BIOINFORMATICS

Vol. 00 no. 00 2011
Pages 1–6

# Gene expression estimation using RNA-Seq data

Wei-Yi Cheng *

Department of Electrical Engineering, Columbia University, 1300 S.W. Mudd, 500 West 120th Street, New York, NY 10027

May 5th, 2011

**ABSTRACT**

**Motivation:** RNA-seq sequences the steady-state messenger RNA in a cell. Using RNA-seq data for gene expression estimation promises several benefits which is inaccessible to previous technologies such as microarrays or PCR. Many applications has been developed to give expression estimation from RNA-seq data, each has its own strengths and weaknesses. In this work, we gave both quantitative and qualitative comparisons on several most popular tools. Then we developed a parallelized method to speedup the expession estimation process.

**Results:** We compared the performance of tophat, SOAP2, edgeR, and cufflinks using a prostate cancer dataset. The precision of predictions is estimated based on a set of PCR-varified differentially expressed genes. The prediction results suggest no matter what aligner is used, the negative-binomial distribution-based normalization tool edgeR seems to be able to have higher precision. We also implemented a python-based scriprt, sgeHTSeqCount, that aggregates aligned reads on each exon region to create a count table. The sgeHTSeqCount script can also be executed in parallel on a Sun Grid Engine to speedup the count aggregation process.

**Availability:** The implemented python script, sgeHTSeqCount, is attached in the attachment of the submission.

**Contact:** wc2302@columbia.edu

## 1 INTRODUCTION

The advent of high-throughput sequencing technology brings new dimensions to traditional microarray experiments. Sequencing messenger RNA, also known as RNA-seq, can sample with fewer biases than the expression microarrays. The data generated, processed as digital counts, can be used as a direct measure of the level of gene expression. It also provides higher resolution of the information. The RNA-seq data can be used not only for abundance estimation, the applications for alternative splicing, RNA editing and novel transcripts have also been developed recently (Oshlack *et al*., 2010).

Estimating gene expression from RNA-seq data requires several steps (Oshlack *et al*., 2010): **(1)Read alignment.** The first step is to find a unique location for each of the millions of short reads in the reference genome. Since the short reads are the results of sequencing mRNA, this task usually involves building a transcriptome library, that is, a library that contains the linkage of exons, or the discovery of *d*e novo junctions. **(2)Count aggregation.** After obtaining the aligned reads, we need to aggregate the reads over some biological meaningful unit, such as exons, transcripts or genes, depending on the purpose of the study. **(3)Normalization.** In order to acquire the significance of the expression levels between and within samples, the normalization procedure accounts for factors that may affect the number of reads that have been sequenced, such as the length of the gene or the library size, and estimates scaling factors to be used within the statistical models that test for differential expression. **(4)Differential expression identification.** Unlike microarrays, RNA-seq data gives a discrete measurement for each gene. To test for significant changes in abundance, a Poisson distribution is a basic model for the count data. Negative binomial distribution has also been used as an extension of the Poisson distribution, because it can account for biological variability by estimating a dispersion parameter.

In this work, we selected four popular tools used to process RNA-seq data, including two alingment tools and two summarization/normalization tools. We first give a qulitative comparison of their underlying mathematical models or data structures, Input/Output compatibilities, and special features. Then four piplines based on the combinations of the tools (2-by-2) were created. We actually test the pipelines on a prostate cancer (Li *et al*., 2008) dataset. Using 21 PCR verified differentially expressed genes, we estimated the precision of the pipelines. The selection of tools are: (1)SOAP2: an alignment tool that utilized an improved version of Burrows-Wheeler Transformation (BWT) compression indexing (Langmead *et al*., 2009), that can speedup the alignment while reducing memory usage. (2)TopHat: an alignment tool that performs alignment based on Bowtie, an ultra fast aligner, and also predicts *d*e novo splice junctions. (3)Cufflinks: A downstream tool of TopHat, it summarize the alignment results using both reference genome and *d*e novo junction regions discovered according to the alignment tools, then normalize the reads by calculating the reads per kilobase of exon model per million mapped reads (RPKM). (4)edgeR: a tool incorporated as part of the Bioconductor tools (http://www.bioconductor.org/). It uses the negative binomial distribution for statistical inferences on differential expression between two phenotypes.

