Sequencing of RNA has been an important application of DNA sequencing technology since its invention. RNA is usually sequenced by first converting it to complementary DNA (cDNA) with the reverse transcriptase enzyme (RNA-dependent DNA polymerase). Reverse transcriptase was originally isolated from Rous sarcoma retrovirus and Rauscher mouse leukemia retrovirus (R-MLV) by Baltimore (1970) and independently by Temin (1970). In 1972, Verma et al. and Bank et al. developed efficient systems to copy messenger RNA (mRNA) to cDNA by adding DNA nucleotide triphosphates and short pieces of oligo(dT), which hybridize to the poly(A) tail of the mRNAs and act as a primer.

cDNA is frequently the subject of sequencing studies, because this an efficient method to discover the coding sequence of expressed genes or for finding gene coding regions in genomic DNA sequence. Craig Venter expanded this method by collecting large numbers of short single reads from the 3′ ends of mRNA, which were called expressed sequence tags (ESTs). Early EST sequencing of human cells was extraordinarily productive, resulting in the discovery of many thousands of new genes (Adams et al. 1991, 1992). The EST method allowed for a rough form of gene expression measurements in a variety of cell types and some differential expression studies were conducted in this manner. EST sequencing also became a valuable component of de novo sequencing projects, providing a layer of gene expression information and seeding annotation and gene finding efforts.

Microarray technology, developed in the 1990s, measures the hybridization of labeled cDNA to an array of DNA probes that correspond to the sequences of known genes (or ESTs). The microarray method allows for the discovery, in a genome-wide fashion, of gene expression changes (as reflected in changes of mRNA levels) resulting from any biological treatment or condition.
RNA sequencing with NGS technology (RNA-seq) can be used for a number of different scientific applications. The NGS reads are mapped to a reference genome, then the number of reads mapping within a feature of interest (such as a gene or exon), is a measure of expression. Direct sequencing of mRNA provides a measurement of gene expression for the entire transcriptome that is more accurate and has a greater dynamic range than microarray-based technologies (Marioni et al. 2008). Just as in microarray experiments, the most common application of RNA-seq is to identify genes that change expression between experimental conditions. RNA-seq can also be used to detect mutations in transcribed portions of the genome for the native germline cells of an individual or for somatic mutations in tumor cells. RNA-seq is also an excellent platform to measure alternative splicing events that produce different transcripts (and ultimately different proteins) from a single gene. Alternative transcript isoforms can be detected with great accuracy by using RNA-seq reads mapping at splice junctions, specifying both known as well as novel isoforms. With appropriate sample preparation methods, RNA-seq can also be used to interrogate a wide variety of non-protein-coding RNAs.

Protocols for sequencing of RNA have been developed by all of the major NGS vendors. Ribosomal RNA (rRNA) and transfer RNA (tRNA) are very abundant in the total RNA extracted from both prokaryotic and eukaryotic cells (~75% of RNA molecules). Sequencing of abundant non-protein-coding RNA reduces yield and sensitivity of RNA-seq methods for mRNA and increases cost. Most protocols for RNA-seq in eukaryotic cells use poly(T) oligonucleotides to isolate mRNA with poly(A) tails, or use poly(T) primers in combination with random short oligomers for reverse transcription. After poly(A) enrichment, and cDNA synthesis, most protocols shatter cDNA molecules into small fragments (from 100 to 300 bp) that are then ligated with oligomers specific for the sequencing system. Some protocols have also been developed to sequence small non-protein-coding RNA molecules such as micro-RNA (miRNA), small interfering RNA (siRNA), small nuclear RNA (snRNA), small nucleolar RNA (snoRNA), Piwi-interacting RNA (piRNA), and others.

Another method of removing rRNA, tRNA, and other highly abundant RNAs before sequencing is called duplex-specific nuclease (DSN) normalization. This method uses a nuclease (Kamchatka crab hepatopancrease) that specifically degrades double-stranded DNA, while leaving single-stranded DNA molecules intact. This method takes advantage of reassociation kinetics. First, total RNA is reverse transcribed to double-stranded cDNA. Then the cDNA is denatured at high temperature. Under selective annealing conditions, the most abundant cDNA molecules (cDNA clones of rRNA, tRNA, mtRNA, and the most highly transcribed messages) form double strands and are degraded, whereas less abundant molecules remain single stranded and are preserved. Illumina has presented data using DSN normalization for RNA-seq (http://www.illumina.com/documents/seminars/presentations/2010-06_sq_03_lakdawalla_transcriptome_sequencing.pdf)
that indicates very good removal of rRNA, high retention of small noncoding RNAs, and no 3' bias compared with poly(A) purification methods. Relatively few RNA-seq experiments have been published using the DSN method, so it is not clear what bias it creates in gene expression or differential expression values.

Despite purification methods (poly(A) selection or DSN normalization), RNA-seq data may still contain substantial amounts of rRNA, tRNA, and also mitochondrial RNA. These can be filtered out in the bioinformatics pipeline by simply using a “contaminant” file of rRNA, tRNA, and mtDNA sequences for the target species to prefilter all sequence reads (by alignment with a short read aligner such as BWA or Bowtie) before mapping the remaining reads to the genome and/or splice junction database.

**DEPTH OF COVERAGE AND NUMBER OF REPLICA***

To accurately measure changes in gene expression for a specific gene between two experimental conditions (differential expression), the number of mRNA reads sequenced from the transcript of that gene must be above the threshold of detection in each sample, or at least in samples from conditions in which that gene is expressed. In addition, given the biological and technical variability of counting mRNA molecules, the absolute counts per sample must be large enough to allow for an accurate variance measurement across several replicate samples for each experimental condition. As the total amount of sequence reads per sample increases, expression levels for each gene can be estimated more accurately and statistical power to detect differential expression (DE) increases. Somewhat counterintuitively, Tarazona et al. (2011) suggest that as the number of reads per sample increases, the number of false positives for DE calls increases for many statistical methods.

Although the goal of most RNA-seq experiments is to accurately profile the expression levels of all genes, different cell types express genes at dramatically different levels, creating unique transcriptome profiles. For any given cell type, some genes may be expressed at very high levels, perhaps as much as 5% or 10% of the total mRNA, whereas some genes are not expressed at detectable levels (i.e., less than one mRNA molecule per cell). Therefore, it is probably not possible to sequence enough reads per sample to accurately assess DE for every single gene in the genome. Blencowe et al. (2009) suggested that 700 million reads per sample were required to obtain accurate quantitation for 95% of all expressed transcripts in mammalian cells. These very low-expressed transcripts may not be good targets for DE analysis because the variance of read counts may be quite high across replicates.

Given that there is some limit on the sensitivity of RNA-seq to detect transcripts and accurately assess differential gene expression, the relevant question for most investigators becomes: What is the practical limit of sensitivity in which reliable
expression information can be obtained for the majority of transcriptionally active genes? In fact, a steep curve of diminishing returns has been observed in a number of studies that have explored the depth of coverage for RNA-seq. In Figure 1, data from three studies all show a consistent pattern of decreased discovery of new genes (covered by at least five reads) at increasing depth of coverage. Marioni et al. (2008) discover 232 new genes per each additional million reads at a depth of 22 million, the MAQC study finds 70 new genes per million at a depth of 45 million, and Griffith et al. (2010) find 19 new genes per million at a depth of 200 million (and by subsampling the Griffith data, the discovery rate at a depth of 20 million is 210 and at a depth of 45 million is 75). Data from Toung et al. (2011) (Fig. 2) is also consistent, with very few new transcripts discovered as sequencing depth increases from 100 million reads to nearly one billion reads.

Of course, the actual number of expressed genes and their relative abundance within the cell varies depending on organism, tissue type, cell type, and developmental status. Low abundance transcripts may be biologically important regulators, but deep coverage sequencing of in vivo samples may also capture transcripts from non-target cell types mixed into the sample. Very deep sequencing studies also observe rare noncoding transcripts from much of the genome. In fact, the percentage of non-coding transcripts increases among newly detected genes at deeper levels of coverage, whereas discovery of new protein-coding genes reaches near saturation at much lower coverage levels (Tarazona et al. 2011).

