressed transcripts to Cufflinks, to measure Cufflinks' abundance estimation accuracy when working with a "perfect" assembly (Figure 3.3). Estimated FPKM was very close to true *in silico* FPKM across a dynamic range of expression of nearly six orders of magnitude ($R^2 = 0.95$).

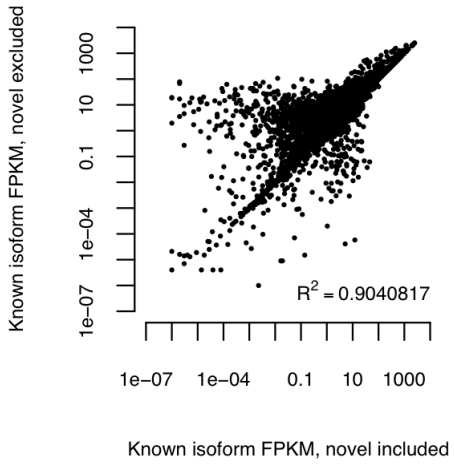


Figure 3.4: Excluding novel C2C12 transcripts from abundance estimation results in inaccurate estimates for known transcripts.

Estimation of transcript abundances by assigning fragments to them may be inaccurate if one is working with an incomplete set of transcripts for a particular sample. To evaluate the impact of missing transcripts, we removed the newly discovered transcripts from our high-confidence set and re-estimated the abundances of known transcripts, and then compared them to those obtained when working with the complete high-confidence set. While estimates of known transcripts were overall similar or identical when working with both sets, reflecting single-isoform or fully annotated genes, isoforms of some alternatively spliced genes differed greatly. (Figure 3.4)

Chapter 4

Assembly of full-length transcripts

To recover the minimal set of transcripts supported by our fragment alignments, we designed a comparative transcriptome assembly algorithm. EST assemblers such as PASA introduced the idea of collapsing alignments to transcripts based on splicing compatibility[23], and Dilworths Theorem[14] has been used to assemble a parsimonious set of haplotypes from virus population sequencing reads[16]. Cufflinks extends these ideas, reducing the transcript assembly problem to finding a maximum matching in a weighted bipartite graph that represents compatibilities[23] among fragments. Non-coding RNAs[22] and microRNAs[10] have been reported to regulate cell differentiation and development, and coding genes are known to produce noncoding isoforms as a means of regulating protein levels through nonsense-mediated decay[32]. For these biologically motivated reasons, the assembler does not require that assembled transcripts contain an open reading frame. Since Cufflinks does not make use of existing gene annotations during assembly, we validated the transcripts by first comparing individual time point assemblies to existing annotations.

Cufflinks takes as input alignments of RNA-Seq fragments to a reference

This chapter discusses a transcript assembly algorithm and is joint work with Lior Pachter and Steven Salzberg. Geo Pertea wrote Cuffcompare, described in Section 4.4, which is included with the assembler. The validation of this assembler is joint work with Geo Pertea, Ali Mortazavi, Brian Williams, Marijke J. van Baren, and Barbara Wold.

genome and, in the absence of an (optional) user provided annotation, initially assembles transcripts from the alignments. Transcripts in each of the loci are assembled independently. The assembly algorithm is designed to aim for the following:

1. Every fragment is consistent with at least one assembled transcript.

2. Every transcript is tiled by reads.

3. The number of transcripts is the smallest required to satisfy requirement (1).

4. The resulting RNA-Seq models (in the sense of Section 3.3) are identifiable.

In other words, we seek an assembly that parsimoniously explains the fragments from the RNA-Seq experiment; every fragment in the experiment (except those filtered out during a preliminary error-control step) should have come from a Cufflinks transcript, and Cufflinks should produce as few transcripts as possible with that property. Thus, Cufflinks seeks to optimize the criterion suggested in[70], however, unlike the method in that paper, Cufflinks leverages Dilworth's Theorem[14] to solve the problem by reducing it to a matching problem via the equivalence of Dilworth's and König's theorems (Theorem 19 in Appendix A).

4.1 A partial order on fragment alignments

The Cufflinks program loads a set of alignments in SAM format sorted by reference position and assembles non-overlapping sets of alignments independently. After filtering out any erroneous spliced alignments or reads from incompletely spliced RNAs, Cufflinks constructs a partial order (Definition 16), or equivalently a directed

acyclic graph (DAG), with one node for each fragment that in turn consists of an aligned pair of mated reads. First, we note that fragment alignments are of two types: those where reads align in their entirety to the genome, and reads which have a split alignment (due to an implied intron).

In the case of single reads, the partial order can be simply constructed by checking the reads for *compatibility*. Two reads are *compatible* if their overlap contains the exact same implied introns (or none). If two reads are not compatible they are *incompatible*. The reads can be partially ordered by defining, for two reads x, y, that $x \leq y$ if the starting coordinate of x is at or before the starting coordinate of y, and if they are compatible.

In the case of paired-end RNA-Seq the situation is more complicated because the unknown sequence between mate pairs. To understand this, we first note that pairs of fragments can still be determined to be incompatible if they cannot have originated from the same transcript. As with single reads, this happens when there is disagreement on implied introns in the overlap. However compatibility is more subtle. We would like to define a pair of fragments x, y to be compatible if they do not overlap, or if every implied intron in one fragment overlaps an identical implied intron in the other fragment.

However it is important to note that it may be impossible to determine the compatibility (as defined above) or incompatibility of a pair of fragments. For example, an unknown region internal to a fragment may overlap two different introns (that are incompatible with each other). The fragment may be compatible with one of the introns (and the fragment from which it originates) in which case it is

incompatible with the other. Since the opposite situation is also feasible, compatibility (or incompatibility) cannot be assigned. Fragments for which the compatibility/incompatibility cannot be determined with respect to every other fragment are called *uncertain*. Finally, two fragments are called *nested* if one is contained within the other.

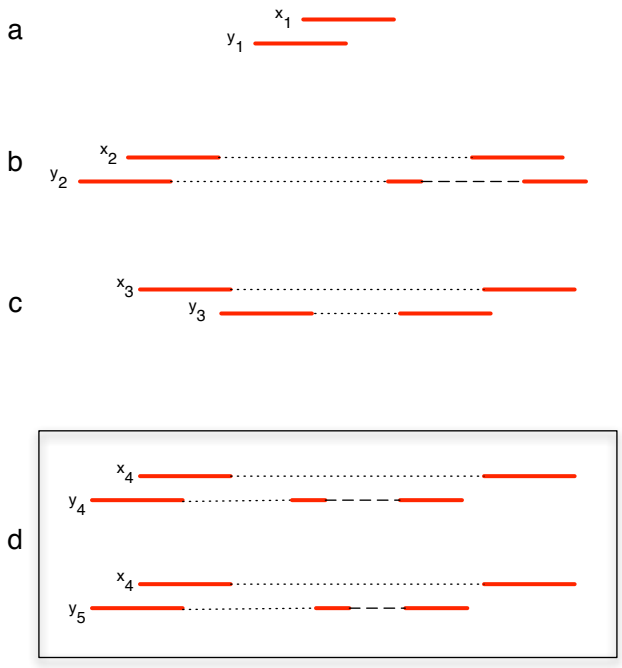


Figure 4.1: Compatibility and incompatibility of fragments. End-reads are solid lines, unknown sequences within fragments are shown by dotted lines and implied introns are dashed lines. The reads in (a) are compatible, whereas the fragments in (b) are incompatible. The fragments in (c) are nested. Fragment x_4 in (d) is uncertain, because y_4 and y_5 are incompatible with each other.

Before constructing a partial order, fragments are extended to include their nested fragments and uncertain fragments are discarded. These discarded fragments are used in the abundance estimation. In theory, this may result in suboptimal (i.e. non-minimal assemblies) but we determined empirically that after assembly

uncertain fragments are almost always consistent with one of the transcripts. When they are not, there was no completely tiled transcript that contained them. Thus, we employ a heuristic that significantly speeds up the program, and that works in practice.

A partial order P is then constructed from the remaining fragments by declaring that $x \leq y$ whenever the fragment corresponding to x begins at, or before, the location of the fragment corresponding to y and x and y are compatible. In what follows we identify P with its Hasse diagram (or covering relation), equivalently a directed acyclic graph (DAG) that is the transitive reduction.

Proposition 4. *P is a partial order.*

Proof: The fragments can be totally ordered according to the locations where they begin. It therefore suffices to check that if x, y, z are fragments with x compatible with y and y compatible with z then x is compatible with z. Since x is not uncertain, it must be either compatible or incompatible with z. The latter case can occur only if x and/or z contain implied introns that overlap and are not identical. Since y is not nested within z and x is not nested within y, it must be that y contains an implied intron that is not identical with an implied intron in either x or z. Therefore y cannot be compatible with both x and z. □

4.2 Assembling a parsimonious set of transcripts

In order to assemble a set of transcripts, Cufflinks finds a (minimum) partition of P into chains (see Definition 16). A partition of P into chains yields an assembly

because every chain is a totally ordered set of compatible fragments $x_1, \ldots, x_l$ and therefore there is a set of overlapping fragments that connects them. By Dilworth's theorem (Theorem 17), the problem of finding a minimum partition P into chains is equivalent to finding a maximum antichain in P (an antichain is a set of mutually incompatible fragments). Subsequently, by Theorem 19, the problem of finding a maximum antichain in P can be reduced to finding a maximum matching in a certain bipartite graph that emerges naturally in deducing Dilworth's theorem from König's theorem 18. We call the key bipartite graph the "reachability" graph. It is the transitive closure of the DAG, i.e. it is the graph where each fragment x has nodes L_x and R_x in the left and right partitions of the reachability graph respectively, and where there is an edge between L_x and R_y when $x \leq y$ in P. The maximum matching problem is a classic problem that admits a polynomial time algorithm. The Hopcroft-Karp algorithm[27] has a run time of $O(\sqrt{V}E)$ where in our case V is the number of fragments and E depends on the extent of overlap, but is bounded by a constant times the coverage depth. We note that our parsimony approach to assembly therefore has a better complexity than the $O(V^3)$ PASA algorithm[23].

The minimum cardinality chain decomposition computed using the approach above may not be unique. For example, a locus may contain two putative distinct initial exons (defined by overlapping incompatible fragments), and one of two distinct terminal and a constitutive exon in between that is longer than any read or insert in the RNA-Seq experiment. In such a case, the parsimonious assembly will consist of two transcripts, but there are four possible solutions that are all minimal. In order to "phase" distant exons, we leverage the fact that abundance inhomo-

geneities can link distant exons via their coverage. We therefore weight the edges of the bipartite reachability graph based on the percent-spliced-in metric introduced by Wang *et al.*[68]. In our setting, the percent-spliced-in ψ_x for an alignment x is computed by counting the alignments overlapping x in the genome that are compatible with x and dividing by the total number of alignments that overlap x, and normalizing for the length of the x. The cost $C(y, z)$ assigned to an edge between alignments y and z reflects the belief that they originate from different transcripts:

$$C(y, z) = -\log(1 - |\psi_y - \psi_z|). \tag{4.1}$$

Rather than using the Hopcroft-Karp algorithm, a modified version of the `LEMON` (http://lemon.cs.elte.hu/trac/lemon) and `Boost` (http://www.boost.org) graph libraries are used to compute a *min-cost* maximum cardinality matching on the bipartite compatibility graph. Even with the presence of weighted edges, our algorithm is very fast. The best known algorithm for weighted matching is $O(V^2 logV + VE)$.

Some transcripts in a sample may be present at very low relative abundance, and may not be sequenced at sufficient depth to be fully covered by reads. Cufflinks will thus only report the parts of each transcript covered by reads, or "transfrags". Because we isolated total RNA, we expected that a small fraction of our reads would come from the intronic regions of incompletely processed primary transcripts. Moreover, transcribed repetitive elements and low-complexity sequence result in "shadow" transfrags that we wished to discard as artifacts. Thus, Cufflinks heuristically identifies artifact transfrags and suppresses them in its output. We also

filter extremely low-abundance minor isoforms of alternatively spliced genes, using the model described in Chapter 3 as a means of reducing the variance of estimates for more abundant transcripts. A transcript x meeting any of the following criteria is suppressed:

1. x aligns to the genome entirely within an intronic region of the alignment for a transcript y, and the abundance of x is less than 15% of y's abundance.

2. x is supported by only a single fragment alignment to the genome.

3. More than 75% of the fragment alignments supporting x, are mappable to multiple genomic loci.

4. x is an isoform of an alternatively spliced gene, and has an estimated abundance less than 5% of the major isoform of the gene.

Prior to transcript assembly, Cufflinks also filters out some of the alignments for fragments that are likely to originate from incompletely spliced nuclear RNA, as these can reduce the accuracy abundance estimates for fully spliced mRNAs. These filters and the output filters above are detailed in the source file `filters.cpp` of the source code for Cufflinks.

In the overview of this Section, we mentioned that our assembly algorithm has the property that the resulting models are identifiable. This is a convenient property that emerges naturally from the parsimony criterion for a "minimal explanation" of the fragment alignments. Formally, it is a corollary of Dilworth's theorem:

Proposition 5. *The assembly produced by the Cufflinks algorithm always results in an identifiable RNA-Seq model.*

Proof: By Dilworth's theorem, the minimum chain decomposition (assembly) we obtain has the same size as the maximum antichain in the partially ordered set we construct from the reads. An antichain consists of reads that are pairwise incompatible, and therefore those reads must form a permutation sub-matrix in the fragment-transcript matrix $A_{R,T}$ with columns corresponding to the transcripts in a locus, and with rows corresponding to the fragments in the antichain. The matrix $A_{R,T}$ therefore contains permutation sub-matrices that together span all the columns, and the matrix is full-rank.

4.3 The myogenic transcriptome

We recovered a total of 13,689 known isoforms from 10,372 genes, and 12,712 new isoforms of known genes. We estimate that 77% of the reads originated from previously known transcripts (Table 4.2). Of the new isoforms, 7,395 (58%) contain novel splice junctions, with the remainder being novel combinations of known splicing outcomes. 11,712 (92%) have an open reading frame (ORF), 8,752 of which end at an annotated stop codon. Although we sequenced deeply by current standards, at least 80% of the detected transcripts were recovered with a single lane of GAII transcriptome sequencing. Because distinguishing a full-length transcript from a partially assembled fragment is difficult, we conservatively excluded novel isoforms that were unique to a single time point from further analyses. Out of the

new isoforms, 3,724 were present in multiple time points, and 581 were present at all time points. 6,518 (51%) of the new isoforms and 2,316 (62%) of the multiple time point novel isoforms were tiled by high-identity EST alignments or matched RefSeq isoforms from other organisms, and endpoint RT-PCR experiments confirmed new isoforms in genes of interest (Table 4.4). We concluded that a majority of the unannotated transcripts we found are in the myogenic transcriptome, and that the mouse annotation remains incomplete.

