1

Gene Expression Analysis Using RNA-Seq from Organisms Lacking Substantial Genomic Resources

Yingjia Shen, Tzintzuni Garcia and Ronald B. Walter
Texas State University, USA

1. Introduction

Development of massively parallel "next generation" sequencing technology (NGS) has dramatically revolutionized biological studies. Among the many applications of NGS, RNA-Seq is one of the most important uses of this technology. RNA-Seq enables investigators to accurately probe the current state of a transcriptome and assess many biologically important issues, such as; gene expression levels, differential splicing events, and allele-specific gene expression. Compared with previous technologies (e.g., microarrays, etc.) NGS has the clear advantage of not being limited to experimental systems having well characterized genomes or transcript sequence libraries. This positions RNA-seq approaches as important and versatile techniques for experimental systems and species where specific genetic information may be limited or altogether lacking.

A major goal of most transcriptomic studies is the identification and characterization of all transcripts within a developmental stage or specific tissue. NGS techniques have made the massive amount of data required to carry out such studies both inexpensive and available to an unprecedented extent. Clever computer algorithms have made the assembly of these massive data sets the work of one or two people with reasonably powerful workstations or a moderate analytical server.

Once a reference transcriptome has been assembled, analyses can be carried out that involve several steps, such as; mapping short sequence reads to transcriptome, quantifying the abundance of genes or gene sets, and comparing differential expression patterns among all samples. Herein we outline the processes from obtaining raw short read data to advanced comparative gene expression analysis and we review bioinformatic programs currently available, such as Tophat, Cufflinks, DESeq, that are specifically designed to address each of the above steps. We will discuss both accuracy and ease of use of these tools by biologists beginning to pursue these types of analyses. In addition to individual programs, we will also discuss integration of multiple programs into pipelines for more rapid and complete expression analyses. Overall, the future applications of RNA-Seq will open new avenues for transcriptome analyses of less well-studied and/or wild caught species that could not have previously been approached. This will yield a wealth of new comparative data highlighting the many ways plants and animals have developed to survive in this rapidly changing environment.

2. *De Novo* sequence assembly and expression analysis with NGS data

There are many phases to an NGS research project where the end goal is expression analysis in a non-model organism. This chapter is dedicated to the many phases and options available to the researcher. In general however, bioinformatic analyses at some point begin with gathering raw sequence data from a biological sample of interest and having it sequenced. The raw data will often need to be filtered for quality. If any pre-existing sequences are available from a closely related species, their use as a reference should be considered, but is not necessary. Assembling the short reads derived from one or more of the NGS platforms comes next, but should not be considered a definitive, terminal process. Most frequently assembly of short read data entails an iterative refinement phase in which a wide range of parameters are modified in the search for a sufficiently contiguous and complete assembly. Analyzing the assembly can entail searching for signatures of assembly errors and trying to identify the assembled contigs. Once a satisfactory group of transcripts is produced they are locked for the expression level analysis. Mapping the short reads to the assembled transcripts is the first step in assessing gene expression levels. The next is determining the expression levels of each contig based on the number of short reads mapped to it. Generally a comparative gene expression analysis will follow in which two or more samples are compared and alternate regulation patterns or profiles determined. We end the chapter with a specialized comparative expression study in F1 hybrid organisms in which differential expression may reveal evolutionary divergence in gene regulation mechanisms.

2.1 Next-generation sequencing

Next-generation sequencing (NGS) techniques produce millions of reads per run but each read may be as short as 25 bp. Using NGS allows one to apply complex samples (i.e., total DNA or RNA libraries) on the NGS instrument. These mixed samples contain fragments of larger molecule targets sheared to some pre-set fragment length distribution. NGS techniques allow the sequencing of completely unknown samples in a massively parallel fashion. In order to perform massively parallel sequencing most NGS instruments require a run time of days to weeks in carefully controlled conditions for complete data acquisition. There are many competing technologies, and new challengers are in constant development to increase both the speed and quantity of NGS per sample run. It is beyond the scope of this chapter to examine all of the current and upcoming techniques so we will briefly focus on two most common NGS instruments currently in use: the Illumina Genome Analyzer and ABI SOLiD platforms. Each of these platforms has its strengths and weaknesses that are very important to understand when designing research strategies.

2.1.1 The ABI SOLiD platform

The SOLiD system produces short sequencing lengths (i.e., termed “reads”) ranging from 35 to 75 bp and has run times of between one and seven days depending on the amount and type of reads desired. Typically, instruments will have 6 or 12 independent lanes available per run and samples can be multiplexed in each of those for up to 96 unique barcodes. Product literature states the daily sequencing throughput is between 10-30 Gbp.

The SOLiD sequencing process begins by fragmenting high molecular weight DNA into smaller fragments to be sequenced (Fig 1A). Fragments are size selected in a narrow range, typically around 200 bp, and primers are ligated to both ends of the fragments (Fig 1A).

Glass beads coated with complimentary primers are mixed with the fragments (Fig 1A) and emulsified in such manner that an aqueous droplet will contain a single bead and a single fragment along with the biochemistry necessary for PCR (Fig 1B). Several rounds of emulsion PCR later each bead is coated with sequences identical to the original fragment (Fig 1B). The DNA coated beads are then released from the emulsion, and washed into tiny wells in a plate sized to admit a single bead per well (Fig 1B). Finally the cyclic sequencing phase begins during which each position is iteratively read (Fig 1C).

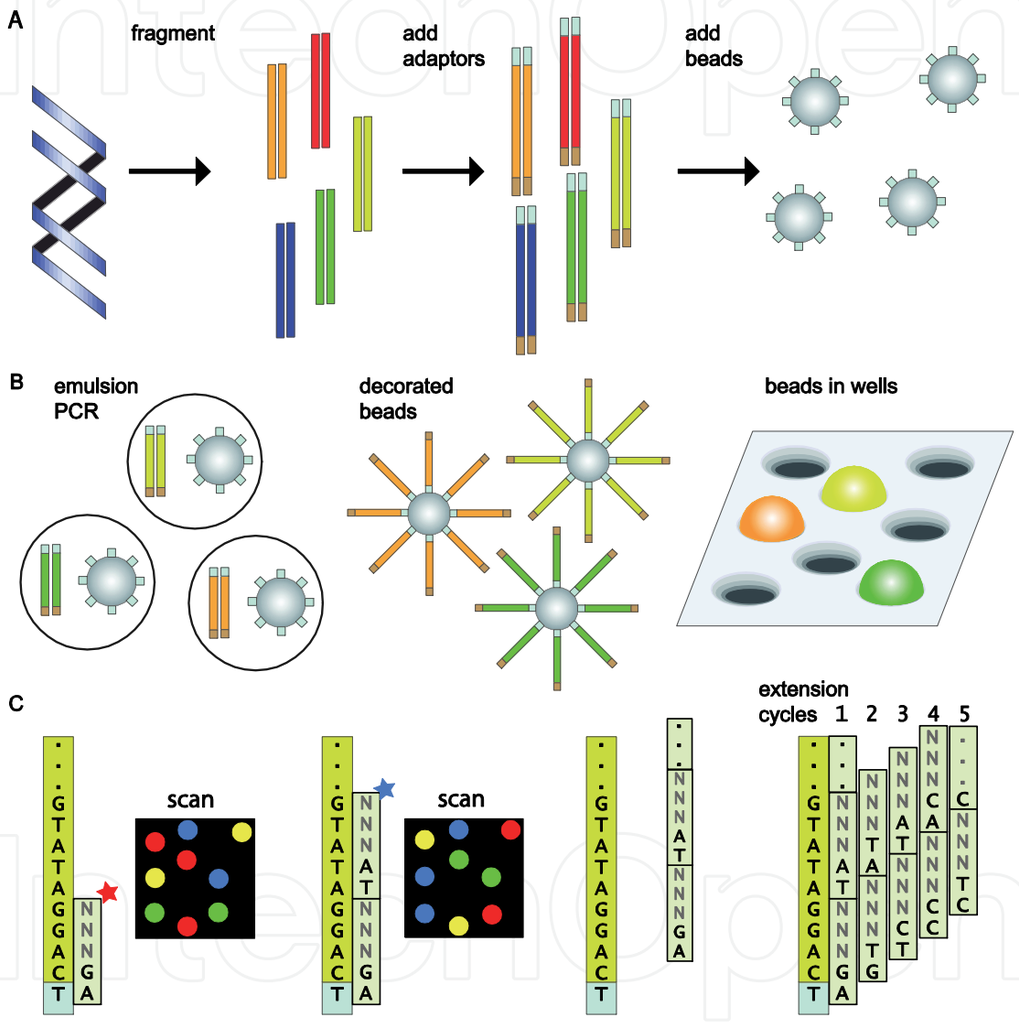


Fig. 1. A simplified outline of the ABI SOLiD sequencing procedure. A) Sample preparation and the addition of glass beads decorated with primers. B) Emulsion PCR amplifies a single template so that its copies are primed by primers bound to a glass bead. C) The sequencing reaction repeats through five extension cycles where the primers are offset by one position in each cycle so that each position in the template is interrogated twice.

The most distinctive feature of SOLiD data is the fact that during the sequencing phase nucleotides are added in dinucleotide probes. In each cycle a nucleotide pentamer is added in which the two 5′ bases are determined by the attached dye (Fig 1C). Once the plate is imaged, the dye is removed leaving the newly added pentamer (Fig 1C). Each cycle therafter

interrogates two more bases offset by three positions from the previous cycle (Fig 1C). As the growing fragment reaches the desired length the entire fragment is washed off and a new primer bound at an offset of one so that a different set of bases are interrogated as this new strand grows (Fig 1C). This process is repeated five times, each one offset by one base from the last so that each position is ultimately interrogated twice (Fig 1C).
Four fluorescent dyes are used but each dye can be carried by one of four nucleotide dimers. As each color is read, the recorded data corresponds to a sequence of colors coded by 0, 1, 2, or 3; this is called color-space. This arrangement means that for any given string of numbers there are four possible nucleotide sequences that it may encode. Given knowledge of the first base it is possible to determine the most likely nucleotide sequence encoded by the entire read. However, to do this prior to assembly of the reads into contiguous sequences (i.e., contigs) for comparison or alignment to a reference genome would result in losing the advantage of SOliD's built-in error checking (afforded by reading each base twice). For example, If a read was determined to possess a position that does not match a consensus reference sequence, it would be ambiguous in other technology platforms whether it were a real difference or sequencing error. With the double-coverage afforded by SOLiD color-space the same "error" is not likely to be made twice in subsequent cycles and it is much more likely that a real variation has been identified instead of a sequencing error.
It should be noted the SOLiD color-space, in which short reads are reported, can be difficult to work with for some assembly applications. Most assembly programs are initially designed to work with nucleotides and require special pre- and post-processing programs to properly assemble color-space reads and these are not always available. Many, but not all, of the specialized alignment programs that can align short reads to a reference library are also able to handle color-space reads but require special options to be enabled.