Another important consideration for the design of an RNA-seq experiment is the number of replicates for each biological condition. Researchers generally wish to know the optimal number of replicates required to achieve a desired level of statistical power to find DE. Li et al. (2013) developed a model to calculate statistical power and estimate sample size for RNA-seq experiments based on a negative binomial model of variation in counts per gene in each sample and an exact test for DE. They show sample size requirements in a simulation experiment and by reanalysis of published data for two experiments with human tissues. In the simulation, in which variance among replicates was low ($\phi^* = 0.1$) and log$_2$-fold change was 2.0 or more, only three to six replicates are required to find all of the DE genes with coverage greater than five reads and a false discovery rate (FDR) less than 5%. Increasing the variance to $\phi^* = 0.5$ triples the number of required replicates, and lowering the log$_2$-fold change to 1.0 (a twofold change in expression) increases the required number of replicates to 20. In real biological data the variance often exceeds 0.6 and is overdispersed compared to the expectation of a Poisson model (Fang et al. 2012). For example, in a data set extracted from Bleckman et al. (2010), RNA-seq of liver samples are compared between three human males and three females. Average read coverage of 13,267 detected genes is 1.6 and the dispersion is $\phi^* = 0.6513$. In this data set, to discover 80% of twofold DE genes with FDR of 10% would require a sample size of 79 per condition. (See Table 1.)
FIGURE 1. Comparison of three published studies of transcript discovery versus RNA-seq sampling depth by Marioni et al. (2008), Shi et al. (2006), and Griffith et al. (2010). (Reprinted, with permission, from Tarazona et al. 2011, © Cold Spring Harbor Laboratory Press.)
The general conclusion from these sample size calculations is that the majority of published RNA-seq experiments have very low power—perhaps only 50% of true DE genes can be discovered with adequate control of false discovery; and the discovery of these DE genes is biased toward genes that are highly expressed and show large fold change between conditions. Combining considerations of sequencing depth and

**FIGURE 2.** 100 million reads detect 81% of genes at FPKM ≥ 0.05. Each additional 100 million reads detect ~3% more genes. (Reprinted, with permission, from Toung et al. 2011, ©Cold Spring Harbor Laboratory Press.)

**TABLE 1.** Sample size simulation for an RNA-seq experiment with 10,000 detected genes with log-fold change \( \log_2(p^\star) \) ranging from 0.5 to 2.5 and dispersion \( \phi^\star \) at 0.1 and 0.5

<table>
<thead>
<tr>
<th>( \log_2(p^\star) )</th>
<th>( \phi^\star )</th>
<th>( \mu_0^\star = 1 ) FDR</th>
<th>( \mu_0^\star = 5 ) FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1%</td>
<td>5%</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td>365 (81)</td>
<td>305 (84)</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>518 (81)</td>
<td>433 (84)</td>
</tr>
<tr>
<td>1.0</td>
<td>0.1</td>
<td>79 (81)</td>
<td>67 (84)</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>119 (81)</td>
<td>99 (83)</td>
</tr>
<tr>
<td>1.5</td>
<td>0.1</td>
<td>31 (82)</td>
<td>26 (83)</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>49 (81)</td>
<td>41 (83)</td>
</tr>
<tr>
<td>2.0</td>
<td>0.1</td>
<td>16 (85)</td>
<td>13 (84)</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>26 (82)</td>
<td>22 (84)</td>
</tr>
<tr>
<td>2.5</td>
<td>0.1</td>
<td>8 (85)</td>
<td>7 (89)</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>14 (83)</td>
<td>12 (87)</td>
</tr>
</tbody>
</table>

Reproduced, with permission, from Li et al. 2013.

The sample size is shown for minimum normalized read counts per gene \( (\mu_0^\star) \) of 1 and 5 and FDR rates of 1%, 5%, and 10%. Numbers in parentheses after sample size are the number of differential genes detected by the exact text using edgeR (in which the true number of DE genes is 80).
sample size leads to a general recommendation that has been repeated by a number of investigators. More biologically relevant DE genes will be discovered by sequencing more samples at lower depth of coverage rather than fewer samples at greater depth. Hart et al. (2013) surveyed 127 RNA-seq experiments and found that a sequencing depth of 10 million reads will ensure that approximately 90% of all annotated genes will be covered by at least 10 reads, and that no greater detection of DE at twofold expression change can be achieved with greater depth of sequencing. The larger constraint on detecting DE for a gene is the variance of the expression measurements across replicates rather than the depth of coverage (see Fig. 3). Rapaport et al. (2013) summarize the extensive RNA-seq DE benchmarking efforts of the MAQC/SEQC group with the simple statement: “Our results demonstrate that increasing the number of replicate samples significantly improves detection power over increased sequencing depth.”

The abundance of transcripts from different genes observed in RNA-seq data has been shown to accurately represent the gene expression profile of various cell samples when validated by other technologies such as RNA microarray and quantitative polymerase chain reaction (PCR) (Maroni et al. 2008). As the total yield of NGS machines has increased, the sensitivity of RNA-seq has greatly exceeded microarray-based methods of measuring transcripts from genes expressed at low levels. Because RNA-seq does not rely on existing sequence data for the creation of probes, it can measure the expression of unannotated genes and portions of known genes not previously observed in transcripts such as 5′ and 3′ extensions as well as a variety of alternatively spliced isoforms that include regions annotated as introns. Pickrell et al. (2010) found that about 15% of mapped RNA-seq reads were located outside annotated exons. Figure 4 illustrates the RNA-seq mapping around the ADM gene,
with large numbers of reads mapping to annotated exons, but some reads also map to introns and 5′ regions.

RNA-seq experiments may have goals other than quantifying gene expression and detecting expression changes across experimental conditions. Interrogation of alternative splicing requires adequate coverage of all potential splice junction sites on a transcript (an average coverage of five reads per base across every base in the entire length of the transcript), and discovery of low abundance transcript isoforms may require much deeper coverage. Discovery of sequence variants (single-nucleotide polymorphism [SNPs]) in RNA requires an average depth of coverage greater than 10× for every base in each expressed gene. The actual number of reads required depends of course on the size of the transcriptome for the target species.

**PREALIGNMENT QC OF RNA-seq DATA**

RNA-seq data is evaluated on the sequencer in exactly the same way as DNA sequence data. For Illumina sequencers, information is provided during the sequencing run on flowcell fluorescent intensity, cluster density, phasing/prephasing, quality score distribution (after cycle 25), error rate (based on alignment of spiked in Phix sequences after 52 cycles of sequencing are completed), percentage of clusters passing the signal/noise filter, percentage of successful detection of multiplex barcodes. Once the sequencing run is complete, FastQC or similar tools can be used on the FASTQ files to evaluate overall quality scores and other parameters. Per base quality score graphs can identify possible low-quality regions at 5′ or 3′ ends of reads or problematic drops in quality within the reads (see Chapter 3). Quality metrics of specific concern for RNA-seq include the amount of ribosomal RNA (rRNA) present in each sample (which may be detected as overrepresented sequences), and the overall amount of sequence duplication. High sequence duplication may indicate excess amounts of rRNA, PCR artifacts during sample preparation, or it may simply

**FIGURE 4.** RNA-seq reads mapped to the human genome in the region of the ADM gene.
reflect high expression levels for a few genes that dominate the transcriptome of the target cells. Of greater concern than the absolute duplication level, is the consistency of the duplication measured across samples in a single experiment, because differences in duplication will create bias when DE is calculated.