4.4 Assessment of assembly quality

To compare Cufflinks transfrags against annotated transcriptomes, and also to find transfrags common to multiple assemblies, we developed a tool called Cuffcompare that builds structural equivalence classes of transcripts. We ran Cuffcompare on each the assembly from each time point against the combined annotated transcriptomes of the UCSC known genes, `Ensembl`, and `Vega`. Because of the stochastic nature of sequencing, *ab initio* assembly of the same transcript in two different samples may result in transfrags of slightly different lengths. A Cufflinks transfrag was considered a complete match when there was a transcript with an identical chain of introns in the combined annotation.

When no complete match is found between a Cufflinks transfrag and the transcripts in the combined annotation, Cuffcompare determines and reports if another potentially significant relationship exists with any of the annotation transcripts that can be found in or around the same genomic locus. For example, when all the introns

of a transfrag match perfectly a part of the intron chain (sub-chain) of an annotation transcript, a "containment" relationship is reported. For single-exon transfrags, containment is also reported when the exon appears fully overlapped by any of the exons of an annotation transcript. If there is no perfect match for the intron chain of a transfrag but only some exons overlap and there is at least one intron-exon junction match, Cuffcompare classifies the transfrag as a putative "novel" isoform of an annotated gene. When a transfrag is unspliced (single-exon) and it overlaps the intronic genomic space of a reference annotation transcript, the transfrag is classified as potential pre-mRNA fragment. Finally, when no other relationship is found between a Cufflinks transfrag and an annotation transcript, Cuffcompare can check the repeat content of the transfrag's genomic region (assuming the soft-masked genomic sequence was also provided) and it would classify the transfrag as "repeat" if most of its bases are found to be repeat-masked.

When provided multiple time point assemblies, Cuffcompare matches transcripts between samples that have an identical intron structure, placing all mutually matching transcripts in the same equivalence class. The program reports a non-redundant set of transcript structures, choosing the longest transcript from each equivalence class as the representative transcript. Cuffcompare also reports the relationships found between each equivalence class (transcripts that have a complete match across time points) and reference transcripts from the combined annotation set, where applicable.

Table 4.2 includes the classifications of the transfrags reported by Cufflinks after assembling the C2C12 reads. While only 13.5% of assembled transfrags repre-

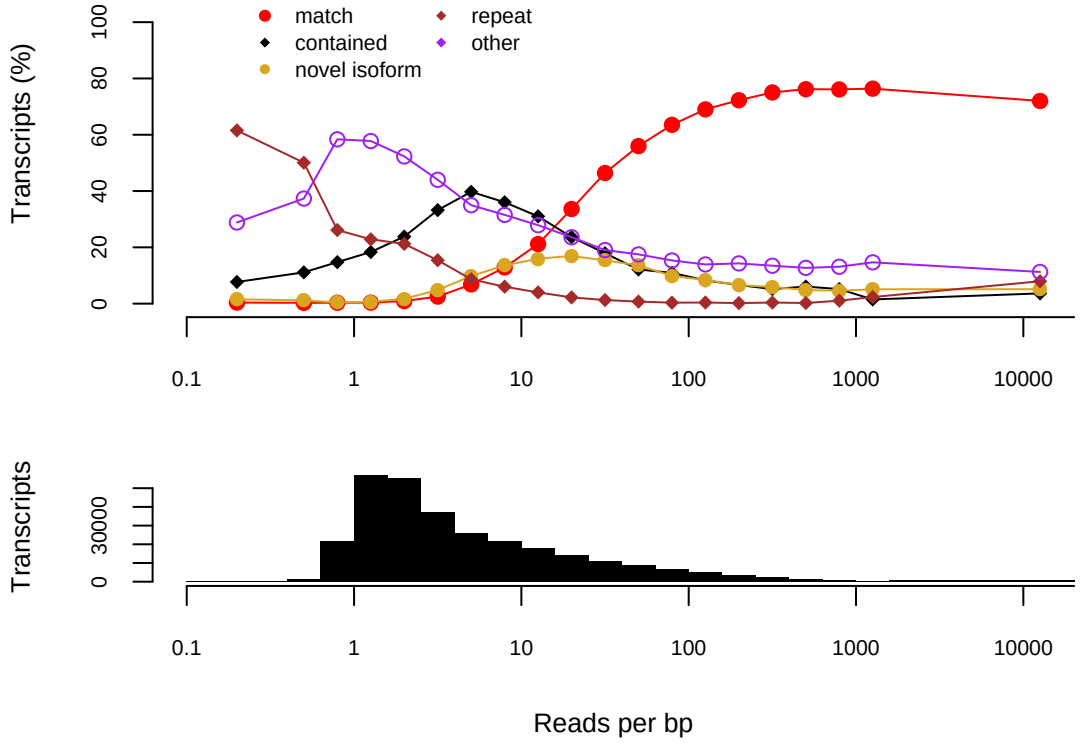


Figure 4.2: Categorization of Cufflinks transcripts by estimated depth of read coverage.

sent known transcripts, Cufflinks assigns more than 76% of reads to these, reflecting the fact that moderate and highly-abundant transfrags generate most of the library fragments in the experiment. Less abundant transcripts receive less complete sequencing coverage, resulting in numerous transfrags that partially but compatibly match known transcripts. Figure 4.2 shows the categories of Cufflinks transfrags as estimated depth of sequencing coverage increasing.

We selected the Cufflinks transfrags that did not have a complete match or "containment" relationship with a known annotation transcript, but were classified by Cuffcompare as putative "novel isoforms" of known genes. We explored the

Category	**Transcripts** (%)	**Assembled reads** (%)
Match to known isoform	13.5	76.7
Novel isoform of known gene	6.3	11.3
Contained in known isoform	24.1	4.6
Repeat	14.2	0.6
Intronic	11.1	0.6
Polymerase run-on	6.3	0.5
Intergenic	16.5	1.2
Other artifacts	7.7	4.5

Table 4.2: Types of predicted transcripts.

sequence similarity between these transfrags and two sets of mRNA sequences: one set representing the mouse transcriptome and consisting of all mouse ESTs in dbEST plus all reviewed or validated RefSeq mouse mRNAs, and the other consisting of all reviewed or validated RefSeq mRNAs from other mammalian species.

We used megablast to map all mouse ESTs onto this set of Cufflinks transfrags, only keeping EST alignments where at least 80% of the EST length was aligned with at least 95% identity. We calculated transfrag coverage by tiling overlapping EST mappings on each transfrag and counted only those transfrags that are covered by ESTs for at least 80% of the transfrag length without any coverage gaps, and with coverage discontinuities only allowed at no more than 10% distance from either end. For the mouse mRNAs alignments we also used megablast with the same basic coverage cutoffs (minimum 80% covered with no more than 10% unaligned on either side of the overlap) but applied to each pairwise alignment independently (i.e. as opposed to EST alignments, no coverage tiling was considered for mRNA

alignments). For alignments with the non-mouse mRNAs we used discontiguous megablast with a dual (combined) discontiguous word template (option -N 2), with the same coverage assessment protocol as in the case of mouse mRNA alignments but with the percent identity cutoff lowered to 80%.

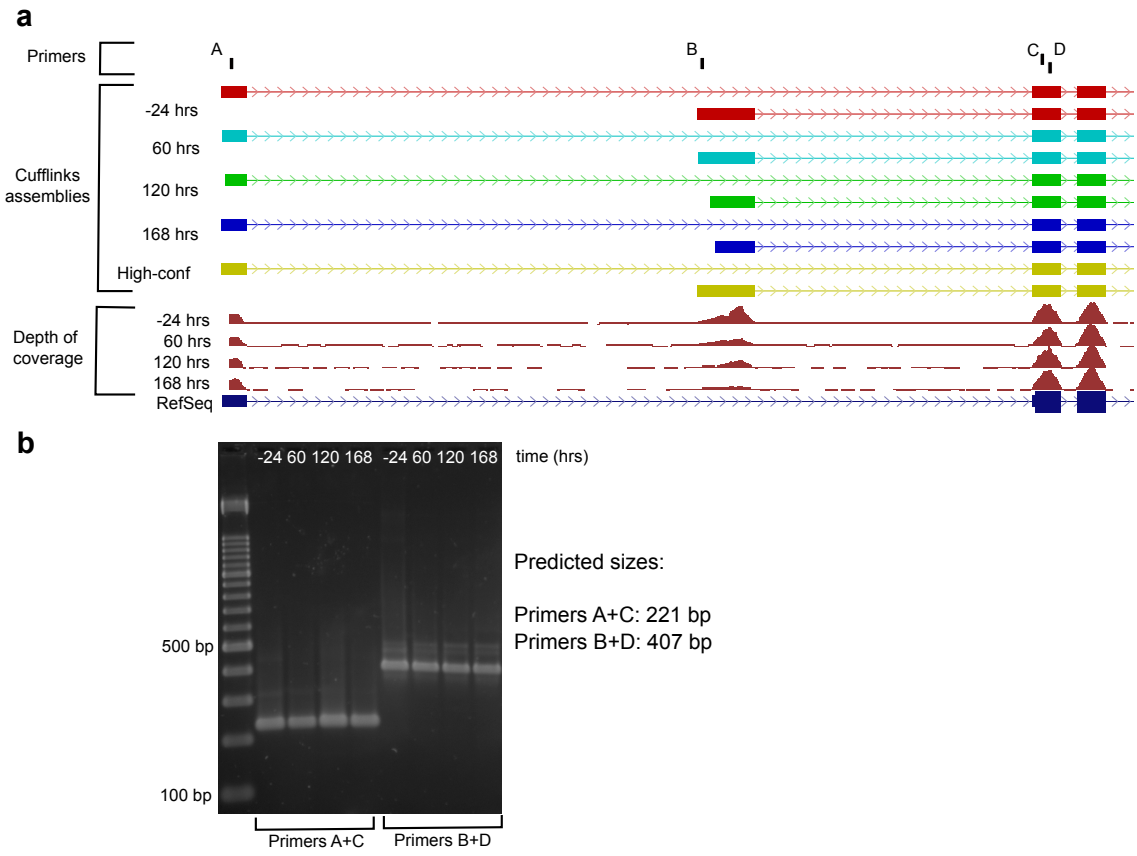


Figure 4.3: New and known isoforms of Fhl3 recovered by Cufflinks at each time point (a) were confirmed by form-specific RT-PCR (b).

4.5 Validation of novel transfrags

We selected the Cufflinks transfrags that did not have a complete match or "containment" relationship with a known annotation transcript, but were classified

by Cuffcompare as putative "novel isoforms" of known genes. We explored the sequence similarity between these transfrags and two sets of mRNA sequences: one set representing the mouse transcriptome and consisting of all mouse ESTs in dbEST plus all reviewed or validated RefSeq mouse mRNAs, and the other consisting of all reviewed or validated RefSeq mRNAs from other mammalian species.

We used megablast to map all mouse ESTs onto this set of Cufflinks transfrags, only keeping EST alignments where at least 80% of the EST length was aligned with at least 95% identity. We calculated transfrag coverage by tiling overlapping EST mappings on each transfrag and counted only those transfrags that are covered by ESTs for at least 80% of the transfrag length without any coverage gaps, and with coverage discontinuities only allowed at no more than 10% distance from either end. For the mouse mRNAs alignments we also used megablast with the same basic coverage cutoffs (minimum 80% covered with no more than 10% unaligned on either side of the overlap) but applied to each pairwise alignment independently (i.e. as opposed to EST alignments, no coverage tiling was considered for mRNA alignments). For alignments with the non-mouse mRNAs we used discontiguous megablast with a dual (combined) discontiguous word template (option -N 2), with the same coverage assessment protocol as in the case of mouse mRNA alignments but with the percent identity cutoff lowered to 80%.

4.6 Library complexity measurements, assembly accessibility

To assess the dependence of assembly quality on the depth of sequencing, we mapped and assembled subsets of our reads at the 60 hour time point. We partitioned the three Illumina lanes' worth of data (a total of 140 million reads) into 64 subsets. We then processed a single subset with TopHat and Cufflinks, as above, and compared the resulting transfrags to the output of Cufflinks on all three lanes using Cuffcompare. We repeated the mapping and assembly with two subsets, four subsets, eight, and so on. Figure 4.5 shows the fraction of reference transcripts captured by Cufflinks using all three lanes that are still captured when less data is available. For transcripts with extremely low abundance (<5 FPKM), increased sequencing yields more full-length transcripts. However, for even moderately abundant transcripts (≥5 FPKM), nearly 80% or more of the transcripts are recovered with only 40 million reads, or a lane's worth of Illumina GA II sequencing.