2.1.2 The illumina genome analyzer platform

The Genome Analyzer (GA) platform typically produces read lengths in the range of 35-150 bp and requires 2 to 14 days for a sequencing run depending on the amount of data desired. Each flow cell contains 8 lanes each of which can produce 80 million reads or more. Daily throughput is estimated at 6.5 Gb for a run in which both ends of fragments (i.e., paired end) are sequenced to 100bp.
The Illumina process also begins by shearing sample DNA (or cDNA) into fragments that are size selected in a target range, often around 200 bp (Fig 2A). These fragments then have short adaptors ligated to both ends of the sample fragments such that unique primer sequences are ligated to either end (Fig 2A). The fragments are then washed onto the flow cell that has sequences complimentary to the two unique primers bound to its surface (Fig 2A). The concentration of fragments on the flow cell is controlled such that they bind sparsely enough on the surface to be optically distinguished from neighboring fragments. Template sequences are only bound by base-paring to primers covalently bound to the flow cell. An initial PCR step produces a complimentary copy of the template now covalently bound to the flow cell, and following this the original template is removed by washing.
The next steps (Fig 2B) are repeated several times to produce a colony of copies of the sequence via 'bridge' PCR. The free end of the template pairs with one of the primers covalently bound to the flow cell and a PCR cycle produces a new copy bound by one end to the flow cell a short distance from the first. After several bridge PCR cycles, a cluster of copies is built up around the originally bound sequence.

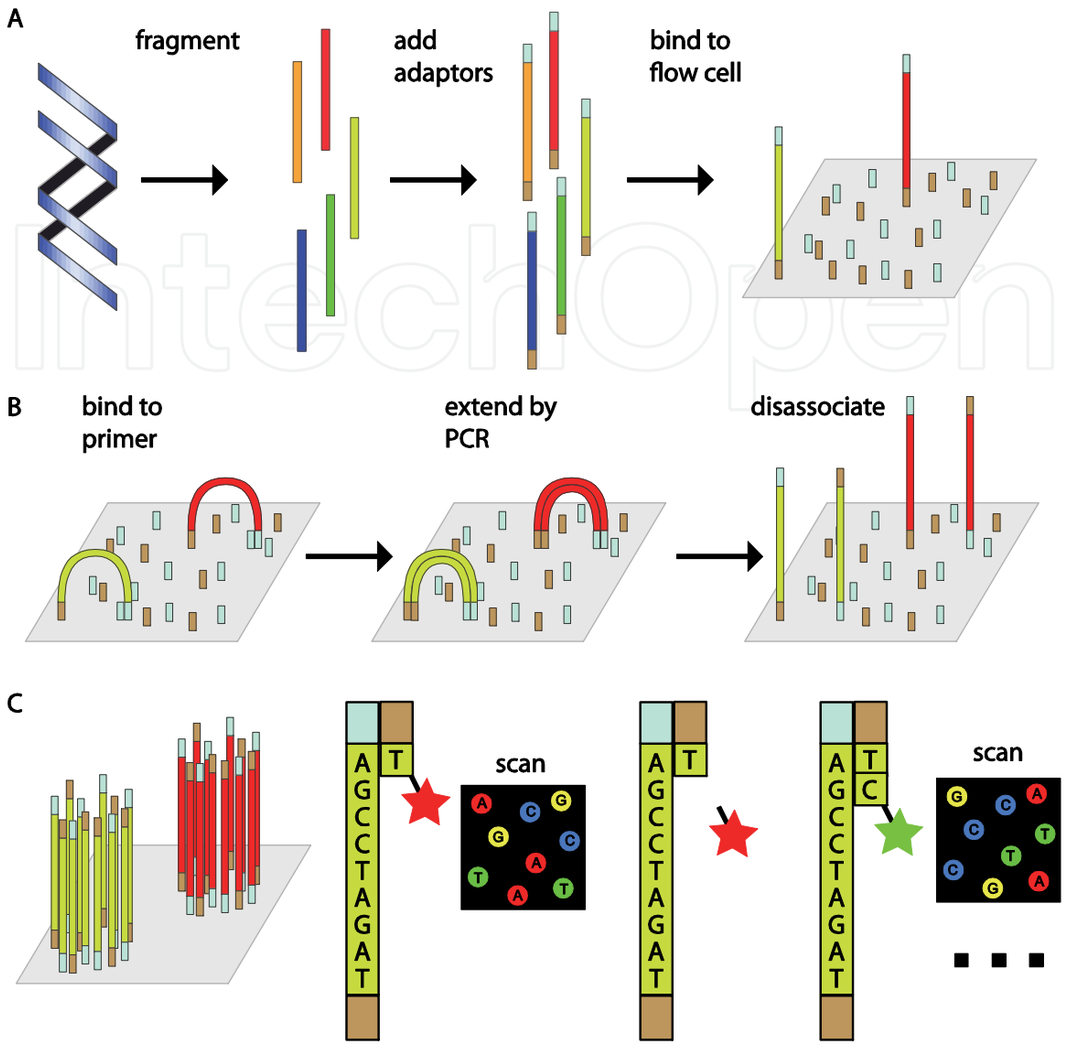


Fig. 2. Simplified outline of the Illumina Genome Analyzer process A) Sample preparation and attachment to the flow cell. B) Bridge PCR amplifies each bound fragment producing a cluster of copies. C) The sequencing reaction extends the growing strand by one nucleotide, excites attached fluorophores which are read optically, and removes the terminator and fluorescent dye before repeating with the next nucleotide.

When spots have reached sufficient density to produce clear signals (Fig 2C) the cyclic sequencing reaction can begin. One of the two unique primers is attached to the free ends and nucleotide addition cycles commence. Each nucleotide contains a different fluorescent reporter tag and a reversible terminator. During each cycle all four bases are flowed onto the reaction chamber, but since each contains a replication terminator only a single one can be incorporated into any elongating sequence (Fig 2C). Laser sources excite the fluorescent reporter of the added nucleotide and an optical sensor detects the wavelength of light emitted. The color of each spot is tracked with each cycle and interpreted directly as a nucleotide base (Fig 2C). This cycle is repeated until the reads reach the desired length and the entire sequencing process is then repeated using the other unique primer to sequence the complementary copies of the DNA.

We have briefly covered two popular NGS sequencing techniques to introduce the capabilities of the technologies and what types of data are produced. There are several other sequencing technologies and many recent reviews covering them are available (Metzker, 2009; Voelkerding et al., 2009; Bräutigam and Gowik, 2010; Nowrousian, 2010). The reader is encouraged to seek out the latest reviews as these technologies are advancing with immense speed and published information quickly becomes outdated.

2.2 Sequence assembly algorithms

When the human genome project first began capillary sequencing base on Sanger technology was the primary sequencing tool employed (Lander et al., 2001). It was extremely labor intensive yet at the time an amazing amount of sequence data was being produced. The Sanger reads produced where about 700 bp in length. Some current NGS techniques are now able to produce reads close to this length, while others hold the promise of producing several hundreds to thousands of base pair length reads.

Package	Availability
phrap	www.phrap.org
wgs-assembler (celera)	sourceforge.net/apps/mediawiki/wgs-assembler/
ARACHNE	ftp.broadinstitute.org/pub/crd/ARACHNE/
Phusion	www.sanger.ac.uk/resources/software/phusion/
RePS	Contact authors at: reps@genomics.org.cn
PCAP	seq.cs.iastate.edu/pcap.html
Atlas	www.hgsc.bcm.tmc.edu/cascade-tech-software_atlas-ti.hgsc

Table 1. Several overlap assembly programs.

The basic strategy for assembling sequences of this length is to use an overlap graph. In an overlap graph nodes represent whole reads and connections represent overlap between the reads. In this case the reads are large and a significant amount of unique information is held in each overlap. Many repetitive features and similar sequence properties that would stymie a short read assembler are easily resolved by long reads and an overlap strategy. Still, assembly problems are not trivial and many packages have continued to mature and acquire a variety of tools. A listing of overlap-based assemblers is given in Table 1.

2.2.1 De Bruijn graph assemblers

As NGS data became available it was quickly apparent that new algorithms were needed to assemble the very short sequences being produced. This problem was addressed by application of discoveries made independently by both De Bruijn and Good in 1946 (de Bruijn, 1946; Good, 1946). All of the most successful short sequence assembly programs in use today utilize the De Bruijn graph as a central data structure and then leverage other aspects of the data to improve upon the assembly process.

The first step in a De Bruijn based assembler is to build the graph. To do so, each short read is broken into k-mers where k is a pre-defined integer length; each k-mer will be a node in

the graph (Fig 3A). The k-mers are defined by recording the sequence in a window of size k and sliding that window down by one position for the length of the short read – producing a new k-mer at each position (Fig 3A). If a short read has a length of L, it will contribute L-k+1 k-mers to the graph. The number of occurrences of each k-mer is also counted and will come into play in a subsequent step.