**MAPPING RNA-seq READS TO GENES**

After sequencing, the standard approach to RNA-seq data analysis requires mapping of all NGS reads to a reference genome, using sequence alignment software, and measuring expression for each gene by counting the number of sequence reads that align to its coding region. Alignment of millions of short RNA-seq reads is generally performed using either a software method based on the Burrows-Wheeler transformation, such as BWA (Li and Durbin 2009) or Bowtie (Langmead et al. 2009) or a method based on a hash table of short \( k \)-mers (strings of sequence letters) such as SHRIMP (Rumble et al. 2009), BFAST (Homer et al. 2009) and the Illumina ELAND aligner (see Chapter 4). All of these alignment tools find perfect matches, matches with one or two mismatches, very small insertion/deletions, and filter out repeated sequences that match the genome in many different locations. Alignment is generally the most computationally demanding step in any NGS analysis workflow, requiring many hours on a cluster of high-performance computers (see Chapter 12).

There are informatics challenges in aligning short NGS RNA reads as query sequences to a genomic DNA reference, because the RNA sequences have introns spliced out. Many (or most) genes have multiple splicing isoforms and some low level of unspliced RNA and anti-sense RNA may be present in an RNA sample. Additional alignment problems include SNPs (and other sequence variants) between the actual biological source of the RNA being sequenced and the reference genome, sequencing errors, and reads that align to multiple positions on the genome. RNA editing is a newly discovered source of differences between sequenced cDNA fragments and the reference genome (Li et al. 2011).

Possible approaches to the intron splicing problem include building a reference database of transcript sequences, using an alignment method tolerant of large gaps, or creating a supplemental database of predicted splice junction fragments with flanking exon sequences. If the reference database (i.e. RefSeq or ENSEMBL gene annotations) includes alternative transcripts for each gene, the quantification of alternative transcripts requires that each read be assigned to a specific transcript, even when they are fully overlapping. Mapping only to known transcripts undermines the power of RNA-seq to find novel splice isoforms and expression from unannotated portions of the genome (new genes or new exon regions of known genes).

Illumina has developed a method in its CASAVA software to quantify gene expression for its sequencers. Reads that are contained entirely in a single exon
are mapped to the reference genome without splicing. In addition, the reference genome, annotated with the locations of exons and introns for each gene, is processed into a splice junction database. The junction sequences are extracted with flanking exon regions several bases shorter than the length of the RNA-seq reads, so that each spliced read is required to straddle the junction and be anchored to both exons. If a gene has multiple isoforms, the coding region is considered to be the union of all regions annotated as exons in any isoforms, and multiple incompatible splice junction fragments may exist in the junction database. This method allows for mapping of most RNA-seq reads to their corresponding gene, regardless of the placement of the read with respect to known splice junctions. The Illumina method calculates the gene expression value as the sum of the total number of bases from all reads that align to exon and splice junction locations for each gene. The CASAVA software reports the number of bases from all reads that map to each exon, bases that map to introns, or reads that map to the genome outside of the annotated exonic regions. Users of Illumina software should be aware that the ELAND/CASAVA RNA-seq method has undergone many changes in various software versions, so RNA-seq gene expression values may change depending on the software version used to calculate them.

The TopHat/Cufflinks software package (Trapnell et al. 2009) takes a different approach to gene expression and intron splicing. First, TopHat calculates its own transcriptome database from a reference genome and a set of gene exon annotations (a GFF file such as from RefSeq or ENSEMBL), then reads are mapped with Bowtie to this transcriptome database. Reads that do not map completely within known transcript isoforms are mapped to the genome and used to predict novel splice junctions from gaps in the alignment and the presence of splice junction consensus sequences (GT-AG, GC-AG, and AT-AC). TopHat uses Bowtie to detect gapped alignments using a split-read strategy, breaking each read into nonoverlapping segments of 25 bp (an adjustable parameter) which are each mapped independently to the reference genome with a maximum of 2 mismatches (an adjustable parameter). The first and strongest source of evidence for a splice junction is when two segments from the same read (for reads of at least 45bp) are mapped with a gap of a specified distance on the reference genome, suggesting that such reads are spanning two exons. A second method, called coverage-search, detects a splice by joining two “coverage islands”, which are distinct neighboring regions of piled up reads in the initial mapping. The coverage-search option should only be used for data sets with short reads (< 45bp) and with a small number of reads (<=10 million), and it will only find introns with the consensus GT-AG junction sequence. It is possible for coverage-search to incorrectly join exons of adjacent genes if they both have high levels of expression. Accurate de novo discovery of splice junctions requires deep sequencing coverage, because each splice site must be covered by multiple reads with adequate overlap with the exons on both sides to allow for alignment. TopHat can also be
used to map reads only to the gene isoforms annotated in the genome reference by specifying the -no-novel-juncs parameter.

An alternative to TopHat/Bowtie for mapping RNA-seq reads to a reference genome is STAR (Spliced Transcripts Alignment to a Reference), developed by Dobin et al. (2013) and used by the ENCODE project. Unlike the split-read segments used by Bowtie, STAR finds splice junctions using a continuous mapping method. Starting from the first base of the read, a seed-matching region is found in the uncompressed suffix array of the reference genome sequence, which is extended to a maximum alignment length (Maximal Mappable Prefix) until the two sequences no longer match (a donor splice site). The unmapped portion of the read is then used to search downstream along the genome for another seed alignment site (acceptor splice site). In a second phase, all seed alignments are stitched together to find the optimal set of splice junctions using a local alignment scoring scheme and parameters that limit minimum and maximum intron size. STAR maps reads up to 50 times faster than TopHat for large RNA-seq data files with similar or better alignment sensitivity and precision. Because of the uncompressed suffix array of the reference genome, STAR requires ~30 GB of RAM for mapping to the human genome. It can be parallelized across multiple CPU cores with shared memory. STAR produces output in SAM format, which can be converted into BAM by SAMtools, which is then compatible with gene expression counting by tools such as Cufflinks or HTSeq.

**POSTALIGNMENT QUALITY CONTROL OF RNA-seq DATA**

After alignment of FASTQ files to a reference genome by a splice-aware aligner, a number of additional quality assessments on the resulting BAM files are possible based on the mapping position of reads with respect to an annotation set of known gene coding regions. The amount or rRNA present in the sample is an important metric for the quality of sample preparation. Large differences in rRNA content among a set of samples can create bias when calculating DE. Many RNA-seq sample preparation methods include poly(A) selection or use poly(T) primers for reverse transcription, leading to 3' bias in the resulting NGS library. This 3' bias can be an important factor in evaluating the abundance of alternate gene isoforms. An RNA sample that has an unusually large percentage of reads that align to intergenic regions may suffer from DNA contamination.

The Picard script CollectRnaSeqMetrics (http://picard.sourceforge.net/picard-metric-definitions.shtml#RnaSeqMetrics) can quickly provide information for each BAM file about the total number of rRNA bases, coding bases, untranslated region (UTR) bases, intron bases, and intergenic bases, as well as an estimate of 5' and 3' bias (based on the 1000 most highly expressed transcripts). Strand specificity of reads is also tested (number of correct and incorrect strand mappings) for strand-specific libraries.
RNA-SeQC (http://www.broadinstitute.org/cancer/cga/rnaseqc_run) is a Java tool that streamlines the Picard RNA-seq analysis method, provides a nice HTML output, and implements it within the GenePattern genomic analysis platform (DeLuca et al. 2012). It has the added option to use BWA to directly compare (1 million) reads to a file of known rRNA sequences, rather than just estimating rRNA content by using mapping positions in the genome alignment BAM file to known rRNA coding regions. Detailed coverage statistics are reported for sets of 1000 highly expressed/1000 median expressed/1000 low nonzero expressed genes, including average transcript coverage and coefficient of variation, coverage of 5′ and 3′ ends, number/total size/percentage of coverage gaps (greater than five bases with zero read coverage), and correlation of reads per kilobase per million (RPKM) between samples. RNA-SeQC requires that the BAM file be preprocessed by sorting with SAMtools sort, create a sequence dictionary with Picard.CreateSequenceDictionary, and mark duplicates with Picard.MarkDuplicates.