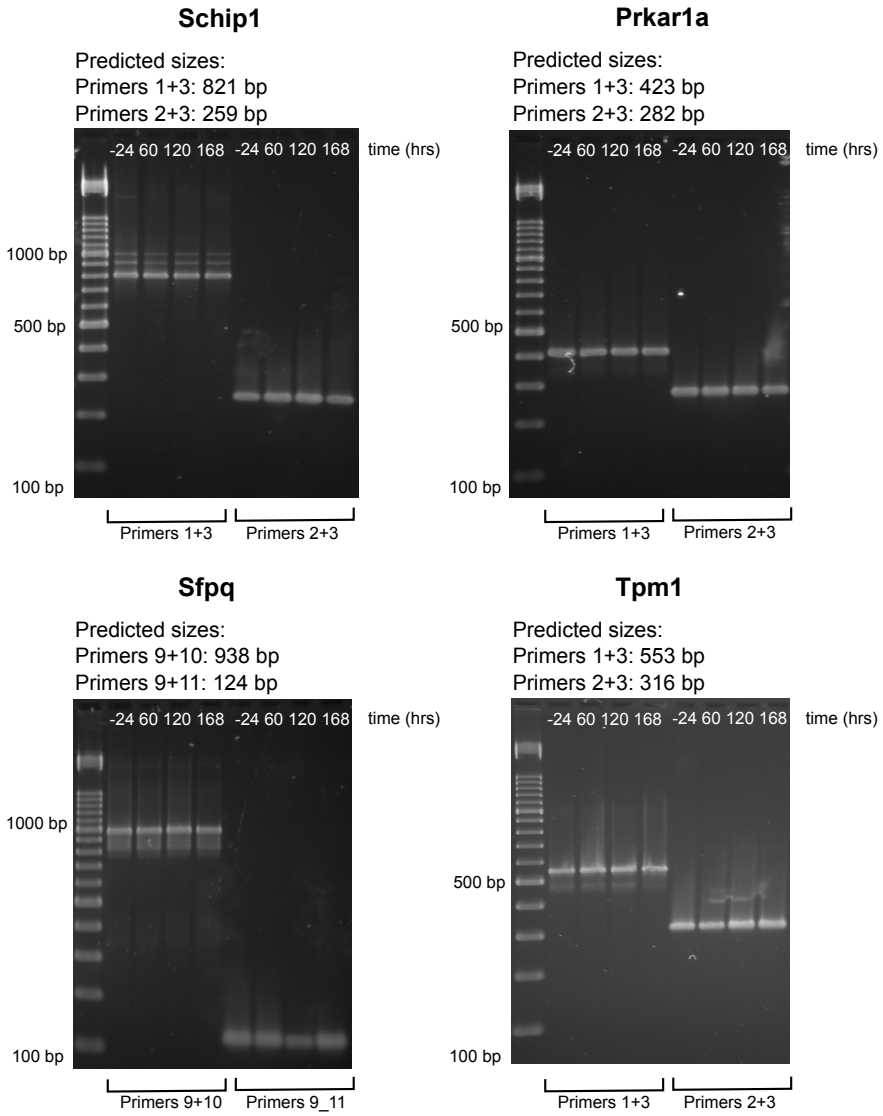
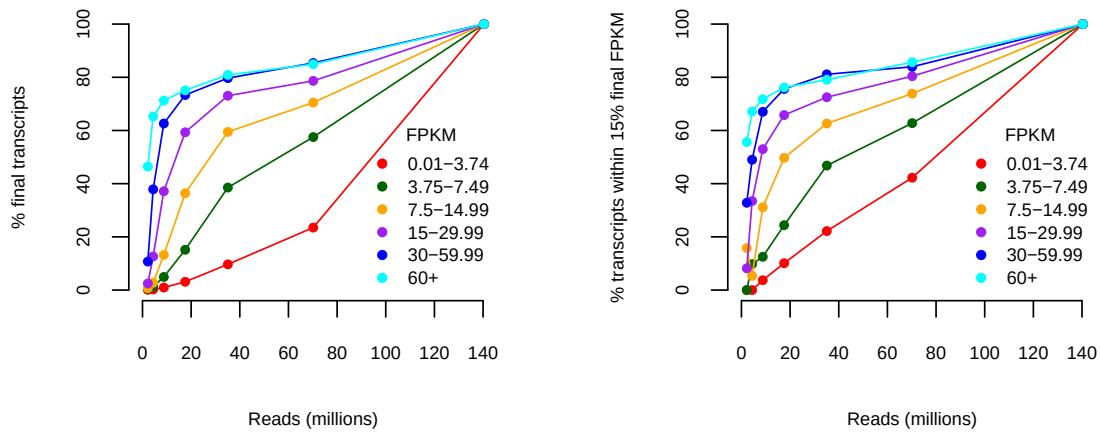


Figure 4.4: RT-PCR of selected genes. For Schip1, Cufflinks assembled a known and a novel isoform (with a new TSS), both of which are detected by RT-PCR. Prkar1a is annotated with two alternate first exons and start sites in `UCSC known genes`, both of which were detected. Cufflinks assembles the known isoform of the splicing factor Sfpq, along with a novel variant that contains most of RIKEN clone. Tpm1, a gene known to have muscle- and non-muscle-specific isoforms displays previously observed alternative first and last exons.

Primer name	sequence	product length	endpoint gel score
FHL3 Ex1Ex3			
Left	CTCGCCGCTGCTCTCTCG	221	+++
Right	GTGTTGTCATAGCACGGAACG		
FHL3 Ex2Ex3			
Left	AGGAAGGGCTCACAAGTGG	407	+++
Right	ATAGCACGGAACGCAGTAGG		
Sfpq Ex9Ex10			
Left	GTGGTGGCATAGGTTATGAAGC	936	+++
Right	CCATTTTCAAAAGCTTTCAAGG		
Sfpq Ex9Ex11			
Left	GTGGTGGCATAGGTTATGAAGC	172	+++
Right	CTCAAGTAAATAAGACTCCAAAATCAGC		
Prkar1aEx1Ex3			
Left	ACAGCAGGGATCTCCTTGTCC	418	+++
Right	CCTCTCAAAGTATTCCCGAAGG		
Prkar1aEx2Ex3			
Left	GCTATCGCAGAGTGGTAGTGAGG	279	+++
Right	CCTCTCAAAGTATTCCCGAAGG		
Schip1Ex1Ex3			
Left	GGCTATGAGGGTGAAAAGTGC	1050	+++
Right	GTATAGATTCCTGGGCCATCG		
Schip1Ex2Ex3			
Left	CAGCATGAGTGGTAACCAAGG	269	+++
Right	GTATAGATTCCTGGGCCATCG		
Tpm1Ex1Ex3			
Left	TGAACAAAAGACCCCAGAGG	565	+++
Right	CTGAAGTACAAGGCCATCAGC		
Tpm1Ex2Ex3			
Left	AGTTTTATTGAGCGTTGAGACG	318	+++
Right	CTGAAGTACAAGGCCATCAGC		

Table 4.4: Form-specific RT-PCR primers for selected genes, designed with Primer3 (http://frodo.wi.mit.edu/primer3/).



(a) Robustness of assembly as depth of sequencing varies

(b) Robustness of quantitation as depth of sequencing varies

Figure 4.5: Robustness of assembly and abundance estimation as a function of expression level and depth of sequencing. Subsets of the full 60-hour read set were mapped and assembled with TopHat and Cufflinks and the resulting assemblies were compared for structural and abundance agreement with the full 60 hour assembly. Colored lines show the results obtained at different depths of sequencing in the full assembly; e.g., the light blue line tracks the performance for transcripts with FPKM greater than 60. (a) The fraction of transcript fragments fully recovered increases with additional sequencing data, though nearly 80% of moderately expressed (15 FPKM) are recovered with less than 40 million 75bp paired-end reads, a fraction of the data generated by a single run of the sequencer used in this experiment. (b) Abundance estimates are similarly robust. At 40 million reads, transcripts determined to be moderately expressed using all 60 hour reads were estimated at within 15% of their final FPKM values.

Chapter 5

Differential transcription and regulation

In order to explore expression dynamics in the myogenic transcriptome, we developed tests for statistically significant changes in transcript- and gene-level expression as determined by our model, and used these to characterize the "trajectories" of the RNAs across the time course. These tests are detailed below, followed by the results of testing on the C2C12 experiment.

5.1 Expression curve shape assignment

Between any two consecutive time points, we tested whether a transcript was significantly (after FDR control) up or down regulated (or flat). This was done using the following testing procedure for absolute differential expression:

In order to test for differential transcription, we employ the standard method used in microarray-based expression analysis and proposed for RNA-Seq[5], which is to compute the logarithm of the ratio of intensities (in our case FPKM), and then use the delta method to estimate the variance of the log odds. We describe this for testing differential transcription of individual transcripts and also groups of

This chapter describes an approach to differential analysis of expression and regulation from RNA-Seq, and is joint work with Lior Pachter. We perform this analysis on differentiating myoblasts in an experiment conducted by Brian Williams, Ali Mortazavi, Gordon Kwan, and Barbara Wold.

transcripts (e.g. grouped by TSS).

We recall that the MLE FPKM for a transcript $t \in g$ is given by

$$\frac{10^9 X_g \hat{\gamma}_t}{\tilde{l}(t) M}. \tag{5.1}$$

Given two different experiments resulting in X_g^a, M^a and X_g^b, M^b respectively, as well as $\hat{\gamma}_t^a$ and $\hat{\gamma}_t^b$, we would like to test the significance of departures from unity of the ratio of MLE FPKMS, i.e.

$$\left(\frac{10^9 X_g^a \hat{\gamma}_t^a}{\tilde{l}(t) M^a}\right) / \left(\frac{10^9 X_g^b \hat{\gamma}_t^b}{\tilde{l}(t) M^b}\right) \tag{5.2}$$

$$= \frac{X_g^a \hat{\gamma}_t^a M^b}{X_g^b \hat{\gamma}_t^b M^a}. \tag{5.3}$$

This can be turned into a test statistic that is approximately normal by taking the logarithm, and normalizing by the variance. We recall that using the delta method, if X is a random variable then $Var[\log(X)] \approx \frac{Var[X]}{E[X]^2}$.

Therefore, our test statistic is

$$\frac{\log(X_g^a) + \log(\hat{\gamma}_t^a) + \log(M^b) - \log(X_g^b) - \log(\hat{\gamma}_t^b) - \log(M^a)}{\sqrt{\frac{\left(\Psi_{t,t}^{g,a}(1+X_g^a)+(\hat{\gamma}_t^a)^2\right)}{X_g^a(\hat{\gamma}_t^a)^2} + \frac{\left(\Psi_{t,t}^{g,b}(1+X_g^b)+(\hat{\gamma}_t^b)^2\right)}{X_g^b\left(\hat{\gamma}_t^b\right)^2}}}. \tag{5.4}$$

5.2 Quantifying transcriptional and post-transcriptional overloading

There are two biologically meaningful groupings of transcripts whose relative abundances are interesting to track in a time course. Transcripts that share a TSS are likely to be regulated by the same promoter, and therefore tracking the change in relative abundances of groups of transcripts sharing a TSS may reveal how transcriptional regulation is affecting expression over time. Similarly, transcripts

that share a TSS and exhibit changes in expression relative to each other are likely to be affected by splicing or other post-transcriptional regulation. We therefore grouped transcripts by TSS and compared relative abundance changes within and between groups.

We define "overloading" to be a significant change in relative abundances for a set of transcripts (as determined by the Jensen-Shannon metric, see below). The term is intended to generalize the simple notion of "isoform switching" that is well-defined in the case of two transcripts, to multiple transcripts. It is complementary to absolute differential changes in expression: the overall expression of a gene may remain constant while individual transcripts change drastically in relative abundances resulting in overloading. The term is borrowed from computer science, where in some statically-typed programming languages, a function may be used in multiple, specialized instances via "method overloading".

In order to test for differential transcription of a group of transcripts, we replace the numerator and denominator above by those from Equations (3.36) and (3.38).

Given significantly differentially expressed isoforms, we defined the *shape* of a transcript's expression by the presence or absence of significant changes in expression between sequential time points. four shapes were considered: "non-decreasing", "non-increasing", "flat" (no significant changes), or a "mixed" pattern (a shift down followed by a shift up, or a shift up followed by a shift down). By shape classification, 1,634 of 3,975 (41.1%) alternatively transcribed genes featured expression overloading. A selection of overloaded genes are displayed in Figures 5.1 and 5.2.

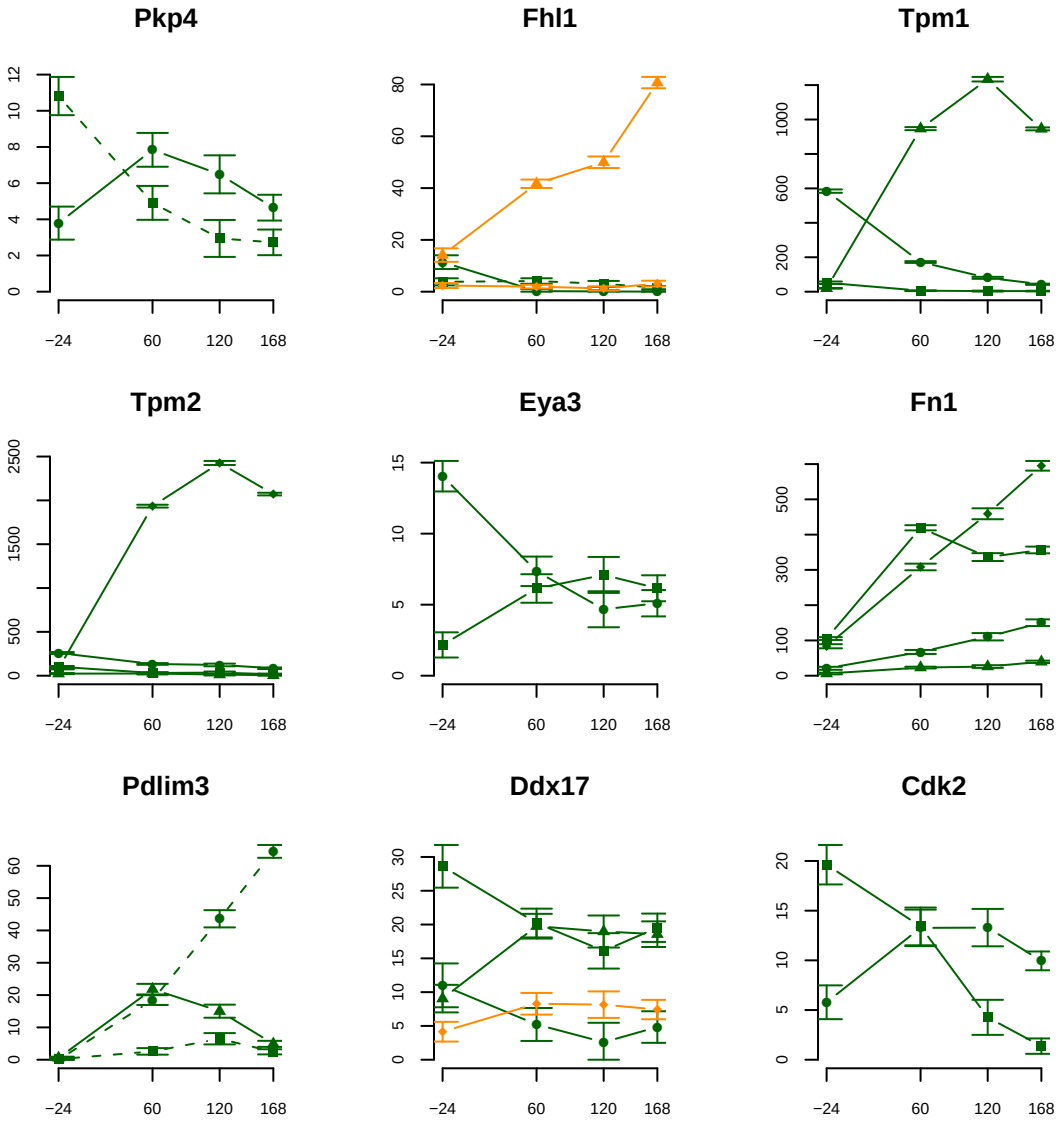


Figure 5.1: Selected genes with post-transcriptional overloading. Trajectories indicate the expression of individual isoforms in FPKM (y axis) over time in hours (x axis). Dashed isoforms have not been previously annotated. Isoform trajectories are colored by TSS, so isoforms with the same color presumably share a common promoter and are processed from the same primary transcript.