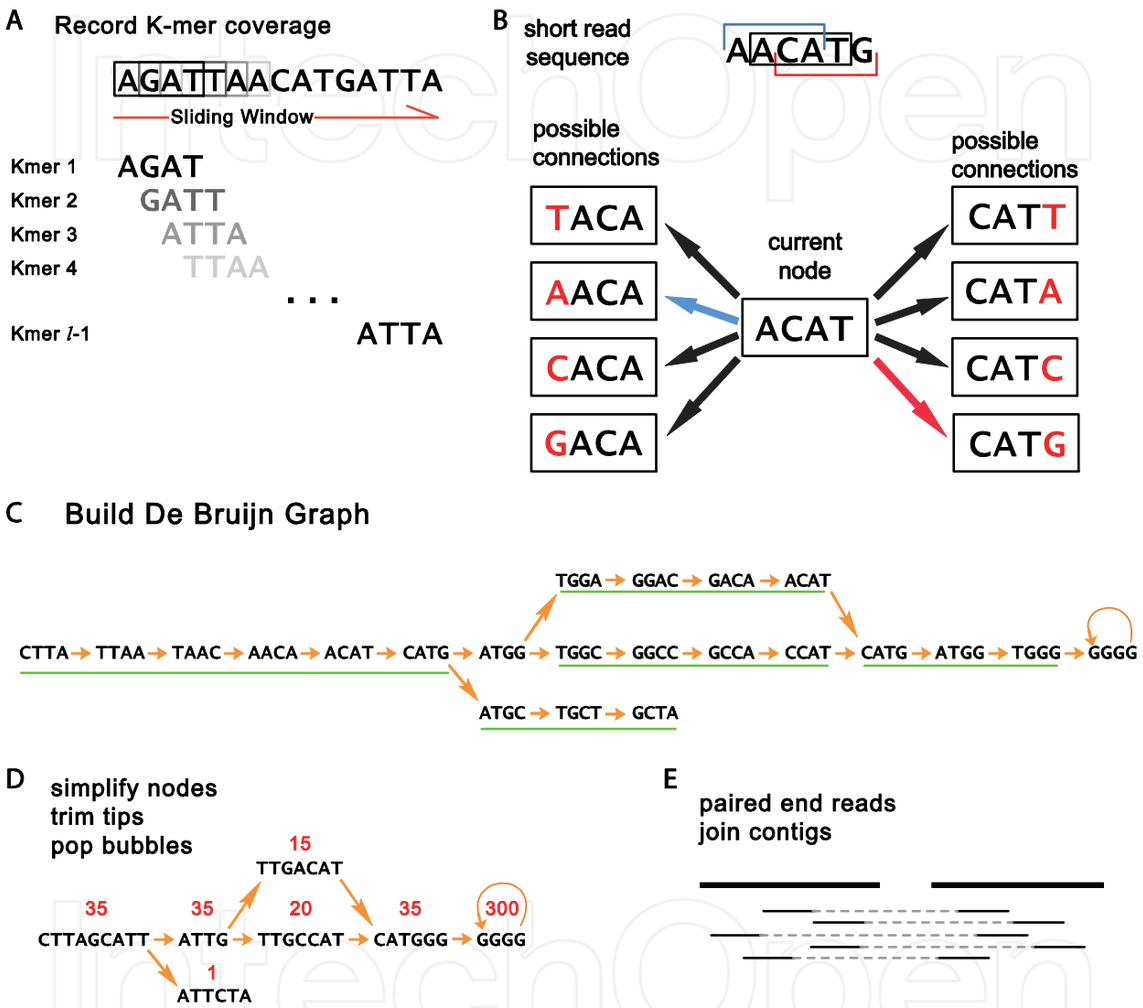


Fig. 3. Outline of a De Bruijn graph based assembler

The edges (or connections between nodes) represent a k-1 overlap between the connected nodes. Thus, we see that each node can have 8 possible connections (Fig 3B). Connections are recorded as they are observed in the raw read data. As reads are passed into the graph building algorithm discrete seed graphs begin to expand and are joined as the reads connecting them are found. In the end several thousand discrete, internally connected graphs exist in the working memory of the computer. An idealized example of one is given in Fig 3C. This is a very simple example but several complicating features are represented here. At this stage a simplified graph can be constructed in which linear stretches (underlined in green in Fig 3C) are condensed into nodes and edges are still k-1 overlaps. The resulting simplified graph is given in Fig 3D, and some of the problems can begin to be

addressed. The leftmost is a 'tip'; a dead end likely caused by a sequencing error near the beginning or end of a short read. A bubble is also present in the center of the graph indicating two alternative possible paths through the k-mer space are present in the short read data. This also could be the result of a sequencing error or a genuine sequence variant. The depth of coverage for each simplified node is indicated by a red number above each node. This information can be used to trim off any tips and remove bubbles with low coverage. Higher coverage anomalies may merit incorporation into alternately assembled contigs depending on the application.

The right-most feature in this graph is a cyclic node. This creates a problem for short read assemblers when repetitive sequence regions are encountered. It could be the sequence has only 4 guanines in a row, or 40, it is impossible to tell from the information generated. This sort of assembly problem is more difficult to resolve by addressing coverage alone and usually results in breaks in contigs. Paired-end information can rescue some of these repetitive situations but scaffold contigs may be broken for many other reasons as well. However, if sufficient paired-end sequences are available that join two contigs it is possible to estimate the size of the gap between them given the expected fragment size (Fig 3E).

One major practical drawback of De Bruijn graph based assemblers is the amount of memory (RAM) required to build, and traverse the graph during an assembly. For example, the Velvet assembler package may require use of 70-100 GB of physical memory to build a vertebrate transcriptome assembly from 100 million reads. Although single machines with such large amounts of memory are not as rare and expensive as they once were, they remain somewhat difficult to find and gain access to. There are several assembler packages that have attempted to address this requirement for large memory. For example, a distributed approach has been implemented in the Abyss assembler and this spreads the workload across several nodes in a computer cluster. Optimizations in the SOAPdenovo package first seek to reduce the amount of memory required by attempting to correct erroneous k-mers produced by sequencing errors. In one study, this approach allowed the number of 25-mers in an assembly of the human genome to be reduced from 14.6 billion to 5.0 billion (Li et al., 2010a).

An alternative to purchasing computer capability with very large memory is use of a cloud computing services, such as the Amazon Elastic Compute Cloud (aws.amazon.com/ec2/). For a fee, computer time is available in dynamically generated computing environments. Several instance types are available including some with up to 68.4 GB of memory and two cluster instance types optimized for traditional compute nodes or GPU nodes. While no assembly process has yet been reported as having used this resource several similarly complex analyses have reported favorable experiences (Afgan et al., 2010; Di Tommaso et al., 2010; Wall et al., 2010).

De Bruijn Assemblers	Availability
EULER-SR	euler-assembler.ucsd.edu/portal/
Velvet	www.ebi.ac.uk/~zerbino/velvet/
ALLPATHS-LG	ftp.broadinstitute.org/pub/crd/ALLPATHS/Release-LG/
Abyss	www.bcgsc.ca/platform/bioinfo/software/abyss
SOAPdenovo	soap.genomics.org.cn/soapdenovo.html

Table 2. Several De Bruijn graph based assemblers

2.3 Overview of sequence assembly process

We have discussed the basic workings of assembly algorithms in order to provide a foundation for further discussion of assembly and the effects that different choices can have on the outcome. We now turn to a larger view of the practical assembly process. At each step we will give recommendations based on our experience and mention other sources for information and help.

2.3.1 Sequence filtration

Prior to NGS read assembly it can be beneficial to remove reads that are more likely to carry erroneous sequences. This is most important for De Bruijn graph based assemblers because each erroneous base call creates up to k erroneous nodes in memory. Thus, large data sets can very quickly exceed even very large memory systems. There are many types of sequencing errors that may need to be removed and some are unique to certain types of techniques. For example, sample DNA can become contaminated by bacterial or vector sequences and so screening reads against appropriate libraries can help to remove some of these contaminants.

Short reads produced by the Illumina GA platform tend to decrease in quality as they are lengthened as well as have an increased error rate in the first few bases. To deal with this some tools (built in options in BWA and Bowtie short read alignment programs) will allow one to trim all reads by a certain length from either end in a set after measuring average quality scores across a read set. Other tools such as the FASTX-Toolkit (hannonlab.cshl.edu/fastx_toolkit/) are more adaptive and deal with each read individually. Another strategy that attempts to correct short reads is to enumerate all the k-mers defined by a set and modify those with very low occurrence frequencies (Schröder et al., 2009; Li et al., 2010b; Shi et al., 2010). Few papers primarily address this issue but the quality of the final assembly can only be as good as data you begin with.

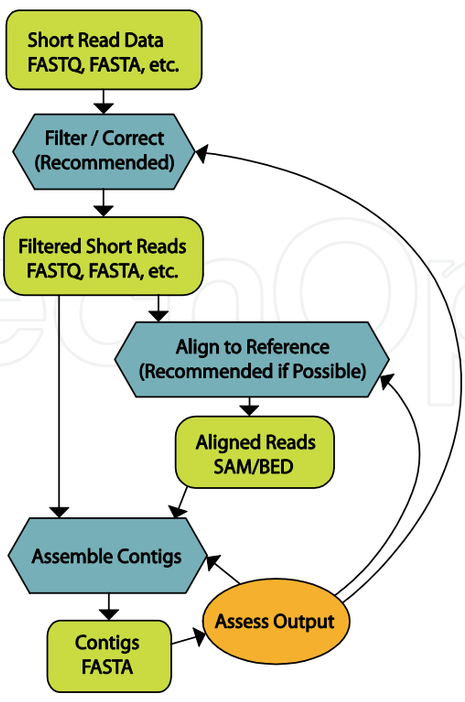


Fig. 4. Outline of an assembly process

2.3.2 Use of a reference library

Reference sequences can be used in many ways to aid in assembly. The most straight forward is to map reads onto a set of closely related reference sequences (using a tool such as BWA, Bowtie, Tophat, etc. section 2.4), then derive a consensus sequence from the reads aligned to each reference sequence. Among others, the samtools pileup or mpileup tools can aid in this approach. This limits the resulting sequences to the set of previously known reference sequences but that is not necessarily a problem. The Cufflinks tool is a unique take on reference based assembly. It is specifically designed to find exons and intron-exon junctions by mapping transcript sequence to a reference genome.