A number of other tools have been developed with similar functions. RSeQC (Wang et al. 2012) (http://dldcc-web.brc bcm.edu/lilab/liguow/CGI/rseqc/_build/html/) is a Python package that includes a very comprehensive set of tools for evaluating RNA-seq data sets. Basic tools include sequence quality, base composition, and read duplication (both by identical sequence and by identical genome mapping position). GeneBody_coverage checks for 5′/3′ bias of read mapping across the normalized transcript length and read distribution counts reads mapping to gene features (coding sequence, UTRs, intron, intergenic near coding regions). Other tools include insert size estimation for paired-end reads, strand specificity for stranded reads, counting of known versus novel splice junctions, and an estimation of junction coverage (saturation by subsampling of data). The RPKM_count tool provides a similar function as HTSeq-count, using a set of gene models (in a BED file) to calculate raw read counts and RPKM values for exons and transcripts in a BAM file. Saturation of coverage is estimated by calculating RPKM for subsets of the data.

**QUANTIFICATION OF GENE EXPRESSION**

Once RNA-seq reads are mapped to the reference genome with a spliced alignment tool, gene expression in each sample is measured by counting the number of reads mapping to each gene, exon, transcript, or some other defined set of genome intervals. Cufflinks is a tool (codeveloped with TopHat) that can quantify the sum of bases in all reads in a BAM file that map to a gene across known exons, newly defined exonic regions, and all transcript isoforms (Trapnell et al. 2010). Cufflinks creates a set of transcript isoforms for each gene that include all of the observed splice junctions. This is a fairly complex combinatorial problem because it is not clear which alternative splice events at different sites on the gene go together in the same
transcript. Cufflinks finds a set of paths covering the directed acyclic graph of the read alignments on the genome to build a set of transcripts with minimum cost and maximum matching.

To estimate isoform-level abundances, reads are assigned to individual transcripts, which may be difficult because a read may align to multiple isoforms of the same gene. The use of paired-end reads greatly improves the accuracy of the transcript isoforms and their abundances reported by Cufflinks. Because a paired set of reads must come from the same RNA molecule, exon regions spanned by the pair must be from a single transcript. Cufflinks uses a linear statistical model of paired-end sequencing experiments to derive a likelihood for the abundances of a set of transcripts given a set of fragments. This likelihood function can be shown to have a unique maximum, which Cufflinks finds using a numerical optimization algorithm. The gene level expression is the sum of the expression. In the case of overlapping genes (or genes contained inside introns of other genes) this algorithm can incorrectly assign some reads to the wrong gene.

Cufflinks can also take advantage of the known gene coding regions on a reference genome as supplied in a GTF (or GFF3) file, versions of which are provided for many genomes by UCSC, GenBank, and ENSEMBL. The Cufflinks \(-g/-GTF\)-guide option creates faux reads that cover the coding regions in the GTF file, which will aid in the assembly of known and novel isoforms. The detection of novel isoforms can be completely shut off by using the \(-G/-GTF\) option, which only counts the expression of reads overlapping known exons as described in the reference GTF file.

The output of Cufflinks is a GTF file of predicted transcripts (transcripts.gtf), and tab-delimited tables for gene and isoform expression values. Because Cufflinks creates transcript isoforms on the fly as it quantifies gene expression, the set of transcripts and genes from each sample (each FASTQ file) in an experiment will be different (unless the \(-G\) option is used). An additional script called Cuffmerge is provided that will combine all of the transcripts created by Cufflinks on a related set of RNA-seq samples/data files. It is also possible to provide Cuffmerge with a reference genome GTF file to gracefully merge novel isoforms and known isoforms. Cuffmerge also automatically filters novel transcript isoforms that are probably artifacts.

HTSeq-count is a Python script, part of the HTSeq package, which counts reads in a BAM file of RNA-seq reads aligned to a reference genome. HTSeq-count defines a gene as the union of a set of exon intervals listed in a gene annotation GTF (or GFF) file. This method does not discover novel splice isoforms or new expressed regions and it produces gene expression counts that are not normalized by gene length. The output from HTSeq-count is a tab-delimited table of expression values per gene and per exon. One challenge for this simple method of expression counting is dealing with reads that overlap more than one feature, or features (exons) that overlap each other. The union method (see Fig. 5) is the default. Ambiguous reads...
are not counted. A similar set of base coverage counts per feature can be computed using \textit{BEDTools coverageBed}.

**FILTERING THE DATA**

After quantification, RNA-seq gene expression values are stored in a data matrix (rows for genes, columns for samples). For any RNA-seq experiment, some genes will have zero counts, which just means that in the set of mapped reads, none map to the genomic coordinates assigned to that gene. It is perfectly reasonable to find genes that are not expressed in some cell types under some conditions. In highly differentiated cells, a majority of genes may be completely repressed. It is also possible that for a specific gene in the GTF file, the annotation information is wrong, and no true expressed gene is located at that genomic position. Many of the data analysis methods for finding DE genes do not handle zero values well, giving a divide by zero...
error or undefined value. The removal from the data matrix of genes with zero values for all samples is an obvious first step. Some genes may be expressed (or repressed) in only one biological condition under examination, so it is important to preserve data in which zero counts occur in some conditions but not others. A common compromise is to remove all genes with zero values in the majority of samples under all experimental conditions. Using this method, the number of genes in an RNA-seq data set is often reduced by 20%–30%. Clustering methods such as principal component analysis (PCA) or multidimensional scaling (MDS) may show an improved separation between experimental treatments after removal of rows with zero counts in many samples. The overall average variance computed by several normalization methods is also reduced by removing these extremely low count, highly variable genes. Removing rows from the data matrix has the added advantage of reducing the magnitude of the multiple testing correction, allowing greater sensitivity to find DE genes. To further protect against computational errors, many algorithms replace all remaining zero counts with a small “pseudocount” value. Fold changes and $p$ values of DE statistics computed for genes that have zero counts in all replicates of some conditions are unreliable, but may still be important to call attention to genes with big changes in expression.

**NORMALIZATION**

The fundamental design of gene expression experiments for RNA-seq is very similar to microarrays: comparison of two or more biological conditions with several replicate samples for each condition. The most common goal of RNA-seq experiments is to measure changes in genome-wide gene expression (DE). Statistical analysis is performed to identify individual genes (and functional groups of genes) that have changed significantly in abundance across experimental conditions. Although the technical reproducibility of RNA-seq is very good (same RNA sample sequenced several times), the underlying variability of abundance of RNA molecules in biological samples is high, so it is necessary to use replicate samples to study changes in gene expression across experimental treatments.

Although the sensitivity and specificity of RNA-seq is very good, there are some informatics issues associated with accurate quantitation of differential gene expression. Longer mRNA transcripts produce more reads because they contribute more fragments to the sequencing library, creating greater statistical power to detect DE in these longer transcripts. Sequence-specific bias may also occur in various steps of the sample preparation, sequencing, and alignment. Fortunately, many of these biases cancel out when calculating the DE between biological conditions for the same set of genes (i.e., guanine-cytosine [GC] bias, DNase hypersensitivity, or resistance to shearing, etc.). However, sample prep and machine-specific sequencing issues often lead to differences in the total yield of sequence reads in each sample.
of an experiment. This creates a need for scaling or normalization methods across samples to make accurate comparisons and calculate DE scores for each gene. Li et al. (2012) and others have clearly showed that the methods for normalization of RNA-seq expression values have a large impact on the calculations of DE (Hoffmann et al. 2002). Several software packages, such as edgeR, allow the user to apply different normalizations to the data before calculation of DE, which can produce strikingly different lists of DE genes. Many of the software packages for DE analysis require nonnormalized count data as input and include their own built-in normalizations. Overall, it is clear that the normalization method and calculations of DE are inextricably linked, and software packages must handle these two operations jointly. It is generally not feasible to mix and match normalization from one algorithm with a significance test from another.