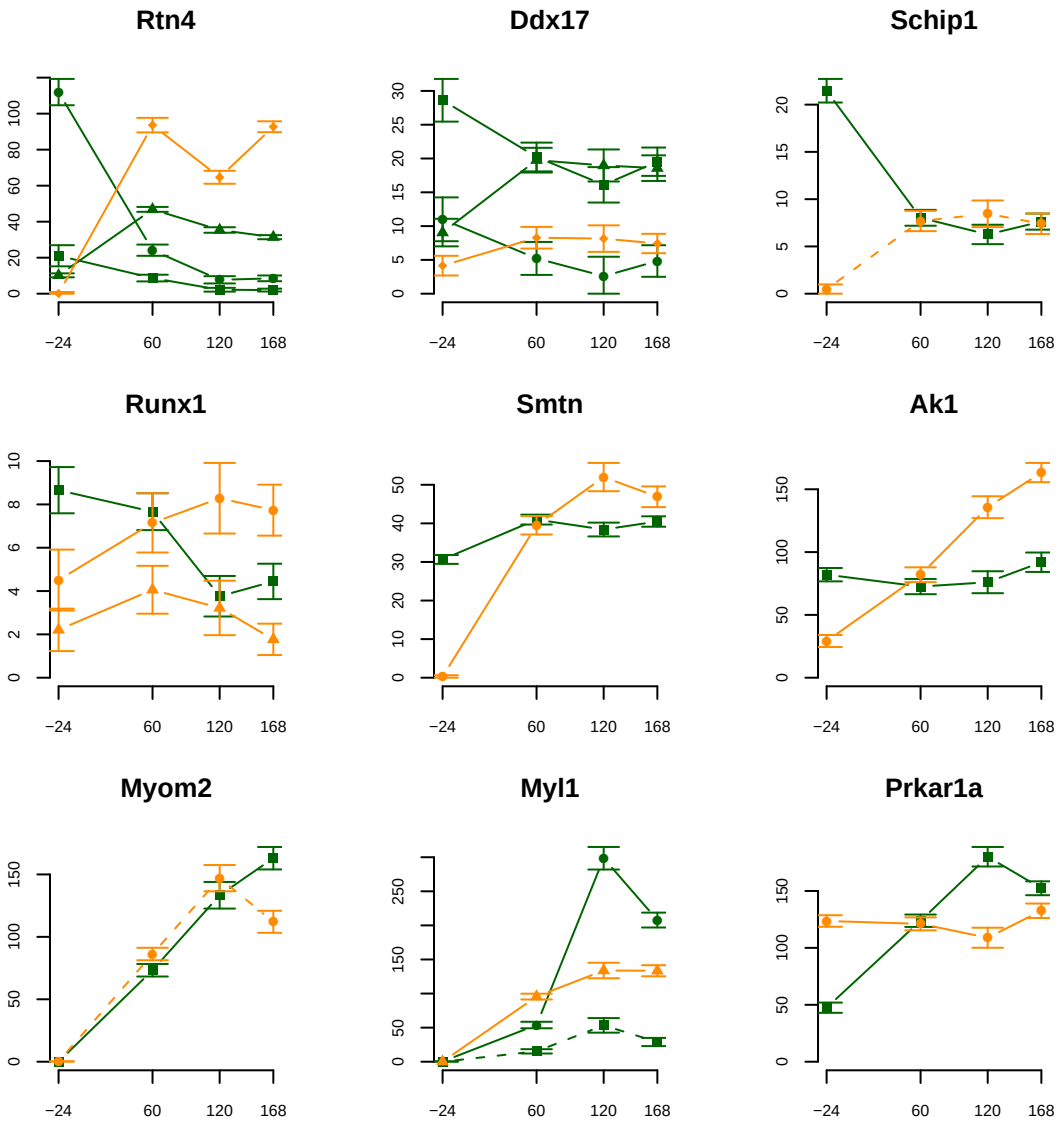


Figure 5.2: Selected genes with transcriptional overloading. Trajectories indicate the expression of individual isoforms in FPKM (y axis) over time in hours (x axis). Dashed isoforms have not been previously annotated. Isoform trajectories are colored by TSS, so isoforms with the same color presumably share a common promoter and are processed from the same primary transcript.

In order to infer the extent of differential promoter usage, we decided to measure changes in relative abundances of primary transcripts of single genes. Similarly, we investigated changes in relative abundances of transcripts grouped by TSS in order to infer differential splicing. These inferences required two ingredients:

1. A metric on probability distributions (derived from relative abundances).

2. A test statistic for assessing significant changes in differential promoter usage and splicing as measured using the metric referred to above.

In order to address the first requirement, namely a metric on probability distributions, we turned to an entropy-based metric. This was motivated by the methods in Ritchie *et al* where tests for differences in relative isoform abundances were performed to distinguish cancer cells from normal cells[54]. We extend this approach to be able to test for relative isoform abundance changes among multiple experiments in RNA-Seq.

Definition 6 (Entropy)**.** The entropy of a discrete probability distribution $p = (p_1, \ldots, p_n)$ $(0 \leq p_i \leq 1$ and $\sum_{i=1}^{n} p_i = 1)$ is

$$H(p) = -\sum_{i=1}^{n} p_i \log p_i. \tag{5.5}$$

If $p_i = 0$ for some i the value of $p_i \log p_i$ is taken to be 0.

Definition 7 (The Jensen-Shannon divergence)**.** The Jensen-Shannon divergence of m discrete probability distributions $p^1, \ldots, p^m$ is defined to be:

$$JS(p^1, \ldots, p^m) = H\left(\frac{p^1 + \cdots + p^m}{m}\right) - \frac{\sum_{j=1}^{m} H(p^j)}{m}. \tag{5.6}$$

In other words, the Jensen-Shannon divergence of a set of probability distributions is the entropy of their average minus the average of their entropies.

In the case where $m = 2$, we remark that the Jensen-Shannon divergence can also be described in terms of the Kullback-Leibler divergence of two discrete probability distributions. If we denote Kullback-Leibler divergence by

$$D(p^1 \| p^2) = \sum_i p_i^1 \log \frac{p_i^1}{p_i^2}, \tag{5.7}$$

then

$$JS(p^1, p^2) = \frac{1}{2} D(p^1 \| m) + \frac{1}{2} D(p^2 \| m) \tag{5.8}$$

where $m = \frac{1}{2}(p^1 + p^2)$. In other words the Jensen-Shannon divergence is a symmetrized variant of the Kullback-Leibler divergence.

The Jensen-Shannon divergence has a number of useful properties: for example it is symmetric and non-negative. However it is *not* a metric. The following theorem shows how to construct a metric from the Jensen-Shannon divergence:

Theorem 8 (Fuglede and Topsøe[19])**.** *The square root of the Jensen-Shannon divergence is a metric.*

The proof of this result is based on a harmonic analysis argument. We therefore call the square root of the Jensen-Shannon divergence the *Jensen-Shannon metric*. We employed this metric in order to quantify relative changes in expression in (groups of) transcripts.

In order to test for significance, we introduce a bit of notation. Suppose that S is a collection of transcripts (for example, they may share a common TSS). We

define

$$\kappa_t = \frac{\frac{\gamma_t}{\tilde{l}(t)}}{\sum_{u \in S} \frac{\gamma_u}{\tilde{l}(u)}} \tag{5.9}$$

to be the proportion of transcript t among all the transcripts in a group S. We let $Z = \sum_{u \in S} \hat{\gamma}_u / \tilde{l}(u)$ so that $\hat{\kappa}_t = \frac{\gamma_t}{\tilde{l}(t)Z}$. We therefore have that

$$Var[\hat{\kappa}_t] = \frac{Var[\hat{\gamma}_t]}{\tilde{l}(t)^2 Z^2}, \tag{5.10}$$

$$Cov[\hat{\kappa}_t, \hat{\kappa}_u] = \frac{Cov[\hat{\gamma}_t, \hat{\gamma}_u]}{\tilde{l}(t)\tilde{l}(u)Z^2}. \tag{5.11}$$

Our test statistic for divergent relative expression was the Jensen-Shannon metric. The test could be applied to multiple time points simultaneously, but we focused on pairwise tests (involving consecutive time points). Under the null hypothesis of no change in relative expression, the Jensen-Shannon metric should be zero. We tested for this using a one-sided t-test, based on an asymptotic derivation of the distribution of the Jensen-Shannon metric under the null hypothesis. This asymptotic distribution is normal by applying the delta method approximation, which involves computing the linear component of the Taylor expansion of the variance of $\sqrt{JS}$.

In order to simplify notation, we let $f(p^1, \ldots, p^m)$ be the Jensen-Shannon metric for m probability distributions $p^1, \ldots, p^m$.

Lemma 9. *The partial derivatives of the Jensen-Shannon metric are given by*

$$\frac{\partial f}{\partial p_l^k} = \frac{1}{2m\sqrt{f(p^1, \ldots, p^m)}} \log\left(\frac{p_l^k}{\frac{1}{m}\sum_{j=1}^m p_l^j}\right). \tag{5.12}$$

Let $\hat{\kappa}^1, \ldots, \hat{\kappa}^m$ denote m probability distributions on the set of transcripts S, for example the MLE for the transcript abundances in a time course. Then from the

delta method we have that $\sqrt{JS(\hat{\kappa}^1, \ldots, \hat{\kappa}^m)}$ is approximately normally distributed with variance given by

$$Var[\sqrt{JS(\hat{\kappa}^1, \ldots, \hat{\kappa}^m)}] \approx (\bigtriangledown f)^T \Sigma (\bigtriangledown f), \tag{5.13}$$

where Σ is the variance-covariance matrix for the $\kappa^1, \ldots, \kappa^m$, i.e., it is a block diagonal matrix where the ith block is the variance-covariance matrix for the κ_t^i given by Equations (5.10,5.11).

We tested for overloaded genes by performing a one-sided t-test based on the asymptotics of the Jensen-Shannon metric under the null hypothesis of no change in relative abundnaces of isoforms (either grouped by shared TSS for for post-transcriptional overloading, or by comparison of groups of isoforms with shared TSS for transcriptional overloading). Type I errors were controlled with the Benjamini-Hochberg[4] correction for multiple testing. A selection of overloaded genes are displayed in Figures 5.1 and 5.2.

We can visualize overloading and expression dynamics with a plot that superimposes transcriptional and post-transcriptional overloading and gene-level expression over the time course. We refer to these as "Minard plots", after Charles Joseph Minard's famous visualization of the progress of Napoleon's campaign against Russia in 1812[66]. An example for Myc is included in 3(c), and others are given in Appendix B. The dotted line indicates gene-level FPKM, with measured FPKM indicated by black circles. Grey circles indicate the arithmetic mean of gene-level FPKM between consecutive measured time points, interpolating FPKM at interme-

diate time points. The total gene expression overloading is visualized as a swatch centered around the interpolated expression curve. The width of the swatch encodes the amount of expression overloading between successive time points. The color of the swatch indicates the relative contributions of transcriptional and post-transcriptional expression overloading.

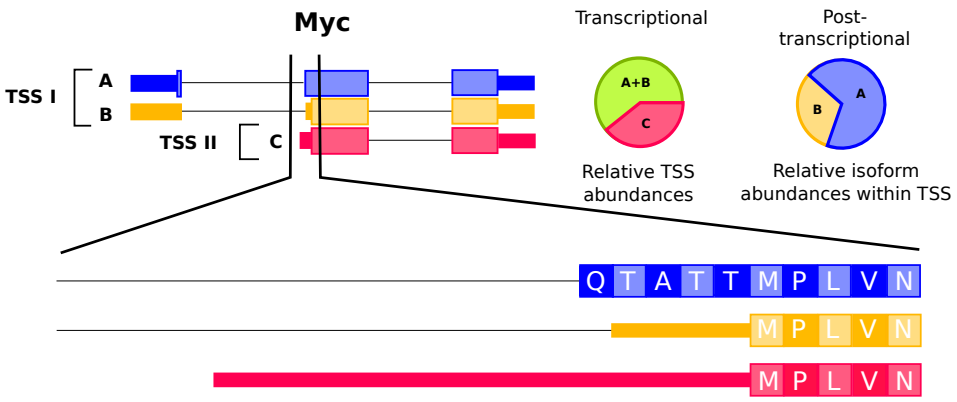
5.3 Differential expression and regulation in the myogenic transcriptome

For the purposes of estimating transcript abundances, we first selected a set of 17,416 high-confidence isoforms, 79% of which were previously known, from 11,079 loci. We identified the TSS for each transcript examined for transcriptional regulatory changes. This was complicated by the results of the simulation study that revealed that Cufflinks' estimates of the abundances of transcripts that have only been partially sequenced is less accurate than for those that have been completely covered and fully assembled. Thus, we restricted our analysis of expression dynamics over the time-course to a set of transcripts we believe are fully sequenced and correctly assembled, and we focused only on known and novel isoforms of annotated genes. This set consisted of transcripts that either were present in the `UCSC genome browser`, `Ensembl`, or `Vega` annotated transcriptomes, or were found in multiple C2C12 timepoint assemblies. We ignored transfrags classified as intronic pre-mRNA or polymerase run-on, as well as intergenic repeats to focus on coding genes and long non-coding RNAs. This high-confidence set contained a total of

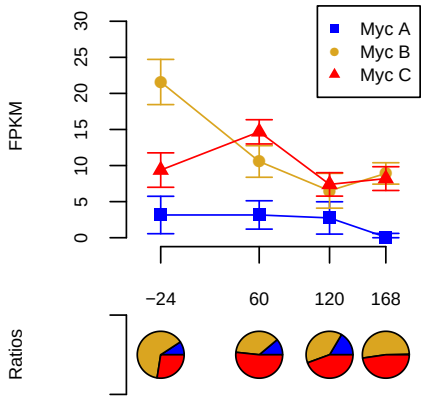
17,416 transcripts, 13,691 of which are in `UCSC known genes`, `Ensembl` or `VEGA` annotation and 3,724 of which are novel. Running Cufflinks' quantitation algorithm on this high-confidence set of transcripts at each time point allowed us to scan for differentially expressed transcripts, differentially spliced pre-mRNAs, and genes with shifts in promoter preference.

Cuffdiff identified 7,770 genes and 10,480 isoforms undergoing significant abundance changes between some successive pair of time points (FDR $< 5\%$). Many genes display substantial transcript-level dynamics that are not reflected in the summed patterns of expression for these genes. For example, Myc, a proto-oncogene which is known to be transcriptionally and post-transcriptionally regulated during myogenesis[15], is down-regulated overall during the time course, and while isoforms A and B follow this pattern, isoform C has a more complex and non-decreasing expression pattern. (Figure 3(b)) We noted that many genes displayed switches in major-minor transcripts, some containing isoforms with muscle-specific functions, such as tropomyosin I and II, which display a dramatic switch in isoform dominance upon differentiation (Appendix B). However, many genes featured dynamics involving several isoforms with behavior too complex to be deemed switching (see Figures 5.1 and 5.2 for selected examples). In light of these observations, we classified the types of expression dynamics for each transcript. Expression changes of a transcript between consecutive pairs of time points were classified increasing, decreasing, or flat based on the significance of changes in FPKM (FDR $<5\%$). Transcripts were then assigned one of four trajectories based on their expression curves being flat, increasing, decreasing or mixed (presence of both increases and decreases expression

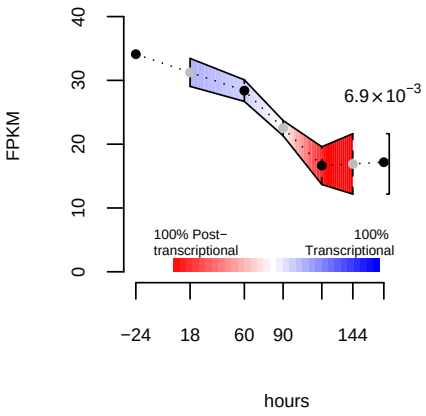
along the time course). In some statically-typed programming languages, a function may be used in multiple, specialized instances via "method overloading". Borrowing this terminology, we refer to a genes expression as "overloaded" when it has multiple isoforms that have different trajectories, possibly reflecting specialization of those isoforms. Expression overloading within a group of transcripts implies that the transcriptional and post-transcriptional machinery is regulating their output differently in two time points. Based on trajectory classification, a total of 1,634 genes were found to be overloaded in the time course, and we hypothesized that differential promoter preference and differential splicing were responsible for overloaded expression.



(a) The three isoforms of Myc



(b) Isoform expression dynamics

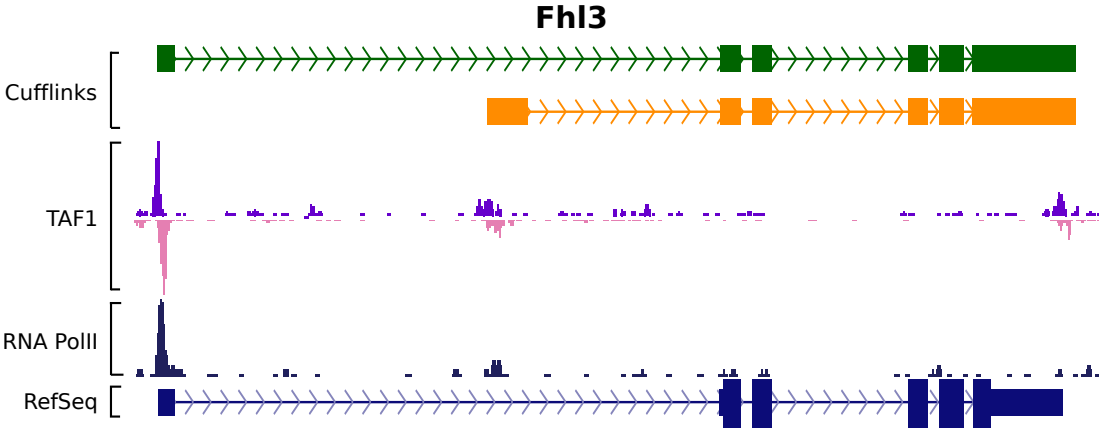


(c) Myc overloading

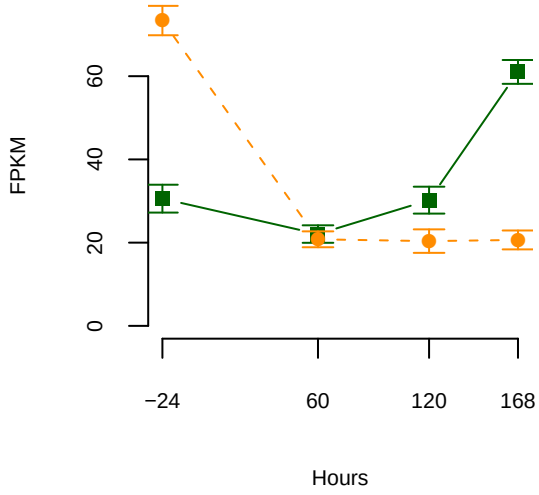
Figure 5.3: Distinction of transcriptional and post-transcriptional regulatory effects on overall transcript output. (a) When abundances of isoforms A, B, and C of Myc are grouped by TSS, changes in the relative abundances of the TSS groups indicate transcriptional regulation, where post-transcriptional effects are seen in changes in levels of isoforms of a single TSS group. (b) Individual isoforms of Myc have distinct expression dynamics. (c) Myc isoforms are overall downregulated as the timecourse proceeds. The width of the colored band is the measure of gene-expression overloading and the color is the log ratio of transcriptional and post-transcriptional overloading contributions (plot construction detailed in section 5.2).

To explore the impact of regulation on mRNA output and to check whether it could explain the variability of trajectories, we grouped transcripts by their start site (TSS) instead of just by gene. Changes in the relative abundances of mRNAs spliced from the same pre-mRNA transcript are by definition post-transcriptional, so this grouping effectively discriminates changes in mRNA output associated with differential transcription from changes associated with differential post-transcriptional processing. Of the 3,486 genes in our high confidence set with isoforms that shared a common TSS, 41% had TSS groups containing different isoform trajectories. Summing the expressions of isoforms sharing a TSS produces the trajectory for their primary transcript, and we identified 401 (48%) genes with multiple distinct primary transcript trajectories. However, measuring overloading based on trajectory classification was not precise enough to prioritize further investigation into individual genes and could not form the basis for statistical significance testing. We formalized and rigorously quantified overloading within and between TSS groups with an information-theoretic metric derived from the Jensen-Shannon divergence. With this metric, relative transcript abundances move in time along a logarithmic spiral in a real Hilbert space[19], and the distance moved measures the extent of expression overloading. Measuring overloading in this way revealed significant (FDR $< 5\%$) differential transcriptional regulation and splicing in 882 of 3,486 (25%) and 273 of 843 (32%) candidate genes respectively across the time course, with 70 genes displayed both types of overloading. Myc (Figures 3(a) and 3(b)) undergoes a shift in transcriptional regulation of transcript abundances to post-transcriptional control of abundances (Figure 3(a) and 3(b)) between 60 and 90 hours, as myocytes are

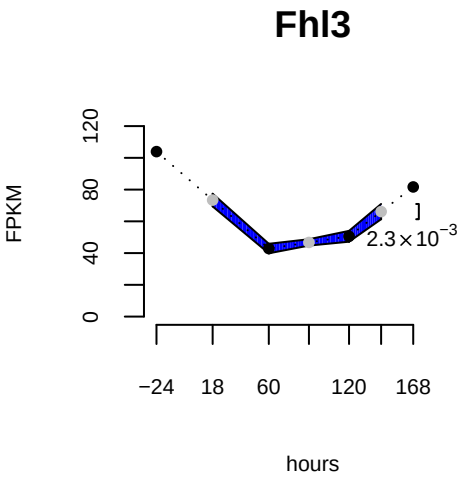
beginning to fuse into myotubes.



(a) A novel promoter for Fhl3



(b) Promoter switching in Fhl3



(c) Fhl3 overloading

Figure 5.4: Four-and-a-half-LIM domains 3 (Fhl3) inhibits myogenesis by binding MyoD and attenuating its transcriptional activity. (a))The C2C12 transcriptome contains a novel isoform that is dominant during proliferation. (b) The known isoform (solid line) is preferred at time points following differentiation. (c) Because FHL3 gives rise to two primary transcripts, but each is processed into a single mRNA, overloading is exclusively transcriptional.

Focusing on the significantly overloaded genes with promoter and isoform changes, we noted that in many cases changes in relative abundance reflected switch-like events in which there was an inversion of the dominant primary transcript. For example, in FHL3, a transcriptional regulator recently reported to negatively regulate myogenesis[11], Cufflinks assembled the known isoform and another with a novel start site (Figure 4(a)). We validated the 5 exon of this isoform along with other novel start sites and splicing events by form-specific RT-PCR (Figure 4.3). Limiting analysis to known isoforms would have produced an incorrect abundance estimate for the known isoform of FHL3. Moreover, the novel isoform is dominant prior to differentiation, so this potentially important differentiation-associated promoter switch would have been missed (Figure 4(a)). In total, we tested and validated 153 of 185 putative novel transcription start sites by comparison against TAF1 and RNA polymerase II ChIP-Seq peaks (Appendix C). We also observed switches in the major isoform of alternatively spliced genes. In total, 10% of multi-promoter genes featured a switch in major primary transcript and 7% of alternatively spliced primary transcripts switched major isoforms. We concluded that not only is the impact of promoter-switching on mRNA output significant, many genes are also post-transcriptionally overloaded supporting a role for dynamic splicing regulation in myogenesis. A key question is whether genes that display expression overloading are differentially regulated in a particular system because they have isoforms that are functionally specialized for that system. Of the genes undergoing transcriptional or post-transcriptional isoform switches, 26% and 24% code for multiple distinct proteins according to annotation. Genes for which Cufflinks reported a

novel isoform were excluded from a coding sequence analysis, so this fraction likely underestimates the impact of differential regulation on coding potential. We thus speculate that differential RNA level isoform regulation, whether transcriptional, post-transcriptional, or mixed in underlying mechanism, suggests functional specialization of a substantial subset of isoforms.

Appendix A

Lemmas

The following elementary/classical results are required for our methods and we include them so that the thesis is self-contained.

Lemma 10. *Let $X_1, \ldots, X_n$ be random variables and $a_1, \ldots, a_n$ real numbers with $Y = \sum_{i=1}^{n} a_i X_i$. Then*

$$Var[Y] = \sum_{i=1}^{n} a_i^2 Var[X_i] + 2 \sum_{i<j} a_i a_j Cov[X_i, X_j]. \tag{A.1}$$

Lemma 11 (Taylor Series)**.** *If X and Y are random variables then*

$$\begin{aligned} Var[f(X,Y)] \approx & \left(\frac{\partial f}{\partial X}(E[X], E[Y])\right)^2 Var[X] \\ & + 2\frac{\partial f}{\partial X}(E[X], E[Y])\frac{\partial f}{\partial Y}(E[X], E[Y]) Cov[X,Y] \\ & + \left(\frac{\partial f}{\partial Y}(E[X], E[Y])\right)^2 Var[Y]. \end{aligned} \tag{A.2}$$

Corollary 12. *If X and Y are independent then*

$$Var\left[\log\left(\frac{X}{Y}\right)\right] \approx \frac{V[X]}{E[X]^2} + \frac{V[Y]}{E[Y]^2}. \tag{A.3}$$

Corollary 13. *If X and Y are independent random variables then*

$$Var[XY] = Var[X]Var[Y] + E[X]^2 Var[Y] + E[Y]^2 Var[X]. \tag{A.4}$$

The above result is exact using the 2nd order Taylor expansion (higher derivatives vanish).

Lemma 14 (Li *et al*[35]). *Let* $a_1, \ldots, a_n, w_1, \ldots, w_n$ *be real numbers satisfying:* $w_i \neq 0$ *and* $0 \leq a_i \leq 1$ *for all* i, $\sum_{i=1}^n a_i = 1$ *and* $\sum_{i=1}^n a_i w_i \neq 0$. *Let* $b_j = \frac{a_j w_j}{\sum_{i=1}^n a_i w_i}$. *Then* $a_j = \frac{b_j \frac{1}{w_j}}{\sum_{i=1}^n b_i \frac{1}{w_i}}$.

Proof:

$$
\begin{aligned}
b_j &= \frac{a_j w_j}{\sum_{i=1}^n a_i w_i} && \text{(A.5)} \\
\Rightarrow \sum_{k=1}^n \frac{b_k}{w_k} &= \sum_{k=1}^n \frac{a_k}{\sum_{i=1}^n a_i w_i} && \text{(A.6)} \\
&= \frac{1}{\sum_{i=1}^n a_i w_i} && \text{(A.7)} \\
&= \frac{b_j}{a_j w_j} && \text{(A.8)} \\
\Rightarrow a_j &= \frac{b_j \frac{1}{w_j}}{\sum_{i=1}^n b_i \frac{1}{w_i}}. && \text{(A.9)}
\end{aligned}
$$

□

Proposition 15. *Let* $f_i(\theta) = \sum_{j=1}^d a_{ij}\theta_j + b_i$ *(*$1 \leq i \leq m$*) describe a linear statistical model with* $a_{ij} \geq$ *for all* i, j. *That is,* $\sum_{i=1}^m f_i(\theta) = 1$. *If* $u_i \geq 0$ *for all* i *then the log likelihood function*

$$
l(\theta) = \sum_{i=1}^m u_i \log(f_i(\theta)) \tag{A.10}
$$

is concave.[Pachter and Sturmfels (eds.)[51]*]*

Proof: It is easy to see that

$$
\left(\frac{\partial^2 l}{\partial\theta_j \partial\theta_k}\right) = -A^T diag\left(\frac{u_1}{f_1(\theta)^2}, \ldots, \frac{u_m}{f_m(\theta)^2}\right) A, \tag{A.11}
$$

where A is the $m \times d$ matrix whose entry in row i and column j equals a_{ij}. Therefore the Hessian is a symmetric matrix with non-positive eigenvalues, and is therefore negative semi-definite. □

Definition 16. A partially ordered set is a set S with a binary relation $\leq$ satisfying:

1. $x \leq x$ for all $x \in S$,

2. If $x \leq y$ and $y \leq z$ then $x \leq z$,

3. If $x \leq y$ and $y \leq x$ then $x = y$.

A *chain* is a set of elements in $C \subseteq S$ such that for every $x, y \in C$ either $x \leq y$ or $y \leq x$. An *antichain* is a set of elements that are pairwise incompatible.

Partially ordered sets are equivalent to directed acyclic graphs (DAGs). The following min-max theorems relate chain partitions to antichains and are special cases of linear-programming duality. More details and complete proofs can be found in [41].

Theorem 17 (Dilworth's theorem)**.** *Let P be a finite partially ordered set. The maximum number of elements in any antichain of P equals the minimum number of chains in any partition of P into chains.*

Theorem 18 (König's theorem)**.** *In a bipartite graph, the number of edges in a maximum matching equals the number of vertices in a minimum vertex cover.*

Theorem 19. *Dilworth's theorem is equivalent to König's theorem.*

Proof: We first show that Dilworth's theorem follows from König's theorem. Let P be a partially ordered set with n elements. We define a bipartite graph $G = (U, V, E)$ where $U = V = P$, i.e. each partition in the bipartite graph is equally to P. Two nodes u, v form an edge $(u, v) \in E$ in the graph G iff $u < v$ in

P. By König's theorem there exist both a matching M and a a vertex cover C in G of the same cardinality. Let $T \subset S$ be the set of elements not contained in C. Note that T is an antichain in P. We now form a partition W of P into chains by declaring u and v to be in the same chain whenever there is an edge $(u, v) \in M$. Since C and M have the same size, it follows that T and W have the same size.

To deduce König's theorem from Dilworth's theorem, we begin with a bipartite graph $G = (U, V, E)$ and form a partial order P on the vertices of G by defining $u < v$ when $u \in U, v \in V$ and $(u, v) \in E$. By Dilworth's theorem, there exists an antichain of P and a partition into chains of the same size. The non-trivial chains in P form a matching in the graph. Similarly, the complement of the vertices corresponding to the anti-chain in P is a vertex cover of G with the same cardinality as the matching. $\square$

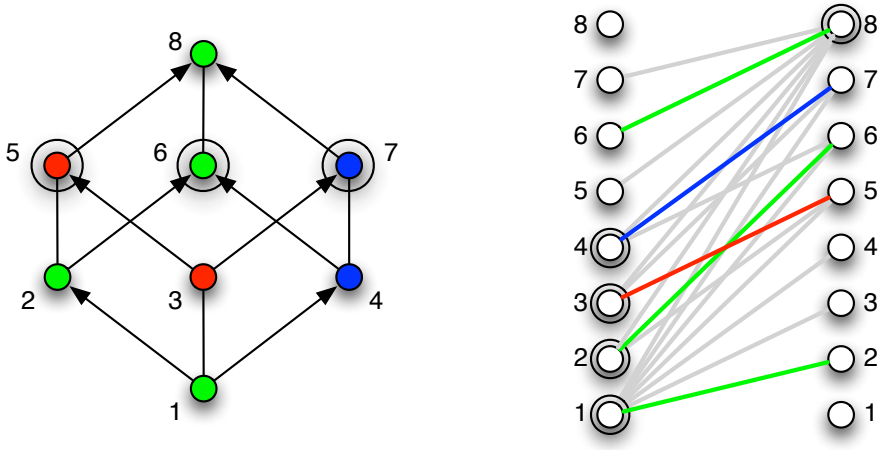
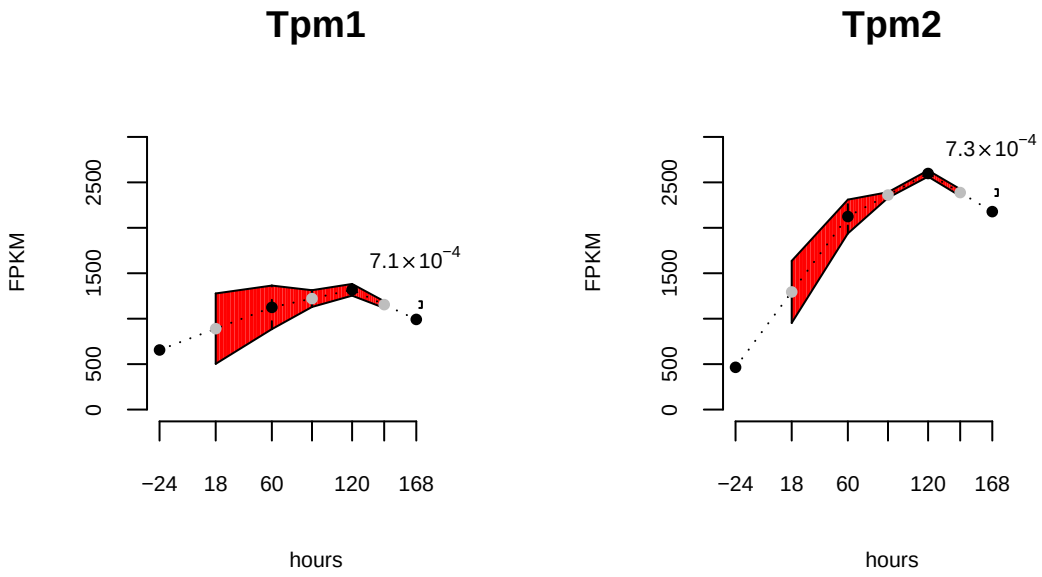
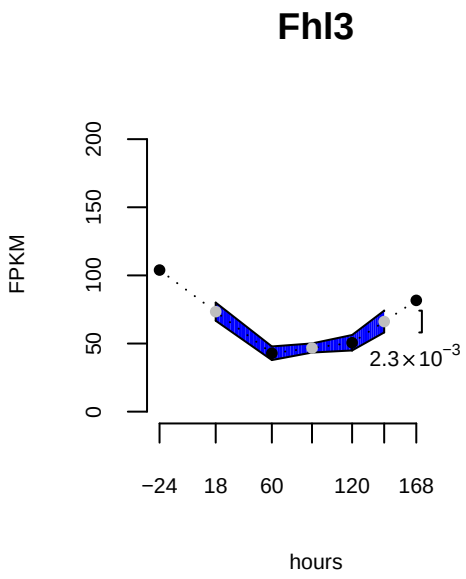


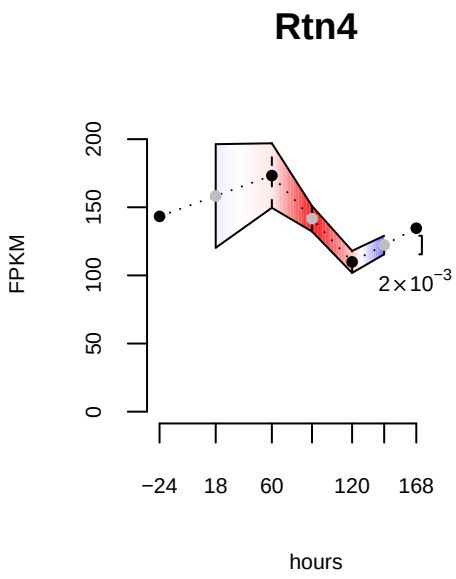
Figure A.1: The equivalence of Dilworth's and König's theorems. The partially ordered set with 8 elements on the left is partitioned into 3 chains. This is the size of a minimum partition into chains, and is equal to the maximum size of an antichain (Dilworth's theorem). The antichain is shown with double circles. On the right, the reachability graph constructed from the partially ordered set on the left is shown. The maximum matching corresponding to the chain partition consists of 5 edges and is equal in size to the number of vertices in a minimum vertex cover (König's theorem). The vertex cover is shown with double circles. Note that 8=3+5.