Reference sequences can also be used to guide a de novo assembly. It is possible there are other tools which enable this procedure but here we describe the use of the Columbus extension in the Velvet package. In this case short reads are again aligned with a separate tool to reference sequences which may be genome or transcript seqeunces. The resulting alignment file and reference sequences are then given as input to Velvet which will initially carry out its de novo assembly process as normal. The reference sequences are treated in a sense as long reads and are used to scaffold together appropriate contigs that resulted from the initial assembly process. This technique uses known sequences to extend assembled contigs while also allowing for the discovery of novel sequences.

2.3.3 Sequence assembly

While many NGS assembly packages utilize the De Bruijn graph to represent k-mer connectivity, each has a slightly different algorithm to traverse the graph, prune it, and extract contigs. Most of the freely-available, academically-developed assembly packages have extensive manuals and, more importantly, active communities of users and developers. An extensive listing of the settings and options that can be modified in even one of these packages is far beyond the scope of this discussion. Some considerations however transcend all of these software packages and are discussed here.

The selection of k (k-mer size or hash length) will have a huge impact on assembly. Short k-mers allow for the assembly of low coverage regions since for any two reads to be linked in k-mer space they must overlap by at least k-1. Conversely a too-short k-mer size could allow contigs to be linked in k-mer-space which are not truly linked; thus leading to a chimeric assembly. Very high k-mer sizes significantly cut down on chimeric contigs but impair the assembly of low expression level transcripts and reduce the contiguity overall. A good approach is to scan a range of k-mer sizes and compare the results of several assemblies to determine a k-mer size that gives the best balance.

Another important parameter to consider is how the assembler uses the coverage levels to assemble contigs. In Velvet, for example, the minimum coverage cutoff and expected coverage parameters define a range of coverage levels to consider. This is fine for genomic sequences where coverage levels should be much more consistent, but is extremely problematic for transcript assembly. The Oases extension in Velvet is designed to adapt to varying coverage depth levels and is allowed to report alternative contigs instead of selecting only high coverage paths through the graph.

The diverse range of De Bruijn graph-based assemblers each take different approaches to traversing the graph and pre- and post-processing the data. Software documentation is an excellent place to begin to understand the various assembly parameter modifications and settings allowed. As previously mentioned most of the academically developed packages have an associated community that communicate via e-mail listserv (many of which are archived online) or internet forum.

2.3.4 Assessing assembly quality

This is likely to be the most challenging step in an assembly. A set of basic statistics that are often seen in literature are the N50 value, overall length, number of contigs, and largest contig. The N50 is the length-weighted median length. Another way to think about it is to say that at the N50 length, half of the length in the set of assembled contigs is in contigs equal to or shorter than this value. It is a measure of contiguity since the N50 length increases as sequence length is shifted into longer contigs. The overall length is simply the sum of the lengths of all contigs, and the number of contigs and largest contig are self-explanatory. These are basic statistics often seen in literature but they are fairly limited in assessing assembly quality.

It is generally desirable to quantify correctly assembled contigs, but this is a very tricky thing to do especially with novel transcriptomes. There is no one good approach to assess this easily so we will present several and discuss advantages and disadvantages of each. One approach is to use BLAST or other similarity search tool to compare the assembly to a well-annotated transcriptome of a closely related species if available or a large curated set like the non-redundant (nr) database maintained by the NCBI. A tool like Blast2Go (Conesa et al., 2005; Conesa and Götz, 2008; Götz et al., 2008) can be used to analyze the BLAST results and select a good match for each contig. Trying to maximize unique hits may be a useful indicator but the annotation by BLAST may give different results for alternate splice forms.

Another useful metric is to measure how completely a reference transcriptome from a closely related species is covered by the assembled contigs. This depends heavily on the quality of the reference transcriptome and may not tell very much about the contiguity of the assembled contigs.

A third indication that contigs have been properly assembled is their ability to map to a reference genome. A tool like gmap (Wu and Watanabe, 2005; Wu and Nacu, 2010) can quickly map a large set of contigs to a large genome and report its results in a variety of formats including some basic statistics for each mapping. This would seem like the best method but some software development may be necessary to extract full meaning from such an alignment.

Analyzing the assembly often leads to another round of refinement possibly reaching all the way back to doing more sequencing. More stringent or different filtering, replacing the reference with the assembled contigs, or modifying assembler settings can all help to refine an assembly. Usually this process continues until a 'good enough' transcriptome is reached and that is defined by each researcher for their specific needs.

2.4 RNA short read mapping

After a reference transcriptome or background genome sequence has been efficiently assembled, the next step in many experimental designs is to accurately map RNA-seq reads derived from specific cell or organisms states to it as a method to profile global gene expression (Fig 5). Generally speaking, programs designed for EST mapping [i.e., MUMmer and BLAT(Kent, 2002; Kurtz et al., 2004)] are suitable for reads generated from Roche 454 platfoms, but are not nearly efficient enough for use with short reads generated by Illumina Gene Analyzer or ABI SOLiD NGS platforms. Alignment algorithms designed specifically for NGS short reads are necessary to map reads from latter two platforms. Over the past two years, a wide variety of different programs have been developed to meet the challenge of efficiently mapping millions of short reads and the number of available programs seems to be continuously growing. The challenge for biological scientists is how to choose the best

program that is optimally suited to their specific project. Table 3 shows five currently popular programs available for short read mapping that will be evaluated herein.

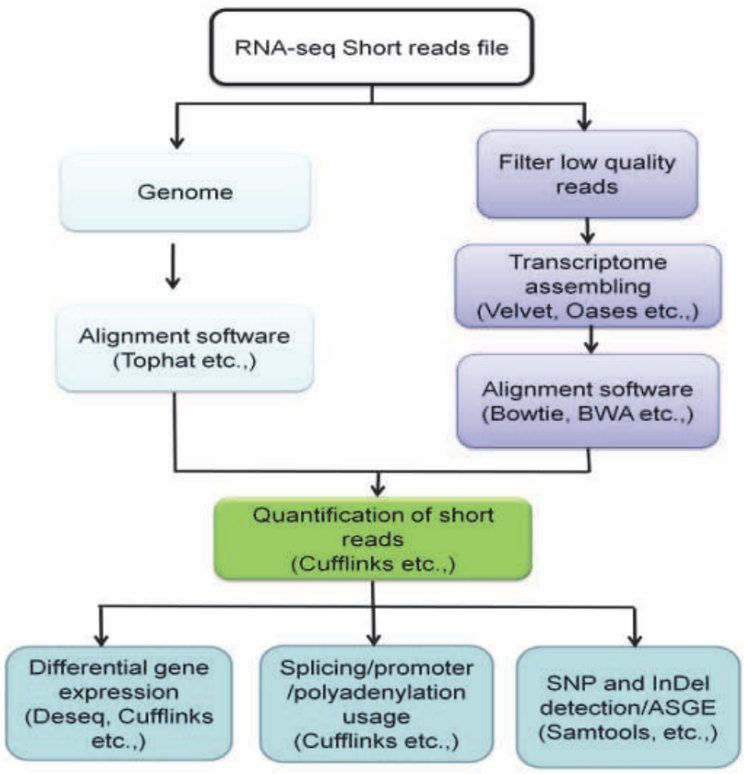


Fig. 5. RNA-Seq project pipeline and commonly used programs. (see; http://en.wikipedia.org/wiki/List_of_sequence_alignment_software#Short-Read_Sequence_Alignment)

To evaluate these programs, we used an *X. maculatus* reference transcriptome [for description of the species see section 2.7 and (Walter and Kazianis, 2001; Kallman and Kazianis, 2006; Meierjohann and Schartl, 2006)] built from over 200 million paired-end reads sequenced from the brain, heart, and liver tissues of mature individuals using the Illumina GAIIx platform (Expression Analysis® Inc. Durham, NC). We used the Velvet assembly package (Zerbino and Birney, 2008) to integrate the combined read set with a hash length (k-mer size) of 43. Oases (http://www.ebi.ac.uk/~zerbino/oases/) was used to perform the final assembly and resulted in a final transcriptome having 110,604 transcripts with an average length of 2,197 bp, and a total size of 243 Mb. In addition, we employed 34 million 60 bp paired-end reads (GAIIx, no custom filtration) sequenced from RNA isolated from *X. maculatus* liver tissue and mapped them back to the reference transcriptome described above to test and compare all five programs in terms of RAM usage, computing time and mapping sensitivity (e.g., the percent of mapped reads).

As shown in the Table 3, the five different programs required different amount of RAM and produced different mapping efficiencies. Bowtie is currently one of the most popular mapping programs and has a reputation for very rapid mapping speeds. It employs a Burrows-Wheeler Transform (BWT) and full-text minute-space index (for review of

alignment algorithms, see (Li and Homer, 2010), which greatly reduces both the memory usage and computational time. In our test, Bowtie proved to be the fastest mapping program and also used a modest amount of RAM. The small RAM usage and speed of Bowtie allows it to run on a standard desktop computer. However, Bowtie's fast performance speed is not without cost. Bowtie only allows non-gapped alignments between reads and references, thus sacrificing some sensitivity for faster mapping speed. Therefore, it was not surprising that Bowtie had the lowest mapping percentage of all tested programs. In addition, using genomic sequences as the reference for mapping RNA-seq reads with Bowtie might not be appropriate since reads spanning two exons cannot be mapped without the support of gap alignment.

Program	Maximum RAM Usage	Time	%of mapped reads	Feature	Reference
Bowtie	2.6G	40 min	42.51	Ultra fast aligner	(Langmead et al., 2009)
BWA	1.2G	64 min	52.05	Support gap alignment	(Li and Durbin, 2009)
Novoalign	1.4G	41 hr[a]	59.81	High sensitivity and allows up to 8 mismatches	www.novocraft.com
SHRiMP	7.0G	14 days	53.08	A collection of mapping tools	(David et al., 2011)
Tophat[b]	63G[b]	5.5hr[b]	52.92[b]	Splice junction reads aligner	(Trapnell et al., 2009)

[a]Only one thread is used for free version. Licensed user can use multi-threads feature of Novoalign.
[b]Transcriptome is used as the reference in this case. Tophat is designed for using genome sequence as reference so the actual time and mapping efficiency may vary when genome is used.