Read counts for each gene can be normalized for each sample as counts per million (cpm) by simply dividing by the total number of reads (or mapped reads) in each data file, which can be called “total count normalization.” If some genes are very highly expressed in one treatment, the total number of reads in the FASTQ file may be reduced for genes with constant expression levels, so when scaled to cpm, they appear to be down-regulated. Within a single sample, longer genes will produce more fragments and therefore more reads than short genes. Although this obvious bias should cancel out when gene expression is compared between samples, genes with more counts will have greater statistical power to detect DE. The RPKM normalization method developed by the Wold Lab at Caltech (Mortazavi et al. 2008) has become the de facto standard. RPKM normalization divides the read count for each gene by the length of the RefSeq transcript for that gene and then scales all read counts per million reads in each data file (aligned reads in a BAM file). The Cufflinks program uses the term FPKM to refer to normalization of paired-end data, in which each read pair is counted as one expression unit that represents a single cloned cDNA fragment. However, RPKM normalization can reverse the bias in favor of long genes and allow more short genes to be found as differentially expressed by some statistical methods.

If a few genes are very highly expressed in some samples, RPKM normalized values will be biased, so a quantile-based normalization method may provide a more accurate measurement of DE for low copy number genes (Bullard et al. 2010). The baySeq package (Hardcastle and Kelly 2010) uses an upper quartile normalization, dividing all expression levels by the gene counts at the 75th percentile in each sample. The PoissonSeq package (Li et al. 2012) defines a set of genes that is least differentiated between two conditions (expressed at a constant level), and scales each sample to the ratio of counts for these genes. This is conceptually similar to using a set of known invariant (i.e., housekeeping genes) or spike-in controls for normalization. The limma package in R/Bioconductor, which was originally developed for microarray analysis, has added a function called voom to provide normalization.
of RNA-seq data with a LOWESS regression (Law et al. 2014), normalizing genes in bins with similar expression levels. In DESeq (Anders and Huber 2010), the sequencing depth is normalized by the median ratio among all genes. DESeq computes a scaling factor for a given sample by computing the median of the ratio, for each gene, of its read count over its geometric mean across all samples. Trimmed mean of \( M \) values (TMM) normalization (Robinson and Oshlack 2010) estimates sequencing depth by a scaling factor computed from a set of genes that exclude genes with high fold change across treatments or for which the average expression is very large.

All of these normalization methods rely on assumptions that for any comparisons among samples, cells produce similar total amounts of RNA, most genes are not differentially expressed, and that the numbers of up- and down-regulated genes are similar. In a detailed comparison among normalization methods using both real and simulated data sets, Dillies et al. (2013) find that the cpm (total count) and RPKM methods "are ineffective and should be definitively abandoned in the context of differential analysis." Quantile normalization relies too strongly on the assumption that all samples have identical relative read count distributions among genes, which is frequently not the case, leading to an increase in within-condition variability. The best results were produced by the TMM and DESeq methods, which maintained a reasonable false-positive rate without loss of power, even when data sets contained some genes with very high read counts in some conditions.

**DIFFERENTIAL EXPRESSION**

The calculation of DE and the associated statistical models for establishing significance is currently one of the most controversial areas in NGS informatics, with many different software packages offering different algorithms and no consensus exists as to best practice methods. This diversity of methods is particularly problematic for investigators because the list of “significant DE genes” is often the key product of a complex and costly experiment, yet this list can be highly variable dependent on what software and parameters are used for the data analysis. Various software packages display large differences in performance in how they score DE for highly expressed versus low-expressed genes (sensitivity) and how they handle genes that are expressed at undetectable levels in some conditions but at higher levels in other conditions (are low counts similar or very different than no counts at all?). Benchmarking studies have compared some of the available software for quantification (Chandramohan et al. 2013), normalization (Dillies et al. 2013), and DE analysis of RNA-seq (Rapaport et al. 2013; Soneson and Delorenzi 2013), but no general consensus has emerged as to the best analysis method for any particular type of RNA-seq data. Benchmarking itself is controversial. What are the true expression values of the genes in the test samples, and what levels of DE should be declared significant? If the
benchmark samples are well characterized (or simulated), are they representative of the variance and bias that are found in the samples that another investigator might generate in a completely different type of study?

A number of statistical methods have previously been developed for calculating DE in microarray experiments, but there are some important fundamental differences between measuring gene expression by array versus by RNA-seq. Array measurements are based on the detection of fluorescence signal, in which the intensity signal for each probe on an array is (relatively) independent of other signals. In RNA-seq all expression levels are relative—each gene contains reads that are a fraction of the total number of reads in the BAM file, so if a few genes are very highly expressed, they dominate the library of sequenced fragments, and there are fewer reads available for the other genes. So far, no statistical method has fully captured the implications of this zero-sum situation in its calculation of DE.

Expression differences can be calculated as the simple fold change ratio for each gene between conditions. If replicate samples are taken for each condition, then the ratio of average expression for each gene per condition is the average fold change. Fold change expressed as a simple ratio is an awkward measurement, because up-regulated genes can have essentially unlimited large positive values, whereas down-regulated genes are squeezed between 1 and 0. The log$_2$ of the fold change ratio gives a value that is positive for up-regulated and negative for down-regulated genes; it is symmetric around 0 and reduces the scale. Fold change values are supported by a measure of statistical significance, which can be summarized as a simple question: Are the expression changes between conditions larger than the variability of measurements within each condition? Significant changes for each gene can be calculated as a p value by using a t-test (Bloom et al. 2009); however, RNA-seq data violates the assumptions of this test, because the expression values are not normally distributed and the variance between treatments is often overdispersed. A number of models of RNA-seq data assume that the counts for each gene are drawn from a Poisson distribution, but this implies that the variance is equal to the mean. The Poisson distribution may be valid for technical replicates; however, the observed variance of RNA-seq counts for a gene across different conditions is larger than the mean. A number of software packages model the dispersion of RNA-seq expression values as a negative binomial distribution, which includes a factor for dispersion. A specific challenge for RNA-seq data is that the number of replicates is generally quite small, so it is difficult to accurately estimate the variance of expression for each gene in each biological condition. Many algorithms assume that the variance for each gene is the same in each experimental condition and that the variance is similar for genes expressed at similar levels, both of which may not be true.

Because RNA-seq is a genome-wide analysis of gene expression, fold change and statistical significance of DE are calculated for all genes simultaneously, so the p values must be corrected for multiple comparisons. The FDR method (Benjamini and
Hochberg 1995) provides an attractive measure of control for multiple testing in genomic settings, but this method depends on accurate p values. When sample size is small or most of the DE genes are changing in the same direction (all up or all down), then p values and FDR estimates are inaccurate. This may lead to no multitest-corrected DE genes being found, or many genes falsely called as DE and many true DE genes being missed (Soneson and Delorenzi 2013).

**Cufflinks** calculates DE (with the Cuffdiff2 program) using a permutation sampling method to improve the estimation of variance for each gene in each condition, assuming a β-negative binomial model for counts. Cuffdiff uses a similar scaling factor procedure as DESeq to account for the different sequencing depths and the Benjamini–Hochberg procedure to control the FDR. Genes with FDR adjusted p values (called q values) \( \leq 0.5 \) are marked as “significant: yes” in the gene_exp.diff output file. Similar output files are created for isoforms, tss_group (expression differences between groups of isoforms sharing the same transcription start site), and cds (expression differences between isoforms that contain different coding sequence). Files are also produced that summarize overall differential splicing (**splicing.diff**) and differential promoter use (**promoters.diff**) for all of the genes. Several investigators have noted that CuffDiff produces an excess of false-positive DE calls with small p values for genes with low absolute expression or low fold change (Anders et al. 2012; Rapaport et al. 2013). A practical response to this problem is to use a fold change threshold as a filter for CuffDiff results.