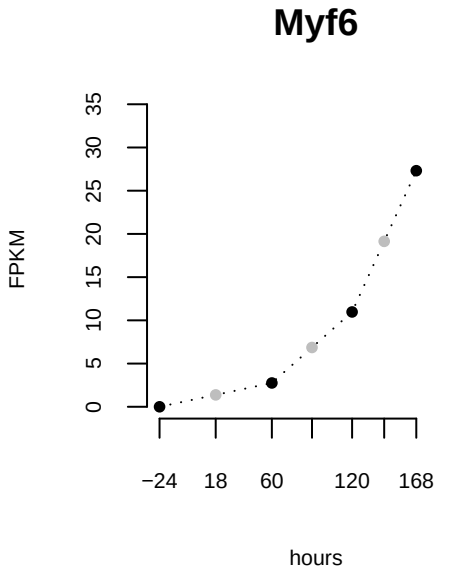
Appendix B

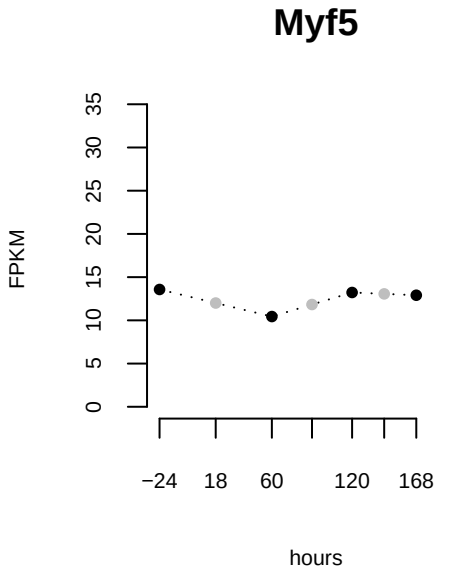
Selected Minard plots

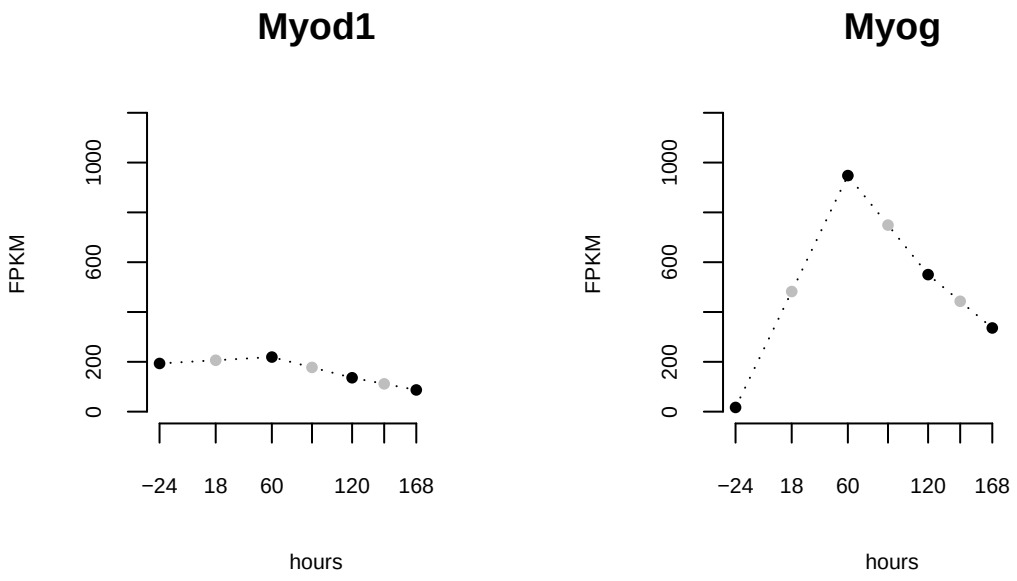


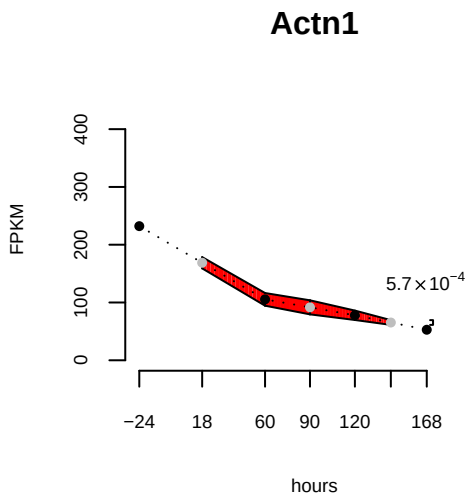

Fhl3
FPKM
0
50
100
150
200
2.3×10−3
−24
18
60
120
168
hours

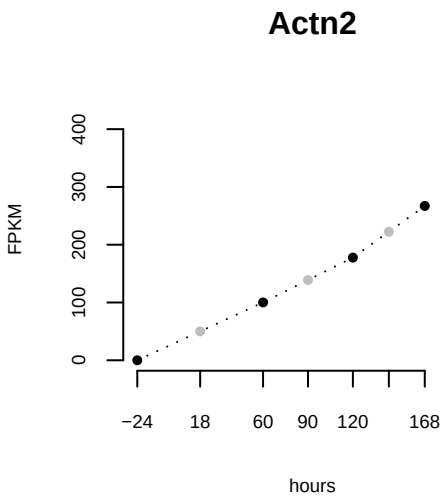

Rtn4
FPKM
0
50
100
150
200
2×10−3
−24
18
60
120
168
hours

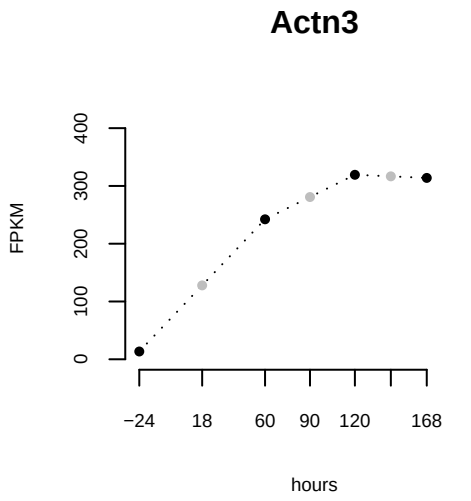

Myf6
FPKM
0
5
10
15
20
25
30
35
−24
18
60
120
168
hours

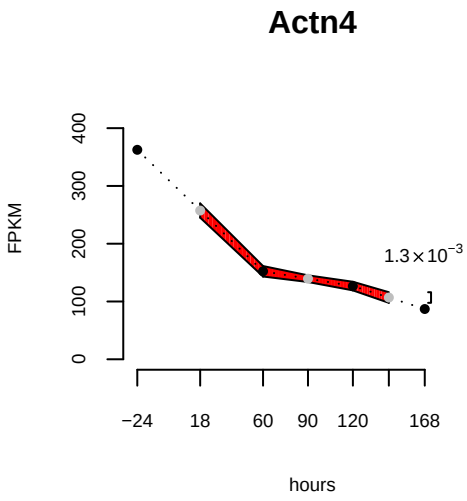

Myf5
FPKM
0
5
10
15
20
25
30
35
−24
18
60
120
168
hours

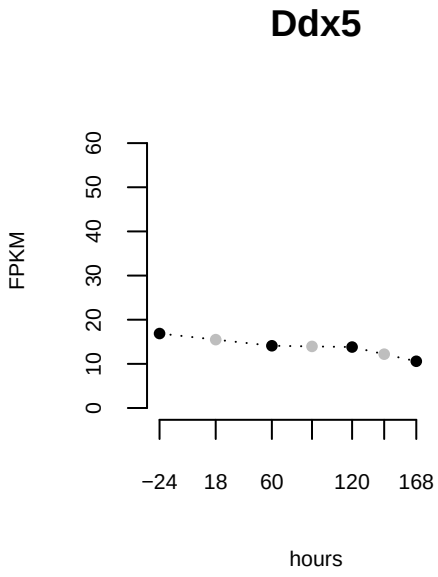

Myod1
FPKM
0
200
600
1000
−24
18
60
120
168
hours
Myog
FPKM
0
200
600
1000
−24
18
60
120
168
hours

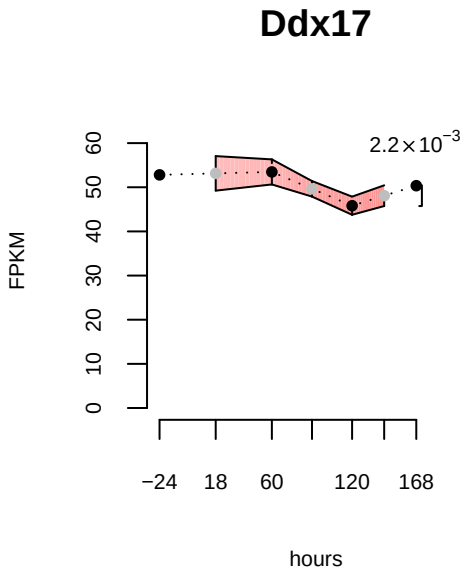

Actn1
FPKM
0
100
200
300
400
5.7×10−4
−24
18
60
90
120
168
hours

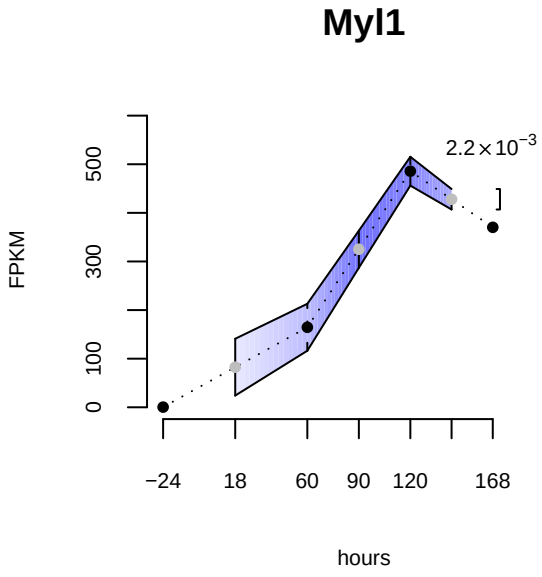

Actn2
FPKM
0
100
200
300
400
−24
18
60
90
120
168
hours

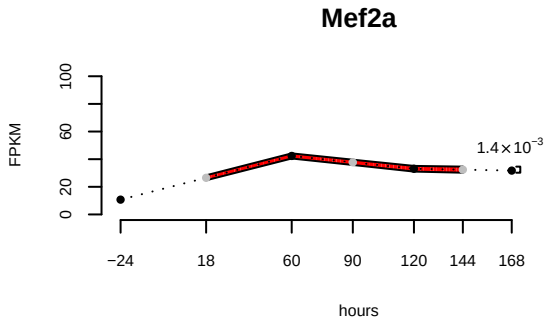

Actn3
FPKM
0
100
200
300
400
−24
18
60
90
120
168
hours

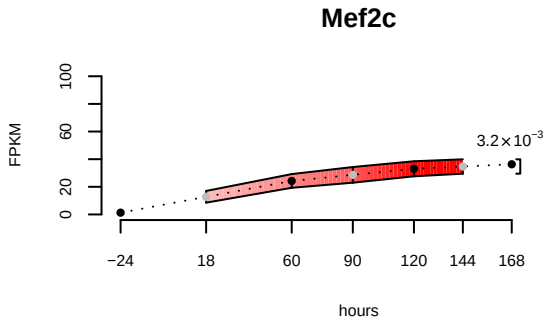

Actn4
FPKM
0
100
200
300
400
1.3×10−3
−24
18
60
90
120
168
hours

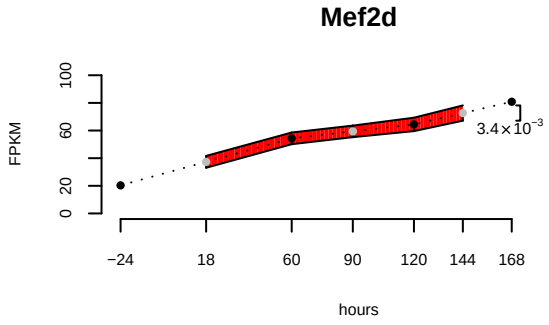

Ddx5
FPKM
0
10
20
30
40
50
60
−24
18
60
120
168
hours


Ddx17
FPKM
0
10
20
30
40
50
60
2.2×10−3
−24
18
60
120
168
hours


Myl1
FPKM
0
100
300
500
2.2×10−3
−24
18
60
90
120
168
hours


Mef2a
FPKM
0
20
60
100
1.4×10^-3
−24
18
60
90
120
144
168
hours


Mef2c
FPKM
0
20
60
100
3.2×10^-3
−24
18
60
90
120
144
168
hours


Mef2d
FPKM
0
20
60
100
3.4×10^-3
−24
18
60
90
120
144
168
hours

Appendix C

Wet experimental methods

Note: Except where otherwise noted, the work described in Appendix C was performed by Brian Williams, and is included for completeness

C.1 RNA isolation

Mouse skeletal muscle C2C12 cells were initially plated on 15 cm plates in DMEM with 20% fetal bovine serum. At confluence, the cells were switched to low serum medium to initiate myogenic differentiation. For extraction of total RNA, cells were first rinsed in PBS and then lysed in Trizol reagent (Invitrogen catalog # 15596-026) either during exponential growth in high serum medium, or at 60 hrs, 5 days and 7 days after medium shift. Residual contaminating genomic DNA was removed from the total RNA fraction using Turbo DNA-free (Ambion catalog # AM1907M). mRNA was isolated from DNA-free total RNA using the Dynabeads mRNA Purification Kit (Invitrogen catalog # 610-06).

C.2 Fragmentation and reverse transcription

Preparation of cDNA followed the procedure described in Mortazavi et al.2, with minor modifications as described below. Prior to fragmentation, a 7 uL aliquot (500 pgs total mass) containing known concentrations of 7 "spiked in" control

transcripts from A. thaliana and the lambda phage genome were added to a 100 ng aliquot of mRNA from each time point. This mixture was then fragmented to an average length of 200 nts by metal ion/heat catalyzed hydrolysis. The hydrolysis was performed in a 25 uL volume at 94C for 90 seconds. The 5X hydrolyis buffer components are: 200 mM Tris acetate, pH 8.2, 500 mM potassium acetate and 150 mM magnesium acetate. After removal of hydrolysis ions by G50 Sephadex filtration (USA Scientific catalog # 1415-1602), the fragmented mRNA was random primed with hexamers and reverse-transcribed using the Super Script II cDNA synthesis kit (Invitrogen catalog # 11917010). After second strand synthesis, the cDNA went through end-repair and ligation reactions according to the Illumina ChIP-Seq genomic DNA preparation kit protocol (Illumina catalog # IP102-1001), using the paired end adapters and amplification primers (Illumina Catalog # PE102-1004). Ligation of the adapters adds 94 bases to the length of the cDNA molecules.

C.3 Size selection

The cDNA library was size-fractionated on a 2% TAE low melt agarose gel (Lonza catalog # 50080), with a 100 bp ladder (Roche catalog # 14703220) run in adjacent lanes. Prior to loading on the gel, the ligated cDNA library was taken over a G50 Sephadex column to remove excess salts that interfere with loading the sample in the wells. After post-staining the gel in ethidium bromide, a narrow slice (2mm) of the cDNA lane centered at the 300 bp marker was cut. The slice was extracted using the QiaEx II kit (Qiagen catalog # 20021), and the extract was

filtered over a Microcon YM-100 microconcentrator (Millipore catalog # 42409) to remove DNA fragments shorter than 100 bps. Filtration was performed by pipeting the extract into the upper chamber of a microconcentrator, and adding ultra pure water (Gibco catalog # 10977) to a volume of 500 uLs. The filter was spun at 500 X g until only 50 uLs remained in the upper chamber (about 20 minutes per spin) and then the upper chamber volume was replenished to 500 uLs. This procedure was repeated 6 times. The filtered sample was then recovered from the filter chamber according to the manufacturers protocol. Fragment length distributions obtained after size selection were estimated from the spike-in sequences and are shown in Figure 2.10.

C.4 Amplification

One-sixth of the filtered sample volume was used as template for 15 cycles of amplification using the paired-end primers and amplification reagents supplied with the Illumina ChIP-Seq genomic DNA prep kit. The amplified product was then cleaned up over a Qiaquick PCR column (Qiagen catalog # 28104), and then the filtration procedure using the Microcon YM-100 microconcentrators described above was repeated, to remove both amplification primers and amplification products shorter than 100 bps. A final pass over a G50 Sephadex column was performed, and the library was quantified using the Qubit fluorometer and PicoGreen quantification reagents (Invitrogen catalog # Q32853). The library was then used to build clusters on the Illumina flow cell according to protocol.

C.5 Endpoint PCR validation of novel isoforms

5 ugs of total RNA from each timepoint was primed with oligodT(20) (Invitrogen catalog # 18418020), and reverse-transcribed at 50C using SuperScript III reverse transcriptase, (Invitrogen catalog # 18080044) according to the manufacturers protocol. One tenth of the cDNA reaction was used as template for 35 rounds of PCR amplification. Amplification primers that cross the Cufflinks predicted spliced-exon junctions were designed using Primer 3 software and purchased from Integrated DNA Technologies, Inc. (San Diego, CA). (Steve Rozen and Helen J. Skaletsky (2000) Primer3 on the WWW for general users and for biologist programmers. In: Krawetz S, Misener S (eds) Bioinformatics Methods and Protocols: Methods in Molecular Biology. Humana Press, Totowa, NJ, pp 365-386. Source code available at http://fokker.wi.mit.edu/primer3/.), One fourth of the PCR product was then loaded on a 1.3% agarose gel, which was post-stained with Sybr Gold (Invitrogen Catalog # S11494) before visualization on a UV transilluminator.

C.6 Validation of novel transcription start sites

Note: The wet work here was performed by Brian Williams, and the validation analysis by Ali Mortazavi

Transcripts with 5 exons not in UCSC, Ensembl, or VEGA were selected for validation. We excluded transcripts with estimated abundances less than 5.0 FPKM at all time points, as well as transcripts with a 5 exon within 200bp of an annotated exon. To validate our novel observed 5 exons, we conducted ChIP-Seq experiments

as previously described28 at -24 and 60 hour time points using an antibody to the unphosphorylated CTD-repeat of RNA polymerase II (8WG16, Covance) as well as an antibody to TAF1 (SC-735, Santa Cruz) which marks promoters. For each candidate 5end, we took the region +/- 200 bp and measured the normalized read density (RPKM) of each ChIP-Seq, requiring at least 1.5 RPKM of ChIP-Seq signal for both polymerase and TAF1 at either time point.

Bibliography

[1] MD Adams, JM Kelley, JD Gocayne, M Dubnick, Mihael Polymeropolous, H Xiao, CR Merril, A Wu, B Olde, R Moreno, AR Kerlvage, WR McCombie, and JC Venter. Complementary dna sequencing: Expressed sequence tags and human genome project. *Science*, 252:1–6, Apr 1991.

[2] Leo A Aroian, Vidya S Taneja, and Larry W Cornwell. Mathematical forms of the distribution of the product of two normal variables. *Communications in Statistics: Theory and Methods*, 7(2):165–172, 1978.

[3] RA Baeza-Yates and CH Perleberg. Fast and practical approximate string matching. *Information Processing Letters*, 59(1):21–27, 1996.

[4] Y Benjamini and Y Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[5] James H Bullard, Elizabeth Purdom, Kasper D Hansen, and Sandrine Dudoit. Evaluation of statistical methods for normalization and differential expression in mrna-seq experiments. *BMC Bioinformatics*, 11:94, Jan 2010.

[6] S Burkhardt and J Kärkkäinen. Better filtering with gapped q-grams. *Fundamenta Informaticae*, Jan 2003.

[7] M Burrows and D. J Wheeler. A block-sorting lossless data compression algorithm. *SRC Research Reports*, Apr 1994.

[8] M Charalambous, P Trancoso, and A Stamatakis. Initial experiences porting a bioinformatics application to a graphics processor. *Advances in Informatics*, Jan 2005.

[9] Nicole Cloonan, Alistair R R Forrest, Gabriel Kolle, Brooke B A Gardiner, Geoffrey J Faulkner, Mellissa K Brown, Darrin F Taylor, Anita L Steptoe, Shivangi Wani, Graeme Bethel, Alan J Robertson, Andrew C Perkins, Stephen J Bruce, Clarence C Lee, Swati S Ranade, Heather E Peckham, Jonathan M Manning, Kevin J Mckernan, and Sean M Grimmond. Stem cell transcriptome profiling via massive-scale mrna sequencing. *Nat Meth*, 5(7):613–619, Jul 2008.

[10] Kimberly R Cordes, Neil T Sheehy, Mark P White, Emily C Berry, Sarah U Morton, Alecia N Muth, Ting-Hein Lee, Joseph M Miano, Kathryn N Ivey, and Deepak Srivastava. mir-145 and mir-143 regulate smooth muscle cell fate and plasticity. *Nature*, 460(7256):705–710, Aug 2009.

[11] D Cottle, M McGrath, B Cowling, and I Coghill. Fhl3 binds myod and negatively regulates myotube formation. *Journal of Cell Science*, Jan 2007.

[12] W Dally, F Labonte, A Das, and P Hanrahan. Merrimac: Supercomputing with streams. *Supercomputing, 2003 ACM/IEEE Conference*, Jan 2003.

[13] A L Delcher, S Kasif, R D Fleischmann, J Peterson, O White, and S L Salzberg. Alignment of whole genomes. *Nucleic Acids Research*, 27(11):2369–76, Jun 1999.

[14] RP Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, pages 161–166, 1950.

[15] T Endo and B Nadal-Ginard. Transcriptional and posttranscriptional control of c-myc during myogenesis: its mrna remains inducible in differentiated cells and does not suppress the differentiated phenotype. *Molecular and Cellular Biology*, Jan 1986.

[16] Nicholas Eriksson, Lior Pachter, Yumi Mitsuya, Soo-Yon Rhee, Chunlin Wang, Baback Gharizadeh, Mostafa Ronaghi, Robert W Shafer, and Niko Beerenwinkel. Viral population estimation using pyrosequencing. *PLoS Computational Biology*, 4(5):e1000074, May 2008.

[17] P Ferragina and G Manzini. Opportunistic data structures with applications. *ANNUAL SYMPOSIUM ON FOUNDATIONS OF COMPUTER SCIENCE*, Jan 2000.

[18] P Ferragina and G Manzini. An experimental study of a compressed index. *Information Sciences*, Jan 2001.

[19] B Fuglede and F Topsoe. Jensen-shannon divergence and hilbert space embedding. *IEEE International Symposium on Information Theory*, Jan 2004.

[20] Scott F. Gilbert. Developmental biology. *Sinauer Associates*, Jan 2009.

[21] N Govindaraju, S Larsen, and J Gray. A memory model for scientific algorithms on graphics processors. *Supercomputing, 2006. SC'06. Proceedings of the ACM/IEEE SC 2006 Conference*, Jan 2006.

[22] Mitchell Guttman, Ido Amit, Manuel Garber, Courtney French, Michael F Lin, David Feldser, Maite Huarte, Or Zuk, Bryce W Carey, John P Cassady, Moran N Cabili, Rudolf Jaenisch, Tarjei S Mikkelsen, Tyler Jacks, Nir Hacohen, Bradley E Bernstein, Manolis Kellis, Aviv Regev, John L Rinn, and Eric S Lander. Chromatin signature reveals over a thousand highly conserved large non-coding rnas in mammals. *Nature*, 457(7235):223–227, Sep 2009.

[23] Brian J Haas, Arthur L Delcher, Stephen M Mount, Jennifer R Wortman, Roger K Smith, Linda I Hannick, Rama Maiti, Catherine M Ronning, Douglas B Rusch, Christopher D Town, Steven L Salzberg, and Owen White. Improving the arabidopsis genome annotation using maximal transcript alignment assemblies. *Nucleic Acids Research*, 31(19):5654–66, Oct 2003.

[24] M Harris, G Coombe, and T Scheuermann. Physically-based visual simulation on graphics hardware. *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, Jan 2002.

[25] D Hiller, H Jiang, W Xu, and W. H Wong. Identifiability of isoform deconvolution from junction arrays and rna-seq. *Bioinformatics*, 25(23):3056–3059, Dec 2009.

[26] LaDeana W Hillier, Gabor T Marth, Aaron R Quinlan, David Dooling, Ginger Fewell, Derek Barnett, Paul Fox, Jarret I Glasscock, Matthew Hickenbotham, Weichun Huang, Vincent J Magrini, Ryan J Richt, Sacha N Sander, Donald A Stewart, Michael Stromberg, Eric F Tsung, Todd Wylie, Tim Schedl, Richard K Wilson, and Elaine R Mardis. Whole-genome sequencing and variant discovery in c. elegans. *Nat Meth*, 5(2):183–188, Feb 2008.

[27] J Hopcroft and R Karp. An $n\hat{5}/2$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, Jan 1973.

[28] H Jiang and W Wong. Statistical inferences for isoform expression in rna-seq. *Bioinformatics*, Feb 2009.

[29] D. S Johnson, A Mortazavi, R. M Myers, and B Wold. Genome-wide mapping of in vivo protein-dna interactions. *Science*, 316(5830):1497–1502, Jun 2007.

[30] W. J Kent. Blat—the blast-like alignment tool. *Genome Research*, 12(4):656–664, Mar 2002.

[31] B Langmead, C Trapnell, M Pop, and S L Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biology*, Jan 2009.

[32] Lareau, M Inada, R Green, J Wengrod, and S Brenner. Unproductive splicing of sr genes associated with highly conserved and ultraconserved dna elements. *Nature*, Jan 2007.

[33] P A Lawrence. The making of a fly: the genetics of animal design. *Wiley-Blackwell*, page 228, Jan 1992.

[34] Timothy J Ley, Elaine R Mardis, Li Ding, Bob Fulton, Michael D Mclellan, Ken Chen, David Dooling, Brian H Dunford-Shore, Sean Mcgrath, Matthew Hickenbotham, Lisa Cook, Rachel Abbott, David E Larson, Dan C Koboldt, Craig Pohl, Scott Smith, Amy Hawkins, Scott Abbott, Devin Locke, LaDeana W Hillier, Tracie Miner, Lucinda Fulton, Vincent Magrini, Todd Wylie, Jarret Glasscock, Joshua Conyers, Nathan Sander, Xiaoqi Shi, John R Osborne, Patrick Minx, David Gordon, Asif Chinwalla, Yu Zhao, Rhonda E Ries, Jacqueline E Payton, Peter Westervelt, Michael H Tomasson, Mark Watson, Jack Baty, Jennifer Ivanovich, Sharon Heath, William D Shannon, Rakesh Nagarajan, Matthew J Walter, Daniel C Link, Timothy A Graubert, John F Dipersio,

and Richard K Wilson. Dna sequencing of a cytogenetically normal acute myeloid leukaemia genome. *Nature*, 456(7218):66–72, Nov 2008.

[35] Bo Li, Victor Ruotti, Ron M Stewart, James A Thomson, and Colin N Dewey. Rna-seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, Dec 2009.

[36] H Li, J Ruan, and R Durbin. Mapping short dna sequencing reads and calling variants using mapping quality scores. *Genome Research*, page 19, Aug 2008.

[37] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–9, Aug 2009.

[38] R Li, Y Li, K Kristiansen, and J Wang. Soap: short oligonucleotide alignment program. *Bioinformatics*, Jan 2008.

[39] H Lin, Z Zhang, M Zhang, B Ma, and M Li. Zoom! zillions of oligos mapped. *Bioinformatics*, Aug 2008.

[40] W Liu, B Schmidt, and G Voss. Streaming algorithms for biological sequence alignment on gpus. *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, Jan 2007.

[41] L Lovász and M Plummer. Matching theory. *AMS Chelsea Publishing*, Jan 2009.

[42] B Ma, J Tromp, and M Li. Patternhunter: faster and more sensitive homology search. *Bioinformatics*, Jan 2002.

[43] J Marioni, C Mason, S Mane, M Stephens, and Y Gilad. Rna-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, page 10, Apr 2008.

[44] Tarjei S Mikkelsen, Manching Ku, David B Jaffe, Biju Issac, Erez Lieberman, Georgia Giannoukos, Pablo Alvarez, William Brockman, Tae-Kyung Kim, Richard P Koche, William Lee, Eric Mendenhall, Aisling O'Donovan, Aviva Presser, Carsten Russ, Xiaohui Xie, Alexander Meissner, Marius Wernig, Rudolf Jaenisch, Chad Nusbaum, Eric S Lander, and Bradley E Bernstein. Genome-wide maps of chromatin state in pluripotent and lineage-committed cells. *Nature*, 448(7153):553–60, Aug 2007.

[45] N Morinaga, S C Tsai, J Moss, and M Vaughan. Isolation of a brefeldin a-inhibited guanine nucleotide-exchange protein for adp ribosylation factor (arf) 1 and arf3 that contains a sec7-like domain. *Proc Natl Acad Sci USA*, 93(23):12856–60, Nov 1996.

[46] Ali Mortazavi, Brian A Williams, Kenneth Mccue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nat Meth*, 5(7):621–628, Jul 2008.

[47] U Nagalakshmi, Z Wang, K Waern, C Shou, and D Raha. The transcriptional landscape of the yeast genome defined by rna sequencing. *Science*, Jan 2008.

[48] Yohei Ohtake, Hideaki Tojo, and Motoharu Seiki. Multifunctional roles of mt1-mmp in myofiber formation and morphostatic maintenance of skeletal muscle. *J Cell Sci*, 119(Pt 18):3822–32, Sep 2006.

[49] S OOta and N Saitou. Phylogenetic relationship of muscle tissues deduced from superimposition of gene trees. *Molecular Biology And Evolution*, 16(6):856–67, Jun 1999.

[50] J Owens, D Luebke, N Govindaraju, and M Harris. A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, Jan 2007.

[51] L Pachter and B Sturmfels. Algebraic statistics for computational biology. *Cambridge Universiy Press*, Jan 2005.

[52] Itsik Pe'er and Jacques S Beckmann. Recovering frequencies of known haplotype blocks from single-nucleotide polymorphism allele frequencies. *Genetics*, 166(4):2001–6, Apr 2004.

[53] S Popov, J Gunther, H Seidel, and P Slusallek. Stackless kd-tree traversal for high performance gpu ray tracing. *Computer Graphics Forum*, Jan 2007.

[54] Ritchie, S Granjeaud, and D Puthier. Entropy measures quantify global splicing disorders in cancer. *PLoS Computational Biology*, Jan 2008.

[55] Gordon Robertson, Martin Hirst, Matthew Bainbridge, Misha Bilenky, Yongjun Zhao, Thomas Zeng, Ghia Euskirchen, Bridget Bernier, Richard Varhol, Allen Delaney, Nina Thiessen, Obi L Griffith, Ann He, Marco Marra, Michael Snyder, and Steven Jones. Genome-wide profiles of stat1 dna association using chromatin immunoprecipitation and massively parallel sequencing. *Nat Meth*, 4(8):651–7, Aug 2007.

[56] Stephen M Rumble, Phil Lacroute, Adrian V Dalca, Marc Fiume, Arend Sidow, and Michael Brudno. Shrimp: accurate mapping of short color-space reads. *PLoS Computational Biology*, 5(5):e1000386, May 2009.

[57] S Salzberg, M Pertea, A Delcher, and M Gardner. Interpolated markov models for eukaryotic gene finding. *Genomics*, Jan 1999.

[58] M Schatz, C Trapnell, A Delcher, and A Varshney. High-throughput sequence alignment using graphics processing units. *BMC Bioinformatics*, Jan 2007.

[59] Andrew D Smith, Zhenyu Xuan, and Michael Q Zhang. Using quality scores and longer reads improves accuracy of solexa read mapping. *BMC Bioinformatics*, 9(1):128, Jan 2008.

[60] T Smith and M Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, Jan 1981.

[61] M Stanke, R Steinkamp, and S Waack. Augustus: a web server for gene finding in eukaryotes. *Nucleic Acids Research*, Jan 2004.

[62] Stephen J Tapscott. The circuitry of a master switch: Myod and the regulation of skeletal muscle gene transcription. *Development*, 132(12):2685–95, Jun 2005.

[63] Vincent Le Texier, Jean-Jack Riethoven, Vasudev Kumanduri, Chellappa Gopalakrishnan, Fabrice Lopez, Daniel Gautheret, and Thangavel Alphonse Thanaraj. Alttrans: transcript pattern variants annotated for both alternative splicing and alternative polyadenylation. *BMC Bioinformatics*, 7:169, Jan 2006.

[64] C Trapnell, L Pachter, and S. L Salzberg. Tophat: discovering splice junctions with rna-seq. *Bioinformatics*, 25(9):1105–1111, May 2009.

[65] Cole Trapnell and Michael C Schatz. Optimizing data intensive gpgpu computations for dna sequence alignment. *PARALLEL COMPUTING*, pages 1–12, Jul 2009.

[66] E Tufte and G Howard. The visual display of quantitative information. *inst.usu.edu*, Jan 1983.

[67] E Ukkonen. On-line construction of suffix trees. *Algorithmica*, Jan 1995.

[68] Eric T Wang, Rickard Sandberg, Shujun Luo, Irina Khrebtukova, Lu Zhang, Christine Mayr, Stephen F Kingsmore, Gary P Schroth, and Christopher B Burge. Alternative isoform regulation in human tissue transcriptomes. *Nature*, page 7, Nov 2008.

[69] Thomas D Wu and Colin K Watanabe. Gmap: a genomic mapping and alignment program for mrna and est sequences. *Bioinformatics*, 21(9):1859–75, May 2005.

[70] Yi Xing, Alissa Resch, and Christopher Lee. The multiassembly problem: reconstructing multiple transcript isoforms from est fragment mixtures. *Genome Research*, 14(3):426–41, Mar 2004.

[71] K Yun and B Wold. Skeletal muscle determination and differentiation: story of a core regulatory network and its context. *Current opinion in cell biology*, 8(6):877–889, 1996.

[72] Daniel R Zerbino and Ewan Birney. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 18(5):821–9, May 2008.