Table 3. Popular short-read alignment software.

An alternative to Bowtie is BWA (Li and Durbin, 2009), which also uses a full-text minute-space index based algorithm but supports gapped alignments. In our test, BWA used least amount of RAM and was comparable to Bowtie in computing time. The gapped alignment feature of BWA makes it more suitable should variations (i.e., small insertion/deletion or InDels) exist between the reference genome or transcriptome and the RNA-seq reads being mapped. This serves to increase the mapping sensitivity of alignments. In our test, BWA reported more reads properly mapped than Bowtie, suggesting BWA is more sensitive in identifying possible alignments between short reads and reference sequences.

Two other programs tested were Novoalign and SHRiMP. They were both noticeably slower than Bowtie or BWA programs. Both Novoalign and SHRiMP programs use a hashing reference based algorithm, which can be traced back to BLAST searching but is optimized for alignment of short reads. For Novoalign, we tested the free version and thus only one thread was used while in all other cases four threads were used during the read mapping trials. Therefore it is likely the licensed version of Novoalign, employing fully multi-thread functions, will exhibit greatly reduced computation time. Novoalign showed the highest mapping percentage in all tested program, indicating the excellent sensitivity of the hashing reference based algorithm. Unlike Novoalign, SHRiMP employs a k-mer hashing index and Smith-Waterman algorithm which gives it robust mapping sensitivity and specificity (David

et al., 2011). However, SHRiMP requires large amounts of RAM and was the slowest program tested. The increased computational time and RAM requirements make SHRiMP less attractive for projects needing high-throughput data analyses.

The final program we tested is Tophat (Trapnell et al., 2009). Tophat is a splice junction mapping program quite different from the previous four programs. Tophat is designed to align RNA-seq reads to a reference genome. Using Tophat, RNA-seq reads can be analyzed to identify novel splice variants of genes. Tophat first employs iterative rounds of Bowtie mapping to identify genomic regions with RNA-seq read mapping, and then to generate potential splice donor/acceptor sites flanking the sequence. Unmatched reads are then mapped to these splice junction sequences again by Bowtie to confirm possible splice junctions. Tophat prefers a genome sequence as a reference and mapping results may not be reliable if only a transcriptome reference is used. Of all five programs tested, Tophat required the most RAM for alignment processing. Thus, Tophat may be best used in a high-performance computing environment.

Overall, the choice of which alignment program to use should be based on both the available computer resources and experimental design. If the alignment process is to be performed on a standard desktop computer (e.g., about 4G RAM), SHRiMP and Tophat should be avoided due to memory constraints. However, Bowtie, BWA, and Novoalign can map reads efficiently on standard office computers. On the other hand, if a genome sequence is available for a reference, or the purpose of study is to identify InDel's between a reference and reads, Bowtie may not be the best choice since it lacks gap alignment capabilities. Tophat is preferred when a genome sequence is present because it fully considers potential donor/acceptor sites in the genome and allows the alignment to cross splice junctions accurately, compared to the other programs. However, should a transcriptome be used as mapping reference, Tophat should be avoided as it is designed for use with genome sequence data. Finally, although all programs tested herein fully support both Illumina and SOLiD single or paired ends reads, SHRiMP and BWA (through its BWA-SW module) also support mapping of mixed RNA-seq short reads with longer Sanger or 454 Roche based reads. In such situations, where mixed reads are to be used, employing a single program saves both time and effort in the subsequent analyses.

Overall, with the continuous increase in throughput for recently developed sequencing technologies, new algorithms are becoming available almost monthly; while older programs are continually refined to reduce computational time and memory demands. However, there is not a perfect program suited for all experimental designs and hardware requirements. The choice of programs will need to be reviewed and evaluated on a case-by-case basis.

2.5 Quantification of gene expression level

Currently most short read alignment programs adopt SAM (or its binary version, BAM) as the alignment output format. SAM (Sequence Alignment/Map format) is a tab-delimited text format designed for recording short read alignment information. Although it is human readable, a typical SAM file will consist of millions of lines of mapping information that is required for downstream analyses. In the next steps of data processing, most RNA-Seq projects aim to utilize read mapping as a means to quantify gene expression levels across entire reference transcriptomes or genomes (Fig 5).

An early approach of using RNA-seq to quantify gene expression relied on simply counting the total number of reads mapping to each transcript in sample. However, since the total number of reads varied between each sample, read counts could not be use for direct

comparison or determination of differential expression between samples. In addition to total read count numbers between samples, the length of transcripts within each sample may vary and longer transcripts are generally more likely to have more reads mapped to them than shorter ones. Thus, performing tasks such as finding the highest expressed genes in a sample via direct read counting proved to be inaccurate. In an effort to normalize the sample size and transcript lengths for head-to-head read count comparisons, Mortazavi and coworkers (2008) developed the term Reads Per Kilobase per Million of mapped reads (RPKM) as a standard to compare different genes within or across different samples (Mortazavi et al., 2008). RPKM and its derived term FPKM (Fragments Per Kilobase per Million of mapped reads) for paired end reads, have been widely adopted in RNAseq studies employing various experimental systems.

Since RPKM is easy to calculate and understand, it provides a platform to facilitate comparison of transcript levels both within and between samples. However, since the purpose of most studies involving RPKM is to compare differential gene expression, one must be aware that RPKM values may be affected by both experimental and computational issues. Experimental issues such as the quality of RNA, contamination of ribosomal RNA and length of output reads (Pepke et al., 2009; Costa et al., 2010) and computational influences including accuracy of gene modeling, and inclusion/exclusion of multiple mapped reads, may all affect the results obtained. One issue deserving special attention is the diminished statistical power one accepts when using RPKM to detect differential expression of longer transcripts (Oshlack and Wakefield, 2009). Employing RPKM, where the number of reads from a given transcript is divided by the length of the transcript, serves to deflate statistical power by producing a large sample size (more reads). To illustrate this, assume a 1000 bp gene (gene A) has 5 and 10 mapped reads in sample 1 and sample 2, respectively. In the same samples, a 10,000 bp gene (gene B) has 50 and 100 mapped reads, respectively. By definition of RPKM, since gene B is 10 times longer and has 10 times more reads mapped, both genes have identical RPKM values and fold changes in the two samples. Thus one would assume the confidence of gene A and gene B being differentially expressed is exactly same. However, since gene A has a much smaller sample size (15 reads in total) compared with gene B (150 reads), gene A is more prone to statistical error when trying to identify a 2 fold-change in expression between samples 1 and 2. Therefore, although RPKM is widely used to provide a scalable value to quantify gene expression levels, it is affected by variation in a transcript length dependent manner and should not be used to directly compare gene expression.

2.6 Comparison of differential expression

One common goal of many large-scale transcriptome studies is to identify differentially expressed genes between two or more samples. While microarrays have been widely used for over a decade to assess transcriptome-wide gene expression levels, RNA-seq technologies have displayed several advantages over microarrays, such as the ability to identify novel transcripts and to assess quantitative allele-specific gene expression. However, it is still debatable which tool is better to accurately assess gene expression values. In a recent study (Bloom et al., 2009), microarray and RNA-seq results were compared using quantitative RT-PCR (qRT-PCR) assays and it was determined that both methods performed similarly in measuring differential gene expression. The microarray had an advantage over RNAseq in better measure of low-abundance transcripts (Bloom et al., 2009); however, when results of microarray and RNA-seq were further assessed with 2D LC-MS/MS the

expression values estimated by RNA-Seq appeared to be better correlated with the proteomics data (Fu et al., 2009). Overall, these studies prove that RNA-Seq may serve as a reliable method to accurately estimate absolute transcript levels.

Since both microarray and RNA-seq are used to quantify expression levels of transcripts, statistical methods developed for microarrays have been adopted to compare gene expression using RNA-seq. However, there are notable differences between the two technologies and methods successfully used for microarray analysis might not be appropriate for RNA-seq data (Costa et al., 2010). First, the gold standard for any microarray studies is to have at least three replicates in each condition while many RNA-seq projects lack the luxury of replicates due to the relatively expensive cost of sequencing RNA-seq libraries. Methods that have been used in microarray analysis range from simple t-testing to more complicated statistical modeling; but all these techniques rely on having multiple replicates to identify differentially expressed genes. The absence of multiple replicates greatly reduces the statistical power of RNAseq methods. Secondly, for microarray analysis, fluorescence intensity is utilized as the measurement of transcript levels and these data may be treated as continuous data. However, RNA-seq studies utilizing read counts (or RPKM) to gauge the expression of a particular transcript generate discrete data. Thus, statistical models developed for continuous data might not be effective when applied to data generated from an RNA-seq experiment.

Many studies have utilized different statistical tools to identify differentially expressed transcripts in RNA-seq experiments. Simple approaches such as classical Z-test and Fishers exact test have been employed for this purpose (Bloom et al., 2009; Hashimoto et al., 2009). Although these methods are appropriate for hypothesis testing of discrete data, they do not consider the global variations of all genes, thus less robust than more advanced approaches discussed below. There are several studies reported where more sophisticated microarray based methods have been modified and made suitable for RNA-seq projects. One of the pioneering reports involved RNAs extracted from liver and kidney of the same individual that were separated into seven aliquots for each sample and sequenced in individual lanes of a Illumina genome analyzer (Marioni et al., 2008). The variations of these technological replicates were then calculated and were found to fit the variance predicted by a Poisson model. Using the Poisson model allowed the authors to identify 30% more differentially expressed genes than a standard statistic analysis and employing microarrays with the same samples (Marioni et al., 2008). Based on the notion that a Poisson distribution can predict the variations in RNA-seq data, DEGseq, a Bioconductor software package, has been developed for examining differential expression of RNA-seq read count data (Wang et al., 2010). DEGseq modeled the number of reads derived from a gene into a Poisson distribution and used the Fisher's exact test and likelihood ratio test to identify differentially expressed genes (Wang et al., 2010). However, it has been argued the Poisson distribution will underestimate actual variations in replicated samples and tends to predict smaller variations than are actually present in the data (Nagalakshmi et al., 2008). As a result, methods based on the Poisson distribution do not control false discoveries very well. In addition to Poisson distributions, two other Bioconductor packages, DESeq and EdgeR, both take read counts as input and use negative binomial distributions to estimate variations of RNA-seq data (Anders and Huber, 2010; Robinson et al., 2010). EdgeR employs negative binomial distributions to account for variability and assesses differential expression based on Empirical Bayes methods (Robinson et al., 2010). The DESeq package models distributions of read count

data by negative binomial distribution, with variance and mean linked by local regression (Anders and Huber, 2010). Compared with previous Poisson based program, both DESeq and EdgeR control the probability of false discoveries and produce good fits when the number of replicates is small (Anders and Huber, 2010).