Several statistical methods rely on “borrowing” information for variance (dispersion) across groups of genes. The **edgeR** package (an R/Bioconductor module) implements methods to “shrink” the dispersion of individual genes toward an overall average dispersion for all of the genes (excluding the most highly expressed and most DE), which relies on the assumption that most genes in an experiment are not differentially expressed (Robinson et al. 2010). The relative weight used for single gene variance versus shared dispersion is an adjustable parameter. The TMM normalization method is used. edgeR determines DE using empirical Bayes estimation and Fisher’s exact test based on a negative binomial model. Multiple testing is controlled by the Benjamini–Hochberg FDR method. Under many conditions, edgeR is too liberal in its assignment of significance, finding false-positive DE genes not found by other methods.

**DESeq** (Anders and Huber 2010) also uses the negative binomial model for gene count dispersion, but dynamically estimates mean and variance from the data. The DESeq normalization method produces similar results as the TMM method used by edgeR. DESeq tends to be more conservative in its significance scores than many other methods, producing fewer false positives, but also finding fewer true DE genes.

**baySeq** (Hardcastle et al. 2010) implements an empirical Bayesian method to estimate the likelihood of DE, assuming a negative binomial distribution of gene count values. The output of this model is a set of posterior probabilities for
significance rather than a \( p \) value, and the multiple testing correction reports a Bayesian FDR estimate. In comparative analysis, baySeq found fewer DE genes at all expression levels than other software methods.

**PoissonSeq** (Li et al. 2012) assumes the gene expression count data are distributed on a Poisson log-linear model and overdispersion of the data is handled by a power transformation. PoissonSeq calculates the FDR using a pooled permutation distribution based on genes that show low fold change across conditions (excluding genes that are likely to be differentially expressed).

**NOIseq** (Tarazona et al. 2011) is a nonparametric method that estimates variance (noise) by comparing fold change and absolute count differences within replicates of a condition and between conditions. NOIseq uses RPKM normalization to scale read counts among samples.

**SAMseq** is a nonparametric method that is more robust with respect to outliers (genes highly expressed in just one sample). After normalization and data transformation, it uses the popular SAMr method that was developed for DE analysis of microarrays. SAMseq is based on Wilcoxon rank statistic and uses a unique resampling procedure to account for different sequencing depths among samples and a permutation-based method to estimate FDR. This method will overestimate FDR for samples sizes smaller than five per treatment and may not find any DE genes if only two replicates are used. SAMseq found significantly more DE genes with low fold change and high expression level than many other methods, but such small expression changes may not be biologically important. SAMseq is also applicable to multiclass, quantitative, time course, and survival outcome experimental designs.

**limma** implements a linear model developed for microarray expression analysis. RNA-seq data is normalized with a method called *voom*, which uses a LOWESS regression to estimate the mean-variance relation and transforms the read counts to log form for linear modeling with an empirical Bayes analysis method. \( p \) values are computed with a modified \( t \)-test that uses a value for variance for each gene (standard error) that is shrunk by a factor estimated from all genes. This method performs well across a wide range of real and simulated data sets.

Given the diversity of algorithmic approaches for these different software packages, each designed to compensate for specific bias characteristics of RNA-seq data, it is not surprising that they provide somewhat different lists of DE genes from a single data set. The quality, validity, and correlation of the computed scores from different software will also vary across various types of experimental data with differing numbers of replicates, in which the numbers of DE genes and the magnitude of expression changes are dramatically different. Software that performs well to find DE genes in brain versus liver may not be the best choice to find DE genes in a single gene knockout study of cultured cells. Although it is not a best practice or SOP, many investigators use several software methods to score expression differences in a data set and then evaluate the genes in the DE lists with common sense criteria such as
absolute expression level, fold change, presence of an outlier score in just one sample, and biological relevance of the gene’s function. (See Table 2.)

**ALTERNATIVE SPlicing**

RNA-seq expression data can be calculated for individual exons. This data can then be further analyzed to identify exons that show changes in expression across biological conditions that differ from the changes in expression levels of the entire gene (exon-specific expression). DE of a particular exon across experimental conditions is an indication of biologically regulated alternative splicing.

The detection and quantitation of alternative splicing events in an RNA-seq data set is a challenging bioinformatics problem. The CASAVA method of mapping reads to exons relies on a database of “nonoverlapping” exons that combines overlapping exons of different length (alternative 3’ or 5’ splice sites) and ignores situations in which an entire intron is alternatively spliced or retained. In fact, one of the most challenging aspects of studying alternative splicing is the definition of the alternative splice events, and building a database of valid alternatively spliced transcripts or

**TABLE 2. Software packages for detecting DE in RNA-seq data**

<table>
<thead>
<tr>
<th>Method</th>
<th>Normalization</th>
<th>Variance model</th>
<th>DE test</th>
</tr>
</thead>
<tbody>
<tr>
<td>edgeR</td>
<td>TMM/upper quartile/RLE</td>
<td>Negative binomial distribution</td>
<td>Exact test</td>
</tr>
<tr>
<td>DESeq</td>
<td>Median of DESeq size factors</td>
<td>Negative binomial distribution</td>
<td>Exact test</td>
</tr>
<tr>
<td>PoissonSeq</td>
<td>Power transformation</td>
<td>Poisson log-linear model</td>
<td></td>
</tr>
<tr>
<td>baySeq</td>
<td>Scaling factors (quantile/TMM/total)</td>
<td>Negative binomial distribution</td>
<td>Empirical Bayesian classification of differentially and non-differentially expressed genes</td>
</tr>
<tr>
<td>NOIseq</td>
<td>RPKM/TMM/upper quartile</td>
<td>Nonparametric method</td>
<td>Contrasts fold changes and absolute differences within versus between conditions</td>
</tr>
<tr>
<td>SAMseq (samr)</td>
<td>SAMseq specialized method based on the mean read count over the null features of the data set</td>
<td>Nonparametric method</td>
<td>Wilcoxon rank statistic and a resampling strategy</td>
</tr>
<tr>
<td>Limma</td>
<td>TMM</td>
<td>Voom transformation of counts</td>
<td>Empirical Bayes method</td>
</tr>
<tr>
<td>Cuffdiff 2 (Cufflinks)</td>
<td>Geometric (DESeq-like)/quantile/classic-FPKM</td>
<td>β-negative binomial distribution</td>
<td>t-test</td>
</tr>
</tbody>
</table>

Adapted from Seyednasrollah et al. 2013, with permission from Oxford University Press.
isoforms. Wang et al. (2008) defined eight types of alternative transcription events: skipped exon, retained intron, alternative 5′ splice site, alternative 3′ splice site, mutually exclusive exons, alternative first exon, alternative last exon, and tandem UTRs (Fig. 6). Any method of quantifying alternative splicing must first define the set of alternative transcripts possible for each gene. Methods that rely on a database of known transcript annotations are very dependent on the database used. The RefSeq database is very stringent in its definition of alternative transcripts, and typically only includes four to five transcripts per gene, whereas the ENSEMBL database contains more transcript isoforms. In contrast, the NCBI AceView database (Thierry-Mieg and Thierry-Mieg 2006) shows evidence for many more splice variants per gene (Fig. 7).

The analysis of alternative splicing in RNA-seq data presents major challenges in transcript assembly and abundance estimation, arising from the ambiguous assignment of reads to isoforms. Alternatively spliced transcripts for a single gene often share many exons, so reads that map to shared exons cannot be unambiguously assigned to one particular isoform. Only reads that span a specific, previously annotated, splice junction or reads that map uniquely to just one isoform (i.e., a cassette exon) are unambiguous. Several software methods have been developed to analyze
RNA-seq data that are intended to provide estimates of alternative splicing events. The Cufflinks program, developed by Cole Trapnell and Steven Salzberg at University of Maryland in collaboration with members of the Wold Lab at Caltech and others (Trapnell et al. 2010), builds sets of synthetic alternative transcripts directly from the genome mapped RNA-seq reads, and then quantitates the abundance of each of these transcripts. Cufflinks was designed to work with the Bowtie and TopHat sequence alignment programs, but it can use any file of RNA-seq reads aligned to a reference genome in SAM or BAM format, such as that produced by the BWA aligner, STAR, or the Illumina ELAND/CASAVA package. Cufflinks works best with paired-end reads, in which it can calculate the size of the RNA fragments used for sequencing, which in turn can provide a great deal of additional data about splice isoforms. Cufflinks includes the program CuffDiff, which can find significant changes in transcript expression, splicing, and promoter use (i.e., alternative transcription start sites). Cufflinks requires deep coverage of the transcriptome without extensive 3' bias to work well.