After surveying the popular tools, we tried to implement a count aggregation tool that can be incorporated as part of the expression estimation pipeline. We used the library in the python package HTSeq (http://www-huber.embl.de/users/anders/HTSeq/doc/overview.html) tp build up a python script that can be executed either on a single machine, or on a Sun Grid Engine for parallel execution. The implemented script accepts the output of any general alignment tool (SAM or BAM file), and output a count table that can be input for statistical test for differential expression.

*to whom correspondence should be addressed

## 2 METHODS AND MATERIALS

### 2.1 RNA-Seq Data

The RNA-seq data used for evaluating the performance of gene experssion estimation is a prostate cancer dataset obtained from Li *et al*.. The dataset profiles double poly(A)-selected RNA from LNCaP cells before and after androgen stimulation. The sequencing was done using the Illumina GA I and produced 36 bp, single end, unstranded reads. The authors also performed quantitative real time PCR on a set of hormone-regulated genes, and found 21 genes that are significantly differentially experssed between treated and untreated samples (Table S1).

### 2.2 Short read alignement tools

*2.2.1 SOAP2* SOAP2 uses BWT compressed indexing to improve alignment speed while reducing memory usage. It determines an exact match by constructing a hash table to accelerate searching for the location of a read in the BWT reference index. For inexact alignment (either mismatch or indel), it splits a read into two or three fragments to allow one or two mismatches, respectively (Li *et al*., 2009). In our comparison, we used release 2.21 downloaded from http://soap.genomics.org.cn/soapaligner.html using default settings, which allows 2 mismatches when performing alignment.

*2.2.2 TopHat* TopHat first aligns the the reads to the reference genome using Bowtie. The unmapped reads are first set aside as "initially unmapped reads." After performing the first round of alignment, the mapped reads are used to assemble islands of contiguous sequence from the sparse consensus. TopHat uses these islands to infer putative exons. To map reads to splice junctions, it first considers all pairings of the islands, looking for (GT-AG) pairings between neighboring islands. Then each possible intron is checked against the unmapped reads for reads that span the splice junction. Therefore, the TopHat is able to align reads to reference without relying on knwon splice sites (Trapnell *et al*., 2009). We used version 1.2.0 downloaded from http://tophat.cbcb.umd.edu/. We perform the alignment with default settings, which up to allows 2 mismatches when aligning the reads, and allows no mismatches when aligning reads to the inferred spliced region.

### 2.3 Summarization/Normalization tools

*2.3.1 Cufflinks* Cufflinks takes the spliced alignments as input. The fragments are first connected in an 'overlap graph' when they are compatible and their alignments overlap in the genome. Each node in the graph stands for one fragment, and paths through the graph correspond to sets of mutually compatible fragments that could be merged into complete isoforms. Fragments are then matched to the transcripts from which they could have originated, producing the abundances in fragments per kilobase of transcript per million fragments mapped, or FPKM (Trapnell *et al.,* , 2010). Cufflinks pipeline can also deal with non-uniform distribution of the cDNA fragments due to the RNA-Seq library preparation by correcting the estimated expression levels using a likelihood based approach. Cufflinks then takes the log fold change of the FPKM's between two phenotypes, and computes the p-value by sampling from empirical distribution of fold changes 50,000 times for each transcript for statistical significance (Roberts *et al*., 2011). Here we used Cufflinks version 0.9.3 with bias correction enabled, with all default settings, to identify the differentailly expressed genes.

*2.3.2 edgeR* `edgeR` is a Bioconductor software package for examining differential expression of replicated count data. It first accumulates short reads counts into a count table by exon, transcripts, or gene level. Then it estimates the genewise dispersion by conditional maximum likelihood, conditioning on the total count of that gene, and shrinks the dispersions towards a consensus value using an empirical Bayes procedure. `edgeR` then fits the count table into a negative binomial distribution, and performs Fisher's exact test adapted for overdispersed data for assessing differential expression (Robinson *et al*., 2010). The version of Bioconductor we used is 2.8.

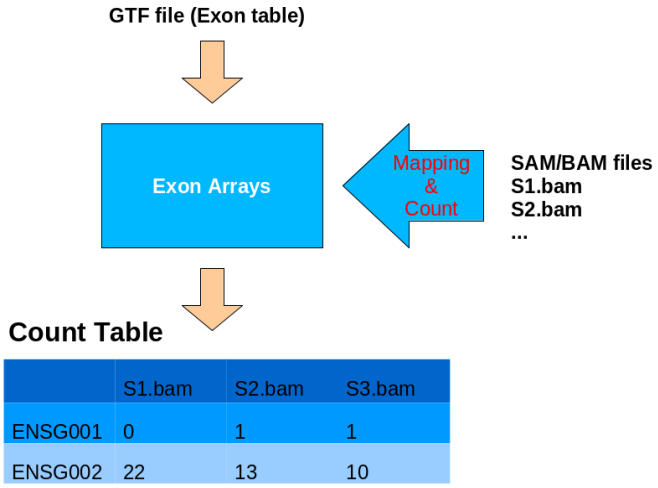


**Fig. 1.** Flowchart of `segHTSeqCount` on a single machine. The program first reads in GTF file, which contains all the locations of exons and the corresponding gene names in the genome, then create an exon array that maps the genomic intervals to the gene names. It then reads in the aligned reads to create a count table

### 2.4 Aligned reads accumulation

We created a python-based scripts, `sgeHTSeqCount`, to perfom reads accumulation. `sgeHTSeqCount` ultilizes the HTSeq package in python to create objects for the task. The script is able to be run in parallel on a Sun Grid Engine with the information of the segment ID and the number of total segments, thus reduce the execution time.

*2.4.1 Procedure overview* A GTF file that describes the genomic locations and features of the exons is first readed in. We uses this information to create an "exon array", which is actually a list that maps the genomic intervals to the gene names, and a count table whose row features correspond to the genes, and columns correspond to the samples. Then it reads in the aligned reads, adds 1 to the count table's corresponding entries. The output full count table can be used as an input to the edgeR to perform differentail expression analysis. The flowchart of the count accumulation procedure is given in Figure 1.

*2.4.2 Parallelize accumulation process* Assume we run `sgeHTSeqCount` on N nodes, as shown in Figure 2 as an illustration for N=4. We first let each node create its own local copy of the exon array. The rationale of localization of the exon array is that since the size of the exon array only depends on the size of genome, it is less scalable than the alignment files, which could increase to many times as large as the genome. Maintain a local copy of it is both feasible and also spends less the on accumulating counts than to distribute it. When reading in the alignment files, each node will be allocated tasks according to their segment ID: the i-th node will process every (N*k + i)-th read, where k is 0 or a natural number. Such dynamic task allocation makes sure that the tasks can be equally distributed even when we do not have prior knowledge on the size of the alignment files. By doing so, we expect the time for accumulating counts can be reduced to 1/N of the serial processing time. We tested `segHTSeqCount` on a Sun Grid Engine, with each computing node having 8 2GHz CPUs and 16GB memory.

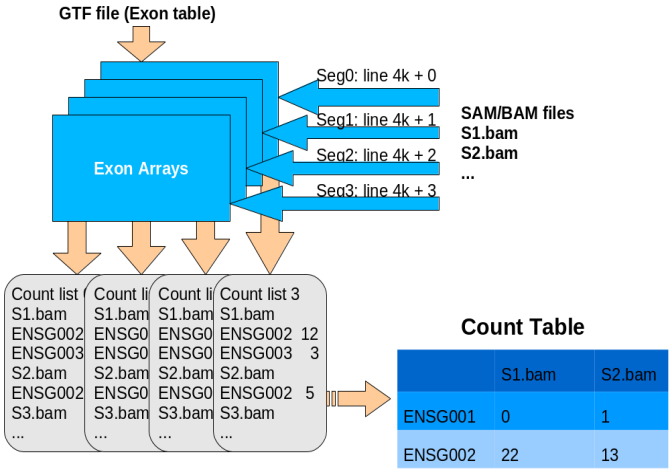


**Fig. 2.** Flowchart of `segHTSeqCount` running on 4 nodes. Each node will create a local copy of the exon array. Then when reading in the alignment files, the i-th node will process every (4k + i)-th read, thus reducing the accumulating time. The counts are recorded as a count list. After all nodes have finished the accumulation process, one node will remain working to merge the count lists into count table

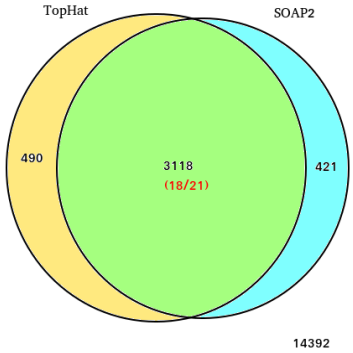


**Fig. 3.** Venn Diagram for overlapping genes between TopHat-edgeR and SOAP2-edgeR pipelines. 18 out of 21 qPCR-verified differentially expressed genes were identified in the overlapping region.

## 3 RESULTS

### 3.1 Qualitative comparison on tools

*3.1.1 SOAP2* SOAP2 is a simple, fast aligner whose execution time is around the same level as Bowtie. However, the output file format of SOAP2 is not the popular SAM or BAM format. The user has to transform the soap file format into SAM format if he/she wants to perform further analysis. It also uses different BWT indexing file format from that used by Bowtie, which requires users to build the indexing file from genomic sequence data. And SOAP2 does not support any kind of color spaced sequencing data.

*3.1.2 TopHat* The Bowtie aligner integrated in TopHat is the state-of-the-art short read alignment tool today. It guarantees the speed and memory efficiency. TopHat extends its feature with splice junction prediction, which increases the success rate of alignment, but also increases the execution time. The time required for aligning unmapped reads to splice junction is more than aligning reads to the original genome itself. And TopHat has known bug when processing ABI SOLiD color spaced sequencing data.

The GenePattern genomic analysis platform has recently started providing RNA-Seq data analysis. TopHat was incorporated as a tool for splice junction prediction, which makes it one of the few tools that can be used as a web application.

*3.1.3 Cufflinks* Cufflinks was designed to be a down-stream tool of TopHat. Therefore, it can perfectly support the output of TopHat and gives differential expression estimation not only on the exon or gene level, but also gives differential expression of alternative spliced genes or novel transcripts. It also includes a feature for sequencing bias correction. Cufflinks is also incorporated as one of the RNA-Seq data analysis tools in GenePattern.

*3.1.4 edgeR* As a package of Bioconductor, edgeR enables the user to import all bioinformatic tools and databases that has been incorporated into Bioconductor. After obtaining the list of differentially expressed genes, the user can directly use it for GO analysis or gene set enrichment analysis, all can be done in R environment. Dealing with the count table also makes it easy for user to keep track of the digital counts of the reads and to understand the procedure. The drawback is that it does not contain the feature of differential expression of splice junction, thus the analysis is confined in the known transcript database.

### 3.2 Analyze prostate cancer dataset

We used these two aligner and two summarization/normalization tools to create four pipelines (TopHat-Cufflinks, TopHat-edgeR, SOAP2-Cufflinks, SOAP2-edgeR). For each pipelines, we used the prostate cancer dataset as input, and compared the output differentially expressed gene list (Table S2 - S5). We then identified the overlapping of two different aligners for each summarization/normalization tools and use it to create the Venn diagrams of identified gene sets.

As shown in Figure 3, when we use edgeR for summarization and normalization, no matter which aligner we use, we are able to identify about a quarter of the genes that are differentially expressed. The gene sets of two aligners are highly overlapped. And when we cross reference the gene sets to the list of 21 qPCR-verified differentially expressed genes in Table S1, we identified 18 genes that exist in the overlapping region. The p-value for the enrichment of the 21 genes in the overlapping region is $9.86 \times 10^{-12}$.

When we use Cufflinks for summarization and normalization, the genes identified as differentially expressed are much less. As shown in Figure 4, there are discrepencies between using two different aligners. Only 2 genes were overlapped. As for the qPCR-verified genes, using TopHat as aligner can identified 5 (p-value = $2.48 \times 10^{-6}$), while using SOAP2 can only identify one gene (p-value = 0.0036).

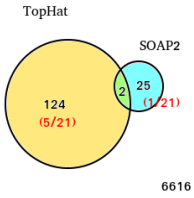


**Fig. 4.** Venn Diagram for overlapping genes between TopHat-Cufflinks and SOAP2-Cufflinks pipelines. Among the 21 qPCR-verified differentially expressed genes, five were identified in the TopHat-Cufflinks gene set, and only one was identified in the SOAP2-Cufflinks gene set.

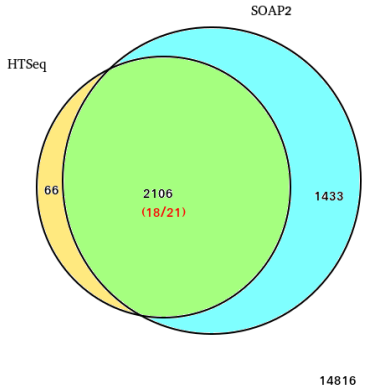


**Fig. 5.** Venn Diagram for overlapping genes between SOAP2-edgeR and Bowtie-HTSeq-edgeR pipelines. Two gene sets are highly overlapped, while HTSeq pipeline only gives two thirds of the number of genes given by SOAP2-edgeR pipeline. The same 18 out of 21 qPCR-verified genes appear in the overlapping region of the sets.

It seems using edgeR for summarization and normalization gives higher power for differential expression detection. And the power of using Cufflinks is much less. It could be the bias correction feature in Cufflinks had also eliminated some real counts.

### 3.3 Using sgeHTSeqCount as a count accumulation tool

We used the SAM files output by Bowtie (an intermediate output of TopHat) as an input to sgeHTSeqCount. After obtaining the count table, we processed it using edgeR. We first observe that the counts in the count table produced by sgeHTSeqCount is less than that created by IRange, which is the count accumulating function in Bioconductor GenomicFeature package. This is due to the sgeHTSeqCount only increases the count when a read is perfectly covered within the genomic interval of an exon, while in IRange it allows some inconsistencies at the ends of the interval. The less counts creates a differentail expression sets with less genes. As shown in Figure 5, the pipeline only gives 2,172 genes compared with the 3,539 genes given by the SOAP2-edgeR pipeline. However, there are large amount of genes that are overlapping. And the same 18 out of the 21 qPCR-verified genes appeared in this overlapping region. If we evaluating the performances of predictions based on precision, it seems the pipeline with sgeHTSeqCount outperformed the other pipelines.

However, when we measured the time requires for accumulating counts, sgeHTSeqCount took about 395 seconds, while using IRange only takes 62 seconds. It is this timing issue that motivated us for parallelizing the sgeHTSeqCount procedure.

### 3.4 Parallelize the count accumulation process

We applied the parallelization strategy described in Methods, and obtained the timing shown in Table 1. We also calculate the speedup of the accumulating time by the equation $Speedup_{Nsegs} = \frac{T_{1seg}}{T_{Nsegs}}$. We can see the speedup gradually flattened as the number of nodes increase. This is mainly caused by the fact that our parallelization strategy requires each node still to go through all files. Therefore, the limit of processing time is the time that a node read through every line of all the files and do nothing. Our test shows that doing so will take 122.12 seconds, which is very close to the time we run on 100 nodes. Therefore, as the short read length is rather short (36 bps), and each BAM file contains not so many reads, we will not see a significant decreasing in time when the number of nodes are large. The maximum speedup we can obtain using this small data is around 3.

**Table 1.** Timing for running sgeHTSeqCount on different number of nodes.

| Time (sec)/# of Nodes | 1 | 2 | 4 | 8 | 16 | 100 |
|---|---|---|---|---|---|---|
| Counting | 394.44 | 261.65 | 202.42 | 165.07 | 139.96 | 127.29 |
| Merge Files | 0 | 0.29 | 0.38 | 0.48 | 0.62 | 11.02 |
| Total | 527.08 | 393.58 | 334.87 | 296.55 | 270.37 | 257.76 |

## 4 DISCUSSION

One important issue that always deters computational biologists from making biological hypothesis is the results of analysis drastically vary when using different approaches. The reason we would advocate the edgeR method, and used it for processing the output of sgeHTSeqCount is that it gave consistent results no matter which upstream alignment tool has been used. The Cufflinks correcting the abundance by referring to gene expression levels and transcript length seems to have negative effects on the prediction it makes. But the negative effects may also be data-dependent. Since the test dataset we used is rather small, the number of total reads is

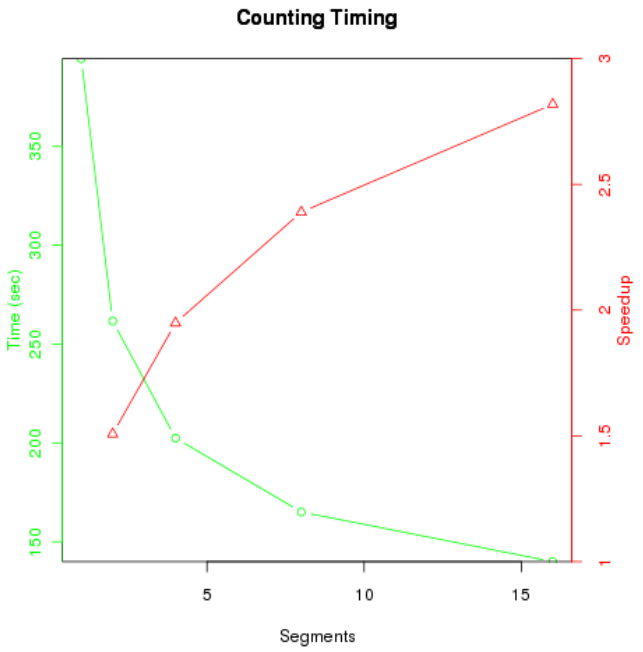


**Fig. 6.** Count accumulating time and speedup when running sgeHTSeqCount on different numbers of nodes.

| Tophat | Soap2 |
|---|---|
| + Novel junction prediction<br>+ Integrate Bowtie, Samtools<br>+ GenePattern<br><br>- Slower<br>- Bugs exist when dealing with color space data | + Fast and clean alignment (3X faster than tophat)<br><br>- Takes longer building indexing files<br>- Doesn't support color space |
| **Cufflinks** | **edgeR** |
| + Junction-based differential expression estimation<br>+ Support several native table format from UCSC<br>+ GenePattern<br><br>? Sequencing bias, mask correction (too conservative?)<br>- Output requires further processing | + R-package, Bioconductor support<br>+ GO integration<br><br>- No differential splice junction detection |

**Fig. 7.** Summarization of pros and cons of the selected RNA-Seq tools.

also small, the effects of any correction could have significant effect than anticipated. Therefore, the test on a larger dataset is definitely needed before we reach the conclusion of the negative effects of bias correction.

RNA-Seq data brings more dimensions and higher resolution on a single experiment. However, it also comes with enormous amount of data. In microarray experiments, to obtain statistical significance from the high dimentional data, it usually requires dozens of samples. When obtaining the RNA-Seq data from the same number of samples, the size of data would easily reaches the level of TB. Therefore, it is also essential to process the data in parallel in order to stay under the limit of the computer. In this work we demonstrated that each node is only required to store a fixed size of table and lists, which are proportional to the size of genome, it is able to process all sequence data under this limitation. We will expect to see the parallelization still work when the number of sample increases as well as the number of reads. And it would also be demanded to parallelize th whole pipeline, from alignment till count accumulation, in order to have a more efficient local memory usage and for further speedup, which is a priority before we officially step into the "digital gene expression" era.

## 5 CONCLUSION

We selected two short read aligner and two summaeization tools to create four pipelines. We first compared their individual features and then compared their performance as pipelines based on a prostate cancer dataset. The advantages and disadvantages have been summarized in Figure 7. The goal of surveying the tools is to give a recommendations on tools when performing RNA-Seq to gene expression analysis in the future. The TopHat-Cufflinks pipeline, which is one of the most popular tools tackling this problem, is still a good choice for those without programming background to save the trouble installing packages or learning programming syntax. The binary package of both is easy to use, and what may be better, the new GenePattern web-based application can even save more trouble while getting additional database support. The edgeR package is for users who needs to keep track of the data processing, while some additional tests on the intermediate output might be necessary. The variety of statistical tools in R and Bioconductor also makes it a powerful choice in the later stage of analysis. As for SOAP2, it could still be a second choice of alingment tool following Bowtie, due to its lack of support for some data format.

We also started parallelizing the count accumulation process. The effect of the parallelization is only obvious when the number of nodes are small due to our small input data. We would expect to increase the speedup by further parallelize upstream alginment process. This will be a priority of our work in this area.

## REFERENCES

Oshlack,A., Robinson MD, Young MD (2010) From RNA-seq reads to differential expression results, *Genome Biology*, **11**:220.

Li H, Lovci MT, Kwon YS, Rosenfeld MG, Fu XD, Yeo GW (2008) Determination of tag density required for digital transcriptome analysis: Application to an androgen-sensitive prostate cancer model, *PNAS*, **105**, no. 51, 20179-20184.

Langmead B, Trapnell C, Pop M, and Salzberg SL (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome, *Genome Biology*, **10**:R25.

Li R, Chang Y, Li Y, Lam TW, Yiu SM, Kristiansen K, and Wang J (2009) SOAP2: an improved ultrafast tool for short read alignment, *Bioinformatics*, Vol. 25, no. 15,

1966-1967.

Trapnell C, Pachter L, and Salzberg S (2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, Vol. 25 no. 9, 1105-1111.

Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, Baren MJ, Salzberg S, Wold BJ, and Pachter L (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, Vol 28 no. 5. 511-515.

Roberts A, Trapnell C, Donaghey J, Rinn JL, and Pachter L (2011) Improving RNA-Seq expression estimates by correcting for fragment bias. *Genome Biology*, **12**:R22.

Robinson MD, McCarthy DJ, and Smyth GK, (2010) edgeR: a Bioconductor package for differentail expression analysis of digital gene expression data *Bioinformatics*, Vol. 26 no. 1, 139-140.