In addition to the Bioconductor packages discussed above, another standalone tool termed "Cufflinks", written by same research group that developed Bowtie and Tophat, may be used to read SAM files produced from Tophat directly and compare differential expression in pair-wise manner (Trapnell et al., 2010). The program extracts read count information from SAM files and computes the entropy of the average distribution minus the average of the individual entropies [Jensen-Shannon divergence; see (Menendez et al., 1997)] and the difference between abundances of transcripts in two conditions may be calculated as the square root of this divergence. Cufflinks can be easily integrated with Bowtie/Tophat workflow and outputs FPKM values for two samples and the significance level of the statistics tests. In addition to transcript expression, Cufflinks may also be used to find significant changes in transcript splicing and promoter usage between two samples.

2.7 SNP identification and allele specific gene expression

One major advantage of RNA-seq technology over microarray based approaches is that one may quantify not only total gene expression, but also allele specific gene expression (ASGE) at same time. To study allele specific gene expression using microarrays, one must have very detailed characterization of genome polymorphisms and then specifically design probes to assess the abundance of each allele independently on the array. Therefore, it is difficult to study ASGE in less well-characterized species or genetic models that possess little information of known polymorphisms. With rapid progress in next generation sequencing technologies (NGS), RNA-Seq has been shown to provide single-base resolution and quantitative information for thousands of genes simultaneously (Pastinen, 2010). Notably, this approach does not rely on previous knowledge of known variations and can be used for both identifying polymorphisms and quantifying ASGE. Using both 454 and Illumina sequencing platforms respectively, allelic expression imbalances have been assessed in *Drosophila* hybrids, *Xiphophorus* fishes, and in humans (Serre et al., 2008; Daelemans et al., 2010; Fontanillas et al., 2010; Shen et al., 2011).

Here we demonstrate our recent ASGE study using *Xiphophorus* interspecies hybrid fishes. The genus *Xiphophorus* has at least 27 species of live-bearing fishes found from northern Mexico south into Belize and Guatemala (Kallman and Kazianis, 2006). The *Xiphophorus* genus couples extreme genetic variability among *Xiphophorus* species with the capability of producing fertile interspecies hybrids that have allowed chromosomal inheritance of complex traits to be followed into individual F_1 and backcross hybrid progeny (Kazianis et al., 2001; Walter and Kazianis, 2001; Meierjohann and Schartl, 2006). Using interspecies hybrids provides a unique opportunity to reveal underlying mechanisms of genetic variation.

We have assembled the transcriptome of *X. maculatus* Jp163 A, a highly inbred line species of *Xiphophorus* (Fig 6) using RNA-seq sequencing from brain, heart, and liver tissues (see section 2.5). We first investigate transcriptome-wide SNP polymorphisms between two highly inbred *Xiphophorus* species: *X. maculatus* Jp 163 B and *X. couchianus*. To do this RNA-seq reads sequenced from *X. maculatus* Jp163 B were mapped to the reference transcriptome of *X. maculatus* Jp163 A by Bowtie (Langmead et al., 2009) and SNPs were called by

Samtools (Li et al., 2009). The density of intraspecific SNPs was about 1 SNP/49 kb of transcriptome [Figure 7; (Shen et al., 2011)].

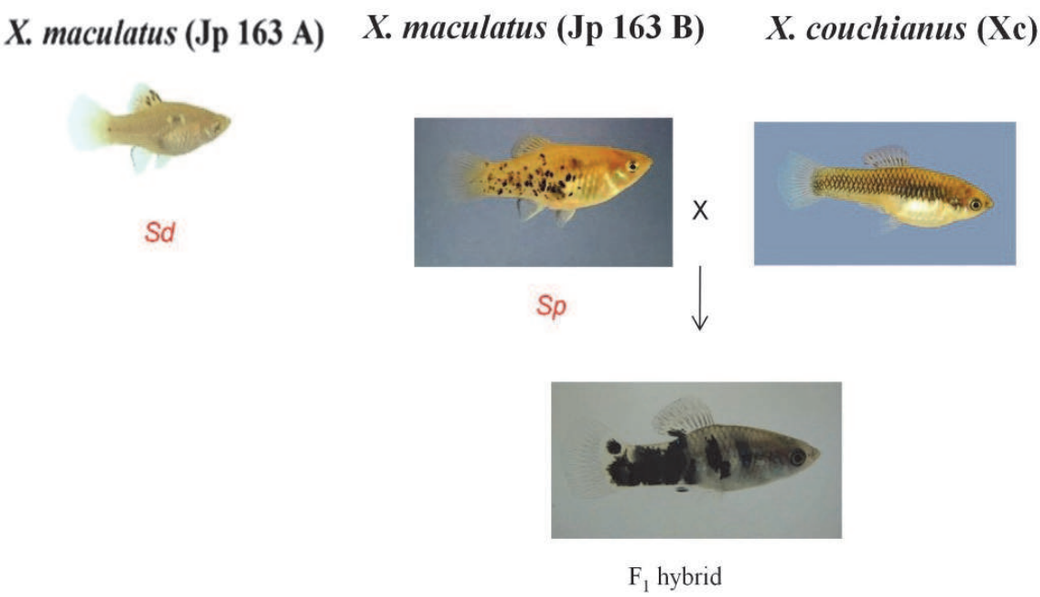


Fig. 6. Fishes used in this study. *X. maculatus* Jp 163 A carrying the Sd pigment pattern is the species utilized for deep transcriptome development and eventual assembly of the reference transcriptome. F1 interspecies hybrids utilized in these studies were produced by crossing the *X. maculatus* Jp 163 B (Sp pigment pattern) and *X. couchianus* parental species. RNA-seq reads analyzed in this study were sequenced from *X. maculatus* Jp 163 B, *X. couchianus* and their F_1 interspecies hybrids respectively.

We wished to ascertain ASGE between *X. maculatus* Jp 163 B, *X. couchianus* and an F_1 hybrid produced from crossing these two species (Fig 6). Thus, we first determined that the 90,788 SNPs, identified between the *X. maculatus* reference transcriptome and *X. couchianus* were also polymorphic between the *X. maculatus* Jp 163 B strain and *X. couchianus*. To improve the accuracy of ASGE analysis in the hybrid, we scored only genes that exhibited greater than 20 SNP supporting reads. These constraints resulted in 38,746 SNPs between *X. maculatus* Jp 163 B and *X. couchianus* that could be clearly assigned to one or the other parental alleles and were unambiguously mapped to 6,524 *Xiphophorus* transcripts in the reference transcriptome.

After identification of SNPs, ASGE can be calculated as number of reads mapped to each allele in the F_1 hybrid (for a diagrammatic illustration of the process, see Fig 7). Since most short alignment programs only allow a limited number of base mismatches (i.e., 2 in case of Bowtie) between reads and reference sequences, the reads representing the *X. couchianus* alleles possessed natural disadvantages in mapping efficiency since they carried an extra mismatch (i.e., the SNP) compared with *X. maculatus* reads. In the F_1 hybrid, we found many transcripts showed more *X. maculatus* mapped reads than *X. couchianus* ones when the mapping was back to the *X. maculatus* reference transcriptome. To eliminate this read mapping bias and create an environment where reads from both *X.*

maculatus and *X. couchianus* alleles had equal chances of mapping to the transcriptome, we first duplicated the *X. maculatus* reference and then introduced all *X. couchianus* specific SNP's into it to produce an *in silico X. couchianus* reference transcriptome (based on *X. maculatus* transcriptome with masked SNPs). The induction of *X. couchianus* reference transcriptome allowed reads with *X. couchianus* alleles to have comparable likelihood of being mapped in ASGE study.

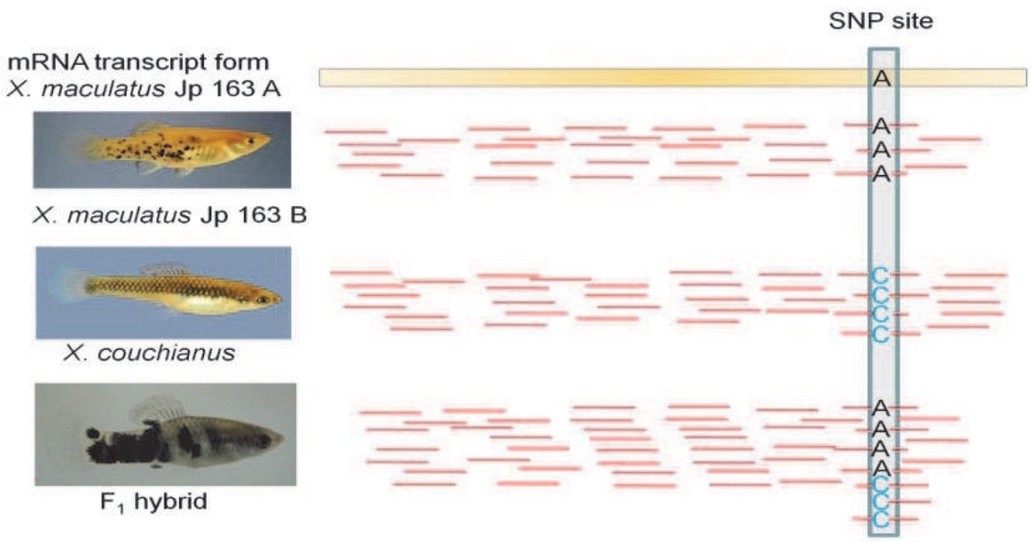


Fig. 7. A diagrammatic example of identification of SNPs and measurement of ASGE in F_1 interspecies hybrids. Red bars represent RNA-Seq reads mapped to the reference transcriptome. Most reads from *X. maculatus* Jp 163 B match perfectly to the Jp 163 A reference transcriptome. RNA-seq reads from *X. couchianus* were also mapped to *X. maculatus* Jp 163 A reference transcriptome and SNPs sites were identified by comparing consensus bases of RNA-seq reads (C in this case) to the corresponding base in the reference transcriptome (A in this case). In the hybrid, reads mapped to SNPs sites are classified by the bases they carry and counted separately as the measurement of ASGE. In this SNP, 4 *X. maculatus* allele reads and 3 *X. couchianus* allele reads were counted in the hybrid.