Current methods of detection for alternative splicing can provide a good picture of genome-wide splicing activity in particular cell types, and can be used to test for

changes in isoform abundance for predefined specific genes across samples. However, these methods tend to have a high FDR when used to screen the entire transcriptome to find any genome-wide changes in isoform use across a set of biological conditions. In other words, if Cufflinks or other tools are used to define a complete set of differentially spliced gene isoforms across an experimental condition—and this list is sorted by \( p \) value, fold change, or some other statistic—many of the top ranking genes are likely to be false positives.

**DOWNSTREAM ANALYSIS**

Once genes have been identified as differentially expressed (or alternatively spliced) by appropriate statistical tests of RNA-seq data, downstream analysis can proceed using similar tools as for microarray gene expression studies. Lists of differentially regulated genes can be analyzed for functional correlation by enrichment of gene ontology (GO) terms using hypergeometric tests or by gene set enrichment analysis (GSEA). Gene functions are defined by the Genome Ontology Consortium using experimental data from model systems, such as *Drosophila*, and then mapping these functions to orthologous genes in other species.

The database for annotation, visualization, and integrated discovery (DAVID) is a web-based tool (http://david.abcc.ncifcrf.gov) that automates functional studies of experimentally derived lists of DE genes. A Nature Protocols paper (Huang et al. 2009) provides a detailed tutorial for the use of DAVID to analyze lists of DE genes. The basic concept is simple: A list of DE genes is mapped to known functions, then the list is examined for statistically overrepresented functions (i.e., enrichment). The DAVID tool is widely used for high-throughput genomic analysis and has been cited in more than 10,000 journal articles.

DAVID maps gene lists to known functions using any of several different identifiers including GenBank accession numbers, RefSeq IDs, or ENSEMBL IDs. Other types of gene IDs may be resolved by the DAVID gene ID conversion tool. The identifiers in gene lists produced by any of the quantification and DE analysis software originate in the GTF or GFF annotation file used to define intervals on the reference genome. DAVID includes annotation categories from more than 40 sources of reference data including GO terms, protein–protein interactions, protein functional domains, disease associations, KEGG and BioCarta pathways, sequence features, homology, gene functional summaries, gene tissue expression, and literature. Gene IDs and functional mappings are very straightforward for human, mouse, fruit fly, and other well-studied systems, but can be more difficult for species with incomplete draft genomes, and very difficult for novel species with little definitive annotation.

The basis for functional enrichment analysis is the idea that genes from a specific pathway will be coregulated, and therefore appear together on a list of DE genes when expression of that pathway is significantly altered in an experiment. The enrichment
of a specific functional term is compared to a background value—the expected number of genes from that category that would occur in an equal size list drawn at random from the genome. DAVID compares the value for the gene list with the background value using a modified Fisher’s exact test, known as EASE, and multiple test correction by the Benjamini–Hochberg FDR method. DAVID works best with gene lists that contain approximately 100 to 2000 genes. Lists smaller than 100 may not have enough genes in specific functional classes to provide statistical power to find significant enrichment. Lists larger than 2000 will have genes in many categories and will have less significant p values owing to multistest correction. In addition to enrichment scores for individual annotation terms, DAVID uses fuzzy clustering to group sets of related significant terms into functional annotation clusters that represent similar biological processes.

GSEA is another approach to functional annotation. One drawback of the DAVID enrichment analysis method is that it is highly dependent on the specific RNA-seq read counting conventions, statistical methods, and significance thresholds used to generate the list of DE genes. Additional functionally important genes may be regulated, but not meet the statistical cutoffs to be declared differentially expressed. GSEA takes as input the expression values for the entire set of genes for which expression data is collected, which for RNA-seq is the entire transcriptome. The complete data matrix of gene expression values is processed to produce a ranked and sorted list based on a statistic for DE. A minimum of three replicates for each experimental condition are required for GSEA to compute the DE metric. The default metric, called Signal2Noise, is simply the difference of the means (μ) for two experimental conditions scaled by the standard deviations (σ): (μa − μb)/(σa + σb). Alternate DE metrics can be used such as the absolute difference of means (μa − μb), fold change (μa/μb), log2-fold ratio, or t-test. The median can be used in place of the mean for any of these metrics. The GSEA software was designed to process microarray gene expression data, and the specific issues discussed above for DE analysis of RNA-seq data were not considered. In the GSEA manual, it is recommended to preprocess a gene expression data matrix with a DE tool designed for RNA-seq (such as Cufflinks, edgeR, DESeq, etc.), and then use the p values to rank the gene list in the GSEAPreranked module. The GSEA method is robust with respect to the exact values assigned to the DE of each gene, and it pays no attention to significance thresholds; the only factor that matters is the rank of the gene on the list from most highly up-regulated to most strongly down-regulated.

Once the ranked and sorted list of genes is produced, it is compared to any of a list of predefined “gene sets” that represent functional classes and biological pathways. The GSEA software uses gene sets stored in a Molecular Signatures Database (MSigDB), which is curated by the GSEA team from GO, BioCarta, KEGG, and other data sources. All gene sets in MSigDB consist of human gene symbols. Just as for analysis with DAVID, it is necessary to match up the gene identifiers in an
experimental expression data set with the gene functions in MSigDB. GSEA automatically recognizes Affymetrix array probe names and HUGO gene symbols. For expression data with any other gene identifiers (or from any other species), the user is required to convert to equivalent human gene IDs. GSEA will also accept custom gene sets that contain any gene identifiers (and functional annotations) that are created by the user.

The core function of GSEA is to test if a ranked gene list is enriched for up-regulated (or down-regulated) genes from a specific gene set. The statistical method is testing a simple question: Are the members of a functional set of genes ranked higher (or lower) in the ranked expression list than would be expected by chance? The primary result of the GSEA is the enrichment score (ES), which reflects the degree to which a gene set is overrepresented at the top or bottom of a ranked list of genes, compared to a random ranking. The ES score is supported by a FDR calculation created by permuting the samples (assigning the samples randomly to experimental conditions and recomputing the DE score and the ranks). The GSEAP-reranked method computes FDR by permuting the gene sets rather than the samples, which is a less stringent method.

GSEA software (http://www.broadinstitute.org/gsea) is available as a free downloadable Java application for Windows/Mac/Linux computers, as an R module, or as a module in the GenePattern platform (which itself is available as a web application or as a locally installed tool).

There is a great value in integration of RNA-seq data with other kinds of count-based genomic data that can be derived from NGS platforms. In particular, integration of RNA-seq gene expression measurements with ChIP-seq measurements of transcription factor binding and/or histone modification patterns can provide a deeper and more nuanced picture of gene expression activity. It is of great practical significance to identify regulatory or epigenetic factors that produce an observable change in gene expression (see Fig. 8). When coupled with genomic sequencing (or genotyping), RNA-seq can be used to measure allele-specific expression.

**TUTORIAL**

RNA-seq analysis is a challenging computational problem—both in terms of the complexity of the algorithms and the usability of the current generation of software, as well as the requirements for high-powered computers to map and summarize millions of reads across genes, exons, transcripts, etc. There are some commercial software packages that simplify the task of calculating RNA-seq gene expression from FASTQ files. Many core laboratories and sequencing vendors offer standard RNA-seq data analysis pipelines either included in the cost of sequencing or for a standard analysis fee. The Galaxy web service offers a simplified interface to the TopHat/Cufflinks RNA-seq analysis toolkit; however, it takes a substantial amount of time to
A combined view of RNA-seq and ChIP-seq data illustrates the effects of histone modification on gene expression and the interaction of histones with RNA-polymerase. (Courtesy of Phillip Ross Smith.)
upload and execute these computations on the free version of Galaxy. Anyone planning on analyzing a substantial amount of RNA-seq samples will either need access to a powerful Unix server where they will learn to install and run TopHat/Cufflinks (or other similar software), install a local copy of Galaxy to use the simplified interface with high-performance computing power (work with your local IT on this one), or use a “cloud”-based service where you rent computing power by the hour on a system with preinstalled tools (see Chapter 12). The Galaxy Cloud project (http://wiki.g2.bx.psu.edu/Admin/Cloud) makes it very easy to rent access to your own private copy of Galaxy set up for you on the Amazon Elastic Compute Cloud (EC2).

The Galaxy Project team have created a tutorial for RNA-seq analysis that includes QC, trimming of low-quality portions of sequence reads, mapping of reads to a reference genome, quantifying reads per transcript, and comparison of two samples to find differential gene expression (Goecks 2011). This is a time-consuming tutorial, even though the training data is already uploaded and stored on the Galaxy server and the training data set represents only a tiny fraction of an actual NGS RNA-seq run. Once again, realize that it is not feasible to analyze substantial amounts of data using this free server.

Galaxy RNA-seq Analysis Exercise

Galaxy provides multiple tools for performing RNA-seq analysis (http://main.g2.bx.psu.edu/u/jeremy/p/galaxy-rna-seq-analysis-exercise). This exercise introduces these tools and guides use of these tools on some example data sets; prominent RNA-seq tools include TopHat and Cufflinks. Familiarity with Galaxy and the general concepts of RNA-seq analysis are useful for understanding this exercise. This exercise should take 1–2 hours.

Below are small samples of data sets from the ENCODE Caltech RNA-seq track; specifically, the data sets are single 75-bp reads from the h1-hESC and GM12878 cell lines (see Fig. 9). The sampled reads map mostly to chr19. Import the data sets to your history by clicking on the green-plus icon labeled “Import.”

http://main.g2.bx.psu.edu/datasets/7f717288ba4277c6/imp
http://main.g2.bx.psu.edu/datasets/257ca40a619a8591/imp

Understanding and Preprocessing the Reads

You should understand the reads a bit before analyzing them. Preprocessing may be needed as well.

Step 1: Compute statistics and create a boxplot of base pair quality scores for each set of reads using the [NGS: QC and manipulation >] FASTQ Summary Statistics
tool and then plot the output using the [Graph/Display Data >] Boxplot. Often, it is useful to trim reads to remove base positions that have a low median (or bottom quartile) score. For this exercise, assume a median quality score of below 15 to be unusable (see Fig. 10). Given this criterion, is trimming needed for the data sets? If so, which base pairs should be trimmed?

**Step 2:** If necessary, trim the reads based on your answers to Step 1 using [NGS: QC and manipulation >] FASTQ Trimmer.

**Map Processed Reads**

The next step is mapping the processed reads to the genome. The major challenge when mapping RNA-seq reads is that the reads, because they come from RNA, often cross splice junction boundaries; splice junctions are not present in a genome’s sequence, and hence typical NGS mappers such as Bowtie and BWA are not ideal without modifying the genome sequence. Instead, it is better to use a mapper such as TopHat that is designed to map RNA-seq reads.

**Step 1:** Use the [NGS: RNA Analysis >] TopHat tool to map RNA-seq reads to the hg18 genome build. This will take about 10 minutes. Look at the documentation to understand the two data sets that TopHat produces. How many splice junctions did TopHat find? Are most splice junctions supported by (i.e., spanned by) more or less than 10 reads? (Hint: The score column in the splice junctions data set is useful for answering this question.)

**Step 2:** To view TopHat’s output, create a simple Galaxy visualization by selecting Visualization > New Track Browser from the main Galaxy menu at the top. After creating the visualization, add data sets to your visualization by clicking on the Options→Add Tracks option. Add the “accepted hits” BAM data set and the “splice junctions” data set produced by TopHat, and also the UCSC human Reference Genes...
FIGURE 10. Galaxy boxplot of sequence quality scores showing severe decline in quality at the 3’ end of the reads.
(refGene chr19) for chromosome 19 (you will need to import the data set before you can add it to your visualization):

http://main.g2.bx.psu.edu/datasets/965c374b65239597/imp

Navigate to chr19 using the select box at the top of visualization and look at the data. This visualization makes clear how little of the data you are working with. Zoom in to view the data at the beginning of the chromosome. You can zoom in by (a) double-clicking anywhere on the visualization to zoom in on that area or (b) dragging on the base number area at the top of the visualization to create a zoom area.

You should be able to see (1) the reads mapped by TopHat, including splice junctions; (2) just the splice junctions produced by TopHat; and (3) how TopHat’s reads and junction correspond to UCSC’s RefSeq gene track (Fig. 11). Find an example of a splice junction between two known exons, and find an example in which a splice junction should be found but is not.

**Assemble and Analyze Transcripts**

After mapping the reads, the next step is to assemble the reads into complete transcripts that can be analyzed for DE and phenomena such as splicing events and transcriptional start sites.

**Step 1:** Run [NGS: RNA Analysis >] Cufflinks on each BAM data set produced by TopHat to perform de novo transcript assembly. Here is the documentation for the data sets that Cufflinks produces. How can you tell when a transcript has multiple exons?

**Step 2:** Run [NGS: RNA Analysis >] Cuffcompare on the assembled transcripts and use the UCSC RefSeq Genes data set as the reference annotation. Cuffcompare requires that the reference annotation be in GTF format, so use this version of the UCSC RefSeq Genes.

Find some transcripts that appear in both samples and have FPKM confidence bands that do not overlap. You can find this information by looking at the “transcript tracking” data set produced by Cuffcompare and reading the Cuffcompare documentation to understand the data in this data set.

**Step 3:** Add the Cufflinks’ assembled transcripts data sets to the visualization you created earlier to view the transcripts alongside the mapped reads, junctions, and reference genes. Can you find examples in which Cufflinks/Cuffcompare assembled a complete or almost complete transcript?

**Step 4:** Run [NGS: RNA Analysis >] Cuffdiff on (1) the combined transcripts produced by Cuffcompare and (2) TopHat’s accepted hits data sets for each data set. Cuffdiff produces quite a few output data sets; you will want to browse the Cuffdiff documentation to get a sense of what they do. Look at the isoform
expression data set—are there any significant isoform expression differences between the two samples? Look at the isoform FPKM tracking data set—find an entry for a novel isoform and an entry for an isoform that matches a reference isoform. What is the nearest gene and transcription start site for each entry? (Hint: You will need to understand the class codes, which are explained in the Cuffcompare documentation.)

**FIGURE 11.** Galaxy visualization of RNA-seq reads aligned by TopHat across splice junctions of refSeq gene NM_006339 on human hg18 build of chromosome 19.
On Your Own:

- Identify all novel splice junctions and transcript isoforms in each set of transcripts.
- Rerun Step 3, but assemble transcripts using the UCSC RefSeq genes as a reference. Find differences between de novo transcript assembly/analysis and reference-guided transcript assembly/analysis.

REFERENCES


WWW RESOURCES


http://www.broadinstitute.org/cancer/cga/rnaseqc_run RNA-SeQC home page, CGA (Cancer Genome Analysis), Broad Institute.

http://www.broadinstitute.org/gsea Gene set enrichment analysis (GSEA) is a computational method that determines whether an a priori defined set of genes shows statistically significant, concordant differences between two biological states (e.g., phenotypes).
http://david.abcc.ncifcrf.gov  DAVID (The database for annotation, visualization and integrated discovery) Bioinformatics Resources 6.7, NIAID, NIH.


http://main.g2.bx.psu.edu/u/jeremy/p/galaxy-rna-seq-analysis-exercise  Galaxy RNA-seq analysis exercise.


http://picard.sourceforge.net/picard-metric-definitions.shtml#RnaSeqMetrics

http://wiki.g2.bx.psu.edu/Admin/Cloud  Galaxy CloudMan.