As shown in Fig 8, using the corrected reference transcriptome allowed both *X. maculatus* and *X. couchianus* alleles to exhibit a more balanced expression pattern (Fig 8b) in the hybrid genetic background than without normalization (Fig 8a). Without proper normalization, we found over 84% of genes in the transcriptome were biased toward over-representation of the *X. maculatus* allele (fraction > 0.5, Fig 8a). After production of the *in silico* reference transcriptome and tolerating 5 mismatches, analyses of the distribution of ASGE in F_1 hybrids indicate that most genes (5,980 of 6,524 genes or 92%) exhibit relatively balanced allele expression in the hybrid genetic background (<70% of preference of one particular allele, those between 0.3 and 0.7 in Fig 8b). Overall, employment of high throughput sequencing technology and proper normalization approaches allow direct and accurate assessment of ASGE in the interspecies hybrids.

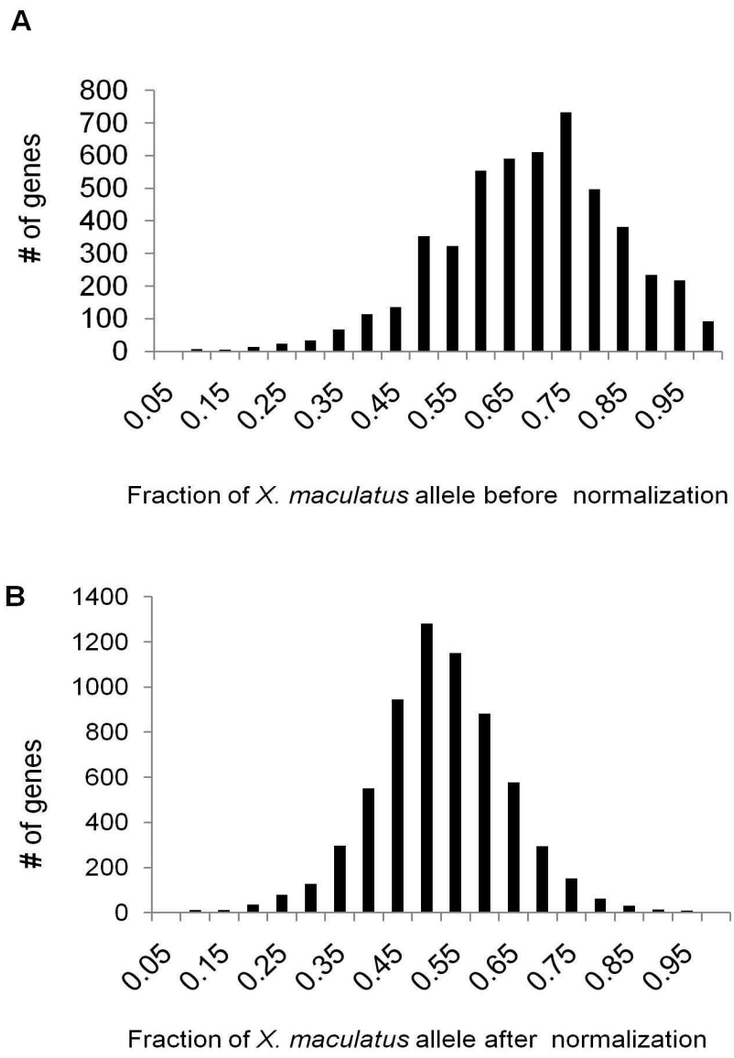


Fig. 8. Allele distribution in F1 hybrid background. A: A histogram shows the distribution of F1 transcripts carrying different parental alleles before normalization. X axis is the fraction of reads carrying *X. maculatus* allele. 0.5 means in that gene, half of F1 hybrid reads can be identified from *X. couchianus* and another half are from *X. maculatus*. 1.0 and 0.0 means reads exclusively carrying *X. maculatus* and *X. couchianus* alleles, respectively. B: Fraction of *X. maculatus in* hybrid background after normalization. We masked *X. maculatus* reference with consensus bases from *X. couchianus* and allowing five mapping mismatches.

3. Conclusion

The bottleneck of large-scale NGS projects has shifted from obtaining experimental data to downstream bioinformatic analyses. With the continuous development of software infrastructure to suit the needs of RNA-Seq analyses, there are several competent programs in each of the analysis step; such as transcriptome assembly, read mapping, and identification of differential gene expression. The real challenge facing many biologists is to find the right tool to use and carefully weighing the strength and weakness of each tool. The

constant advance in sequencing technology will continue to increase the amount of data produced, urging the use of the most efficient tool within the capacity of available computer resources. The combination of the carefully designed experiment and right methodology utilizing NGS data will open a new era for studying species with little historical background genetic information available.

4. References

Afgan, E., Baker, D., Coraor, N., Chapman, B., Nekrutenko, A., and Taylor, J. (2010). Galaxy CloudMan: delivering cloud compute clusters. *BMC Bioinformatics,* Vol.11, (2010), pp. S4, ISSN

Anders, S., and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biol,* Vol.11, No.10, (October 2010), pp. R106, ISSN 1465-6914

Bloom, J.S., Khan, Z., Kruglyak, L., Singh, M., and Caudy, A.A. (2009). Measuring differential gene expression by short read sequencing: quantitative comparison to 2-channel gene expression microarrays. *BMC Genomics,* Vol.10, (May 2009), pp. 221, ISSN 1471-2164

Bräutigam, A., and Gowik, U. (2010). What can next generation sequencing do for you? Next generation sequencing as a valuable tool in plant research. *Plant biology (Stuttgart, Germany),* Vol.12, (2010), pp. 831-841, ISSN

Conesa, A., and Götz, S. (2008). Blast2GO: A Comprehensive Suite for Functional Analysis in Plant Genomics. *International journal of plant genomics,* Vol.2008, (2008), pp. 619832, ISSN

Conesa, A., Götz, S., García-Gómez, J.M., Terol, J., Talón, M., and Robles, M. (2005). Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics (Oxford, England),* Vol.21, (2005), pp. 3674-3676, ISSN

Costa, V., Angelini, C., De Feis, I., and Ciccodicola, A. (2010). Uncovering the complexity of transcriptomes with RNA-Seq. *J Biomed Biotechnol,* Vol.2010, (June 2010), pp. 853916, ISSN 1110-7251

Daelemans, C., Ritchie, M.E., Smits, G., Abu-Amero, S., Sudbery, I.M., Forrest, M.S., Campino, S., Clark, T.G., Stanier, P., Kwiatkowski, D., *et al.* (2010). High-throughput analysis of candidate imprinted genes and allele-specific gene expression in the human term placenta. *BMC Genet,* Vol.11, (June 2010), pp. 25, ISSN 1471-2156

David, M., Dzamba, M., Lister, D., Ilie, L., and Brudno, M. (2011). SHRiMP2: Sensitive yet Practical Short Read Mapping. *Bioinformatics,* Vol.27, No.7, (January 2011), pp. 1011-1012, ISSN 1367-4811

de Bruijn, N.G. (1946). A Combinatorial Problem. *Koninklijke nederlandse Akademie v Wetenschappen,* Vol.49, (1946), pp. 758-764, ISSN

Di Tommaso, P., Orobitg, M., Guirado, F., Cores, F., Espinosa, T., and Notredame, C. (2010). Cloud-Coffee: implementation of a parallel consistency-based multiple alignment algorithm in the T-Coffee package and its benchMarchking on the Amazon Elastic-Cloud. *Bioinformatics (Oxford, England),* Vol.26, (2010), pp. 1903-1904, ISSN

Fontanillas, P., Landry, C.R., Wittkopp, P.J., Russ, C., Gruber, J.D., Nusbaum, C., and Hartl, D.L. (2010). Key considerations for measuring allelic expression on a genomic scale

using high-throughput sequencing. *Mol Ecol,* Vol.19 Suppl 1, (March 2010), pp. 212-227, ISSN 1365-294X

Fu, X., Fu, N., Guo, S., Yan, Z., Xu, Y., Hu, H., Menzel, C., Chen, W., Li, Y., Zeng, R., *et al.* (2009). Estimating accuracy of RNA-Seq and microarrays with proteomics. *BMC Genomics,* Vol.10, (April 2009), pp. 161, ISSN 1471-2164

Good, I.J. (1946). Normal recurring Decemberimals. *Journal of the London Mathematical Society,* Vol.21, (1946), pp. 167-169, ISSN

Götz, S., García-Gómez, J.M., Terol, J., Williams, T.D., Nagaraj, S.H., Nueda, M.J., Robles, M., Talón, M., Dopazo, J., and Conesa, A. (2008). High-throughput functional annotation and data mining with the Blast2GO suite. *Nucleic acids research,* Vol.36, (2008), pp. 3420-3435, ISSN

Hashimoto, T., de Hoon, M.J., Grimmond, S.M., Daub, C.O., Hayashizaki, Y., and Faulkner, G.J. (2009). Probabilistic resolution of multi-mapping reads in massively parallel sequencing data using MuMRescueLite. *Bioinformatics,* Vol.25, No.19, (October 2009), pp. 2613-2614, ISSN 1367-4811

Kallman, K.D., and Kazianis, S. (2006). The genus Xiphophorus in Mexico and central america. *Zebrafish,* Vol.3, No.3, (April 2006), pp. 271-285, ISSN 1557-8542

Kazianis, S., Gimenez-Conti, I., Trono, D., Pedroza, A., Chovanec, L.B., Morizot, D.C., Nairn, R.S., and Walter, R.B. (2001). Genetic analysis of neoplasia induced by N-nitroso-N-methylurea in Xiphophorus hybrid fish. *March Biotechnol (NY),* Vol.3, No.Supplement 1, (June 2001), pp. S37-43, ISSN 1436-2228

Kent, W.J. (2002). BLAT--the BLAST-like alignment tool. *Genome Res,* Vol.12, No.4, (April 2002), pp. 656-664, ISSN 1088-9051

Kurtz, S., Phillippy, A., Delcher, A.L., Smoot, M., Shumway, M., Antonescu, C., and Salzberg, S.L. (2004). Versatile and open software for comparing large genomes. *Genome Biol,* Vol.5, No.2, (February 2004), pp. R12, ISSN 1465-6914

Lander, E.S., Linton, L.M., Birren, B., Nusbaum, C., Zody, M.C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., *et al.* (2001). Initial sequencing and analysis of the human genome. *Nature,* Vol.409, (2001), pp. 860-921, ISSN

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S.L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol,* Vol.10, No.3, (March 2009), pp. R25, ISSN 1465-6914

Li, H., and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics,* Vol.25, No.14, (July 2009), pp. 1754-1760, ISSN 1367-4811

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marchth, G., Abecasis, G., and Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics,* Vol.25, No.16, (August 2009), pp. 2078-2079, ISSN 1367-4811

Li, H., and Homer, N. (2010). A survey of sequence alignment algorithms for next-generation sequencing. *Brief Bioinform,* Vol.11, No.5, (September 2010), pp. 473-483, ISSN 1477-4054

Li, R., Fan, W., Tian, G., Zhu, H., He, L., Cai, J., Huang, Q., Cai, Q., Li, B., Bai, Y., *et al.* (2010a). The sequence and de Novembero assembly of the giant panda genome. *Nature,* Vol.463, (2010a), pp. 311-317, ISSN

Li, R., Zhu, H., Ruan, J., Qian, W., Fang, X., Shi, Z., Li, Y., Li, S., Shan, G., Kristiansen, K., *et al.* (2010b). De Novembero assembly of human genomes with massively parallel short read sequencing. *Genome research,* Vol.20, (2010b), pp. 265-272, ISSN

Marchioni, J.C., Mason, C.E., Mane, S.M., Stephens, M., and Gilad, Y. (2008). RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res,* Vol.18, No.9, (September 2008), pp. 1509-1517, ISSN 1088-9051

Meierjohann, S., and Schartl, M. (2006). From Mendelian to molecular genetics: the Xiphophorus melanoma model. *Trends Genet,* Vol.22, No.12, (December 2006), pp. 654-661, ISSN 0168-9525

Menendez, M.L., Pardo, J.A., Pardo, L., and Pardo, M.C. (1997). The Jensen-Shannon divergence. *J Franklin I,* Vol.334B, No.2, (March 1997), pp. 307-318, ISSN 0016-0032

Metzker, M.L. (2009). Sequencing technologies — the next generation. *Nature Reviews Genetics,* Vol.11, (2009), pp. 31-46, ISSN

Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Methods,* Vol.5, No.7, (July 2008), pp. 621-628, ISSN 1548-7105

Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., and Snyder, M. (2008). The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science,* Vol.320, No.5881, (June 6 2008), pp. 1344-1349, ISSN 1095-9203

Nowrousian, M. (2010). Next-generation sequencing techniques for eukaryotic microorganisms: sequencing-based solutions to biological problems. *Eukaryotic cell,* Vol.9, (2010), pp. 1300-1310, ISSN

Oshlack, A., and Wakefield, M.J. (2009). Transcript length bias in RNA-seq data confounds systems biology. *Biol Direct,* Vol.4, (April 2009), pp. 14, ISSN 1745-6150

Pastinen, T. (2010). Genome-wide allele-specific analysis: insights into regulatory variation. *Nat Rev Genet,* Vol.11, No.8, (June 2010), pp. 533-538, ISSN 1471-0064

Pepke, S., Wold, B., and Mortazavi, A. (2009). Computation for ChIP-seq and RNA-seq studies. *Nat Methods,* Vol.6, No.11 Suppl, (November 2009), pp. S22-32, ISSN 1548-7105

Robinson, M.D., McCarthy, D.J., and Smyth, G.K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics,* Vol.26, No.1, (January 2010), pp. 139-140, ISSN 1367-4811

Schröder, J., Schröder, H., Puglisi, S.J., Sinha, R., and Schmidt, B. (2009). SHREC: a short-read error correction method. *Bioinformatics (Oxford, England),* Vol.25, (2009), pp. 2157-2163, ISSN

Serre, D., Gurd, S., Ge, B., Sladek, R., Sinnett, D., Harmsen, E., Bibikova, M., Chudin, E., Barker, D.L., Dickinson, T., *et al.* (2008). Differential allelic expression in the human genome: a robust approach to identify genetic and epigenetic cis-acting mechanisms regulating gene expression. *PLoS Genet,* Vol.4, No.2, (February 2008), pp. e1000006, ISSN 1553-7404

Shen, Y., Catchen, J., Garcia, T., Amores, A., Beldorth, I., Wagner, J.R., Zhang, Z., Postlethwait, J., Warren, W., Schartl, M., *et al.* (2011). Identification of transcriptome wide SNPs between Xiphophorus lines and species for assessment of allele specific gene expression within F1 interspecies hybrids. *Comparative Biochemistry and Physiology, Part C,* Vol.In press, (2011), ISSN 1532-0456

Shi, H., Schmidt, B., Liu, W., and Müller-Wittig, W. (2010). Quality-score guided error correction for short-read sequencing data using CUDA. *Procedia Computer Science,* Vol.1, (2010), pp. 1129-1138, ISSN

Trapnell, C., Pachter, L., and Salzberg, S.L. (2009). TopHat: discovering splice Junections with RNA-Seq. *Bioinformatics,* Vol.25, No.9, (May 2009), pp. 1105-1111, ISSN 1367-4811

Trapnell, C., Williams, B.A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M.J., Salzberg, S.L., Wold, B.J., and Pachter, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat Biotechnol,* Vol.28, No.5, (May 2010), pp. 511-515, ISSN 1546-1696

Voelkerding, K.V., Dames, S.A., and Durtschi, J.D. (2009). Next-generation sequencing: from basic research to diagnostics. *Clinical chemistry,* Vol.55, (2009), pp. 641-658, ISSN

Wall, D.P., Kudtarkar, P., Fusaro, V.A., Pivovarov, R., Patil, P., and Tonellato, P.J. (2010). Cloud computing for comparative genomics. *BMC bioinformatics,* Vol.11, (2010), pp. 259, ISSN

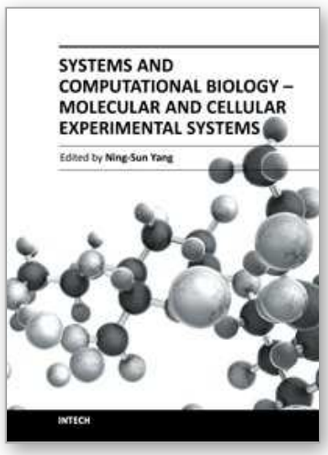
Walter, R.B., and Kazianis, S. (2001). Xiphophorus interspecies hybrids as genetic models of induced neoplasia. *ILAR J,* Vol.42, No.4, (October 2001), pp. 299-321, ISSN 1084-2020

Wang, L., Feng, Z., Wang, X., and Zhang, X. (2010). DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics,* Vol.26, No.1, (January 2010), pp. 136-138, ISSN 1367-4811

Wu, T.D., and Nacu, S. (2010). Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics (Oxford, England),* Vol.26, (2010), pp. 873-881, ISSN

Wu, T.D., and Watanabe, C.K. (2005). GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics (Oxford, England),* Vol.21, (2005), pp. 1859-1875, ISSN

Zerbino, D.R., and Birney, E. (2008). Velvet: algorithms for de Novembero short read assembly using de Bruijn graphs. *Genome Res,* Vol.18, No.5, (May 2008), pp. 821-829, ISSN 1088-9051



Systems and Computational Biology - Molecular and Cellular Experimental Systems

Edited by Prof. Ning-Sun Yang

ISBN 978-953-307-280-7
Hard cover, 332 pages
Publisher InTech
Published online 15, September, 2011
Published in print edition September, 2011

Whereas some â€œmicroarrayâ€ or â€œbioinformaticsâ€ scientists among us may have been criticized as doing â€œcataloging researchâ€, the majority of us believe that we are sincerely exploring new scientific and technological systems to benefit human health, human food and animal feed production, and environmental protections. Indeed, we are humbled by the complexity, extent and beauty of cross-talks in various biological systems; on the other hand, we are becoming more educated and are able to start addressing honestly and skillfully the various important issues concerning translational medicine, global agriculture, and the environment. The two volumes of this book presents a series of high-quality research or review articles in a timely fashion to this emerging research field of our scientific community.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yingjia Shen, Tzintzuni Garcia and Ronald B. Walter (2011). Gene Expression Analysis Using RNA-Seq from Organisms Lacking Substantial Genomic Resources, Systems and Computational Biology - Molecular and Cellular Experimental Systems, Prof. Ning-Sun Yang (Ed.), ISBN: 978-953-307-280-7, InTech, Available from: http://www.intechopen.com/books/systems-and-computational-biology-molecular-and-cellular-experimental-systems/gene-expression-analysis-using-rna-seq-from-organisms-lacking-substantial-genomic-resources

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen