

GENE EXPRESSION VARIATION UNDER
DYNAMIC LIGHT IN *Arabidopsis thaliana*

Kevin Murray

Borevitz Lab, ANU

Tuesday 15th October, 2013

Thesis submitted in partial fulfillment of the requirements of the degree

of

Bachelor of Philosophy (Science) (Honours)

Word Counts:

Introduction: x words

Results: y words

Discussion: z words

Abstract

This is the abstract. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Contents

Todo list	1
1 Development of Improved Methodology for High-throughput RNAseq Experiments	5
1.1 Aims and Hypotheses	5
1.2 Methods	6
1.2.1 Implementation of a High Throughput RNAseq Library Preparation Method	6
1.2.2 External RNAseq Datasets	8
1.2.3 Development of an Improved Analysis Pipeline	9
1.2.4 Measuring the Effect of Sequencing Depth on Analysis of Differential Expression	10
1.3 Results	11
1.3.1 High Throughput RNAseq Library Preparation Protocol	11
1.3.2 A Framework for the Creation of RNAseq Analysis Pipelines	12
1.3.3 An Improved Analysis Pipeline for Large Plant RNAseq Datasets	15
1.3.4 Comparison of Differential Expression Pipelines	17
1.3.5 Substantial reduction of RNAseq coverage is possible	18
2 Appendix	24

Acknowledgements

Thanks everyone!

write Acknowledgements

Questions to people reading this

- Does all code have to go in the appendix? This could make it **VERY** long. Or is it ok to give links to github or similar (somewhere publically available).

Todo list

write Acknowledgegements	2
reword this	5
lots of details missing here	8
check details of this section from Pete	8
part numbers etc	9
Pete: get this from pete, he said I can use one from his PEB slides.	9
this section needs a re-write. Also questioning if it should go here, or chapter 4.	12
write up old comparison R document into thesis document .	18
will be done Wednesday, found bug and needs to re-calculate	18

Chapter 1

Development of Improved Methodology for High-throughput RNAseq Experiments

1.1 Aims and Hypotheses

In this chapter of my thesis, I am to discuss and provide solutions to some bioinformatic challenges in the analysis of RNAseq datasets. Specifically, I am to identify issues arising as a result of two aspects of biological experiments like the one which I have conducted this year: *a*), the dramatic changes in global expression on changing the growth, and particularly the light conditions of a plant, and *b*), the need for high throughput, low cost RNAseq experiments to match the high throughput required for mapping experiments. _____

reword this

1.2 Methods

1.2.1 Implementation of a High Throughput RNAseq

Library Preparation Method

To increase the throughput and the decrease cost of Illumina RNAseq sequencing library preparation, I have attempted to implement the RNAseq library preparation protocol of Kumar et al. (2012), a published protocol enabling the preparation of RNAseq libraries in high throughput using 96 well plates. Specifically, I used a slightly modified version of the High-Throughput RNAseq protocol described in Supplementary Methods 1 of Kumar et al. (2012) (hereafter referenced as the HTR protocol). To test and optimise this protocol, leaf tissue collected from surplus 5-week old *A. thaliana* Col-0 from a colleague's experiment was used. This tissue had been collected into Qiagen 1.2mL collection tubes (PN:<++>) containing a single steel ball bearing, and snap frozen in liquid N₂ before grinding in a TissueLyser (PN <+get details+>) for two one minute pulses at 25Hz at a later date. In collection tubes, 750 μ L Dynabeads Lysis/Binding Buffer was added, before sample was ground for a further 30s in a TissueLyser as before. Then, lysates were prepared per Steps 1.1.6-1.1.8 of the HTR protocol described in Supplementary Methods 1 of Kumar et al. (2012).

Isolated mRNA was obtained and cDNA synthesised and fragmented according to steps 1.2 to 3 of the HTR protocol of Kumar et al. (2012). Working with Dr. Norman Warthmann, the remaining steps of the HTR protocol were validated using sonicated genomic DNA as input material, due to it's similar size and fragmentation properties to fragmented cDNA, and due to the scarcity of cDNA of little value. This DNA was obtained from *Oryza sativa*

seedlings, diluted to a concentration of approximately $7 \text{ ng}\mu\text{L}^{-1}$ (500 ng in $70 \mu\text{L}$) before being sonicated for $<+settings+>$ on a Diagenode Bioruptor DNA sonicator. This sonicated DNA was cleaned up per steps 3.5-3.8 of the HTR protocol, with the modification that $30 \mu\text{L}$ bead binding buffer and $40 \mu\text{L}$ Ampure XP SPRI beads were used. Then, a modified step 4 of the HTR protocol was used to create unamplified sequencing libraries. In step 4.1 and 4.2, double reactions were performed, however the same quantity of SPRI cleanup reagents were used. Before adaptor ligation, the A-tailed libraries were eluted using a mixture of $5 \mu\text{L}$ diluted adaptor oligonucleotide, $1 \mu\text{L}$ 10x ligation buffer and $2 \mu\text{L}$ water. The DNA ligase was diluted in the remaining $1.5 \mu\text{L}$ water and added to each reaction, before proceeding with protocol steps 4.3.3 onwards. The adaptors used were not those specified by Kumar et al. (2012), instead custom bell adaptors were used. These adaptors were designed by Dr Norman Warthmann, and are compatible with the T/A overhang ligation method Kumar et al. (2012) utilise. These adaptors are described in ???. The protocol described in step 5 of the HTR protocol was used, with the forward and index 1 reverse primers (see ???), to amplify the libraries.

RNA quality and quantity was assayed using the Agilent BioAnalyser digital electrophoresis system. The RNA samples were loaded into a Plant RNA Pico analysis chip and an analysis run per manufacturer's protocol. The effectiveness of various steps in this protocol was assayed by digital electrophoresis with the Shimadzu MultiNA instrument, using the DNA1000 kit. The pre-mix protocol was used: $2 \mu\text{L}$ sample was added to $4 \mu\text{L}$ DNA1000 marker solution, the solution mixed, and loaded into the instrument, which was run according to manufacturer's DNA1000-PreMix protocol. Quantitative PCR was performed to test the ligation and PCR effi-

ciency of each adaptor. To do so, 2 μ L of each pre-amplification library was combined with 5 μ L Sybr Green qPCR master mix, 1 μ L each of the forward and index 1 reverse primers (see ??), and 1 μ L Uracil-Specific Excision Reagent (USER) enzyme mix.

lots of details missing here

1.2.2 External RNAseq Datasets

Two RNAseq datasets from collaborators were used both as trial datasets and external references in this thesis. I unambiguously disclaim that Peter Crisp, of the Pogson lab, Research School of Biology, ANU, created these datasets, and authorised their use in this thesis.

check details of this section
from Pete

The Rapid Recovery Gene Silencing excess light time-course experiment (hereafter referred to as the RRGs time-course) consists of samples taken in triplicate from an eleven-point excess light stress and recovery time-course. *A. thaliana* ecotype Col-0 were grown for 3 weeks under standard laboratory growth conditions ($\approx 150 \mu \text{mol photons } m^{-2} s^{-1}$ light intensity, 12 hour photoperiod, 21 °C daytime temperature, 21 °C nighttime temperature). Whole rosette samples were taken before any treatment, after 30, 60 and 120 minutes of 8x excess light ($1000 \mu \text{mol photons } m^{-2} s^{-1}$, unfiltered light from a sodium vapour lamp, hereafter EL), after 60 minutes of EL followed by 7.5, 15, 30 and 60 minutes of recovery under standard growth conditions, after 60 minutes of EL, followed by 60 minutes of recovery, followed by another 60 minutes of EL, and before and after 60 minutes of EL 24 hours after the original 60 minutes of EL. This complex time-course is illustrated in [Figure 1.1](#). RNA extracted from five plants per replicate was pooled, before Illumina libraries were created using the TruSeq V2 library preparation kit () per manufacturer's instructions. These libraries were sequenced across two Illumina HiSeq 2500 sequencing lanes,

yielding the RRGs timecourse RNAseq dataset. This dataset studies a timecourse over a treatment highly similar to that conducted in the dynamic growth condition experiment, allowing development of bioinformatic protocols specific to RNAseq datasets from experiments like these.

part numbers etc

A second dataset, the excess light and drought (EL/D) dataset, was used as a second dataset. This dataset consists of RNAseq libraries sampled in triplicate from plants grown to three week of age under standard laboratory growth conditions, per the RRGs timecourse experiment above. Plants were harvested before any stress, after 1 hour of 8x excess light, and after one week of drought stress, in which water was completely withheld while plants continued to grow under standard laboratory growth conditions.

Pete: get this from pete, he said I can use one from his PEB slides.

1.2.3 Development of an Improved Analysis Pipeline

Bioinformatic experiments were used to validate pipelines against the “gold standard” RNAseq analysis pipeline. In these experiments, programs selected through both literature review and searches of pre-publication software releases (e.g. software on github.com) were tested against a published best-practice pipeline (Van Verk et al. 2013). Specifically, the computational speed and efficiency, and the results obtained with these newer programs were compared to the analysis pipeline of Van Verk et al. (2013). This enables the development of higher-performance analysis pipelines suitable to high throughput experiments, with no cost to the quality of results obtained.

Comparisons between the computational cost of four pipelines were conducted using a sub-sampled dataset. To demonstrate the improved performance of the `aln_subread` pipeline, it was com-

pared to the `aln_tophat`, `aln_tophat_htseq` and `aln_subread_htseq` pipelines. The `time` UNIX command was used to summarise the computational cost of these four pipelines across five identical, independent, non-simultaneous runs. Four time-points of the RRGs timecourse dataset were sub-sampled to 500,000 reads by running `seqtk sample -s 10 500000` on both forward and reverse read files, which extracts 500000 random read pairs preserving read pairing. An ANOVA analysis was performed to find significant differences in runtime and CPU utilisation between analysis pipelines (see ??).

To ensure that the `subread` aligner and `featureCounts` produced comparable results to the analysis pipeline of Van Verk et al. (2013), several diagnostic measures were used. Firstly, the percentage of reads mapped to the genome, and to protein coding loci within the genome was computed and compared. Then, sample-wise correlations between tag-wise counts calculated by each pipeline were calculated. Finally, genes called differentially expressed by each pipeline were compared. These measures allow verification of pipeline performance at three major stages in an analysis pipeline: alignment of short reads to a genome, tag-wise count summarisation, and statistical testing for differential expression.

1.2.4 Measuring the Effect of Sequencing Depth on Analysis of Differential Expression

Six samples from the RRGs-Timecourse experiment (see subsection 1.2.2) were sub-sampled to allow investigation of the effect of sequencing depth on statistical power. To do so, the command `seqtk sample -s 10 X` was run on each pair of read files for these six samples, with X (number of reads to sample) set to 1000, 10000,

20000, 50000, 100000, 200000, 500000, 1000000, 2000000, 5000000 and 10000000. This sub-sampled dataset allows for titration of the optimal sequencing depth (or multiplexing level) for high throughput experiments, balancing sequencing cost with statistical power.

For each subsampled dataset, the `km_subread` pipeline followed by the `de_pairwise` pipeline were applied to find differential expression between the control and 30 minute excess light timepoints (these pipelines are described in [subsection 1.3.3](#)). Several metrics were then used to summarise the effect of sequencing depth on the statistical power of differential expression analysis. The number of tags called as differentially expressed at each sequencing depth was calculated, as was the common biological coefficient of variation. A third measure, the log-transformed mean expression level of the least-expressed differentially expressed gene and the overall least-expressed gene were calculated. These metrics were plotted against sequencing depth to give a graphical overview of the effect of reduction of sequencing depth.

1.3 Results

1.3.1 High Throughput RNAseq Library Preparation Protocol

To increase the throughput and the decrease cost of Illumina RNAseq sequencing library preparation, non kit-based protocols must be used. This subsection of my thesis describes my attempts to implement the RNAseq library preparation protocol of Kumar et al. (2012). The majority of this protocol was successfully implemented. Messenger RNA was successfully extracted in a 96 well plate format using oligo-d(T) magnetic beads, albeit with low yield (see [Figure 1.2](#)). Complimentary DNA (cDNA) was successfully pre-

pared from this mRNA. Enzymatic fragmentation of cDNA was successful, as was end repair and A-tailing. However, a cumulation of possible ligation and PCR biases caused order-of-magnitude variation in library abundance. This variation in library abundance was confirmed with diagnostic qPCR. For this reason, the use of this protocol in my experiment was abandoned, as the optimisation of these difficulties may have been very time consuming and was not feasible in the time remaining in my Honours year.


this section needs a re-write.
Also questioning if it should
go here, or chapter 4.

1.3.2 A Framework for the Creation of RNAseq Analysis Pipelines

In order to allow easy creation of diverse analysis pipelines for the multitude of possible experimental designs and method, I have implemented a generic framework for the creation of RNAseq data analysis pipelines. This framework takes the form of “wrapper scripts”, which act as wrappers around programs which other authors have created, and “pipeline” scripts, which combine these wrapped programs to perform an analysis.


Wrapper scripts are the workhorses of any pipeline created with this framework. All wrapper scripts accept three arguments: an input directory, and output directory, and arguments to be passed to the underlying program. Given these three pieces of information, the wrapper script will run the underlying program, automatically detecting input files from the input folder, and automatically handling different experimental features such as single vs paired-end sequence data. Wrapper scripts abstract away the complexity of command syntax, increasing readability and reproducibility of results.

Pipeline scripts describe processes of analysis of RNAseq data.




Placeholder
Figure!!!

Figure 1.1: Illustration of the RRGs timecourse. This figure was created by Peter Crisp, and is reproduced with his permission




Placeholder
Figure!!!

Figure 1.2: BioAnalyser digital gels of 10 RNA samples extracted. The extraction or quantification of C1 failed.



Placeholder
Figure!!!

Figure 1.3: BioAnalyser digital electrophoretogram of a representative mRNA sample (Sample B1 in [Figure 1.2](#)). Note the smear-like quality of the mRNA sample, and the reduced or absent ribosomal RNA peaks, when compared with a total RNA sample such as in ??



Placeholder
Figure!!!

Figure 1.4: Ligation bias between indices

They combine wrapped programs together to perform an analysis specific to the process or dataset in question. Several generic and some dataset-specific pipelines have been created. These pipelines are described in the following subsection (1.3.3).

1.3.3 An Improved Analysis Pipeline for Large Plant RNAseq Datasets

A series of pipelines to analyses RNAseq datasets of different kinds have been developed. As each RNAseq experiment has subtle differences in experimental design or library construction, developing a one-size-fits-all pipeline is not possible. Thus, a series of pipelines to encompass a variety of RNAseq experiments have been created. These pipelines are two-step pipelines; step one takes raw sequence reads, and produces summarised gene-wise counts. Step two applies statistical normalisation techniques and tests for differential expression. When applied combinatorially, these pipelines allow for different experimental designs to be analysed.

The `aln_subread` pipeline

This pipeline is built around the `subread` aligner, a very fast RNAseq-compatible short read aligner. Firstly, quality of sequencing data is checked using the `fastqc` program, sequencing adaptors are removed with `scythe`, and `seqtk` remove low quality sequences, before the quality is again checked using `fastqc`. The `subread` aligner then aligns reads to the TAIR10 *A. thaliana* genome, accounting for splicing. The resulting SAM file is converted to the BAM format, sorted and indexed, as required by some downstream programs (e.g. the IGV genome browser). Gene expression is then summarised gene-wise by counting the number of reads which align to genic loci with `featureCounts`, completing this section of the

analysis and the `aln_subread` pipeline.

The `aln_subjunc` and `aln_tophat` pipelines

For studies examining alternative splicing of mRNA transcripts, an aligner able to detect splicing *de novo* is required. `Tophat2`, one of the most popular RNAseq aligners, is able to align short reads while detecting slicing isoforms. `Subjunc` is an extension to the `subread` aligner which it allows it to do so. The `aln_subjunc` and `aln_tophat` are identical to the `aln_subread` pipeline, aside from their use of the `subjunc` and `Tophat2` aligners respectively, in place of the `subread` aligner, allowing study of alternative splicing. However, this *de novo* detection of splicing comes at a performance cost, and is not necessary for simple quantitation of gene expression.

`de_pairwise`

Where the experimental design is simple, statistical tests can be performed pairwise between samples. To this end, the `de_pairwise` pipeline implements these tests using the `edgeR` R package. This pipeline first reads count files into a `DGEList` object, then normalises counts using the TMM normalisation method of Robinson and Oshlack (2010). Common and tag-wise (i.e. gene-wise) dispersion are then calculated using the `calcNormFactors`, `estimateCommonDisp` and `estimateTagwiseDisp` functions respectively, yielding a `DGEList` object containing normalised counts and estimates of expression variability. Tests are then conducted pairwise between groups described in the keyfile, from data in this `DGEList`, using `exactTest`. This creates a `list()` of `DGEEExact` objects, from which tables of differential expression and diagnostic plots can be created. A plot showing the relationship between the tag-wise Biological Coefficient of Variation (BCV) and tag expression.

de_glm

If the experimental design is not simplistic, for example if multiple experimental factors (variables such as growth condition, treatment, block, or genotype) exist, pairwise analysis is inadequate. Thus, the more statistically complex Generalised Linear Model based hypothesis testing functions of **edgeR** are required. This pipeline takes a keyfile describing the experimental design as above pipelines do, however it takes an additional R script which describes the statistical model to be fitted, and contrasts within this model to be tested for differential expression. Tag-wise read counts are normalised and dispersions calculated with the GLM-based analogous of the functions used to do so in the **de_pairwise** pipeline. Then, a generalised linear model is fitted with **glmFit**, creating a **glm** object. Then, **glmLRT** is used to test each constant specified in the model script for differential expression. Analogous plots and tables to those produced in the **de_pairwise** pipeline are then produced.

1.3.4 Comparison of Differential Expression Pipelines

There was a highly significant difference between the computational cost of four pipelines (**aln_subread**, **aln_tophat**, **aln_tophat_htseq** and **aln_subread_htseq**). Using an ANOVA model with Tukey’s HSD post-hoc testing, significant differences in “real” time, “user” time and “sys” time were uncovered. As is shown graphically in **Figure 1.5**, the **aln_subread** is the fastest, taking an average of 3.48 ± 0.10 minutes to complete, followed by the **aln_subread_htseq** pipeline, which took 4.88 ± 0.07 minutes. The **aln_tophat** and **aln_tophat_htseq** were almost 400% slower, taking 12.9 ± 0.12 and 13.77 ± 0.07 minutes of real time respectively. The computational cost of user code and kernel processes in CPU-minutes

followed similar patterns, as detailed in [Figure 1.5](#).

Quantification of gene expression by the `aln_subread` pipeline is comparable to that obtained by the `aln_tophat` pipeline. As shown in [Figure 1.6](#), there is a very tight relationship between counts produced by the Tophat2 and subread aligners. The slope of $\log(n+1)$ transformed raw count data when the model *tophatcounts subreadcounts* is fitted is 0.994, with $p < 2e-16$ and R^2 of 0.993.

1.3.5 Substantial reduction of RNAseq coverage is possible

Sequencing is expensive, and multiplexing many samples per lane is important for high-throughput transcriptomics. However, in RNAseq, an optimality exists between sequencing depth per sample and statistical power: as the number of reads per sample decreased, the number of genes called as differentially expressed decreased in a non-linear fashion ([Figure 1.7](#)). Below 5 million reads, the number of genes called as differentially expressed reduces rapidly. Additionally, the median tag-wise mean log expression reduces rapidly below approximately 1 million reads, indicating that at very low coverage, lowly expressed genes begin to not be detected at all ([Figure 1.8](#)). Finally, the common biological coefficient of variation, which indicates how variable a dataset is as a whole, increases as sequencing depth decreases ([Figure 1.9](#) and Appendix ??). For the model system used in this experiment, I would recommend an optimal sequencing depth of approximately 5 million reads (or read pairs) to balance statistical power against sequencing cost.

write up old comparison R document into thesis document

will be done Wednesday, found bug and needs to recalculate

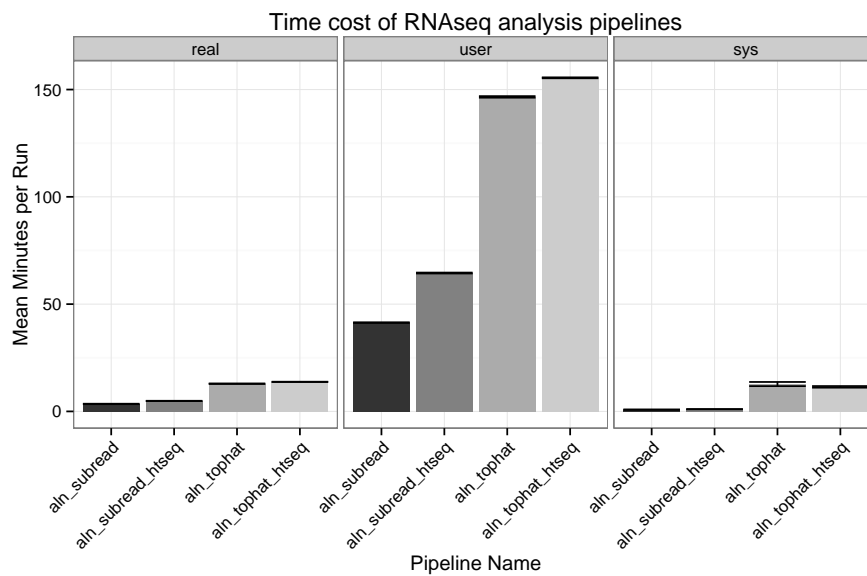


Figure 1.5: Computational cost of the `aln_subread`, `aln_tophat`, `aln_tophat_htseq` and `aln_subread_htseq` RNAseq analysis pipelines differs significantly. The “real” computational cost describes the number of seconds each pipeline took to complete. The “user” and “sys” metrics describe the number of CPU-minutes spent running user code (i.e. the pipeline components) and performing kernel operations on behalf of user code (e.g. input/output, memory (de)allocation and other system calls) for each pipeline execution.

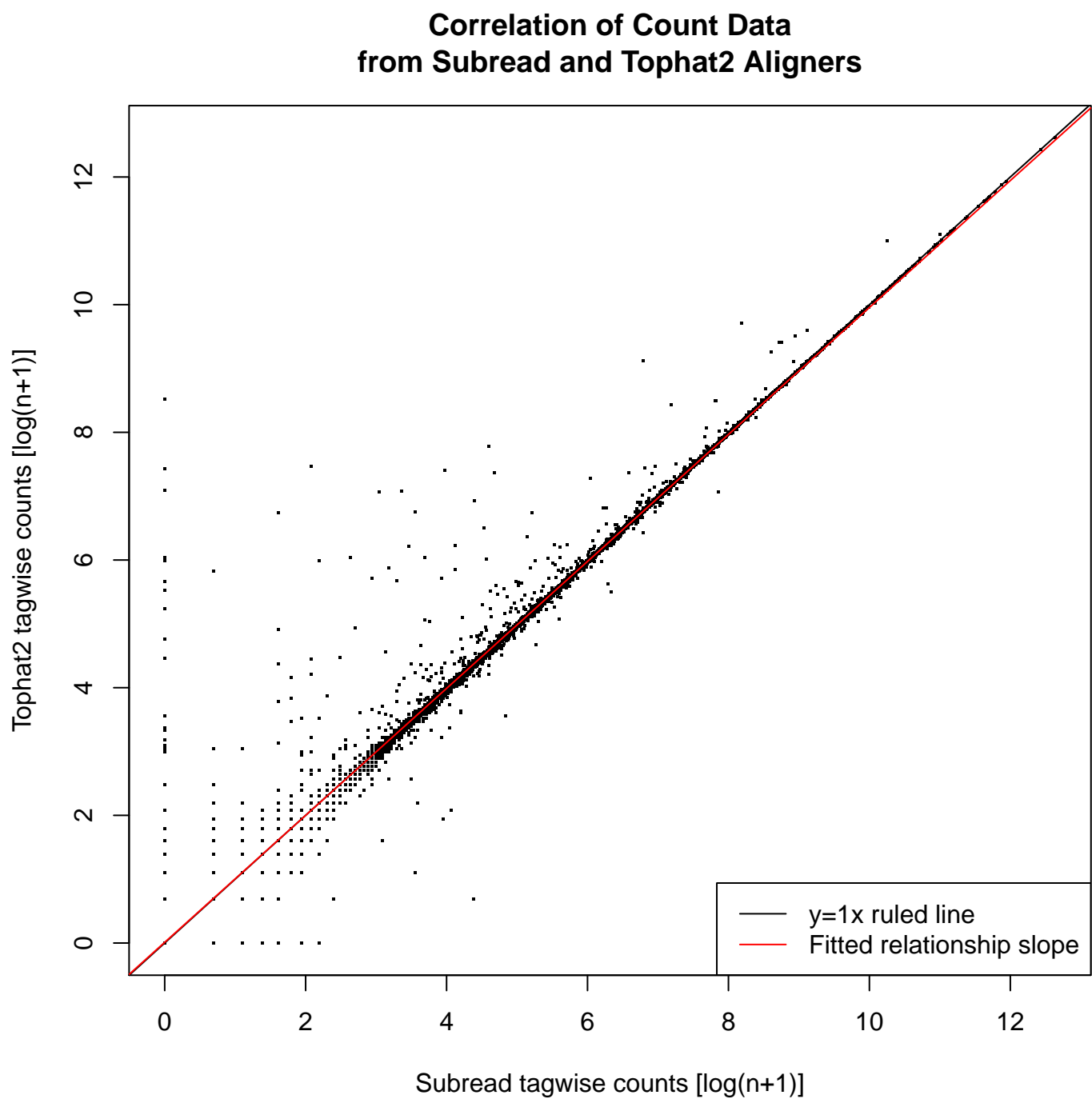


Figure 1.6: A very tight relationship is observed between count data generated by the tophat and subread aligners.

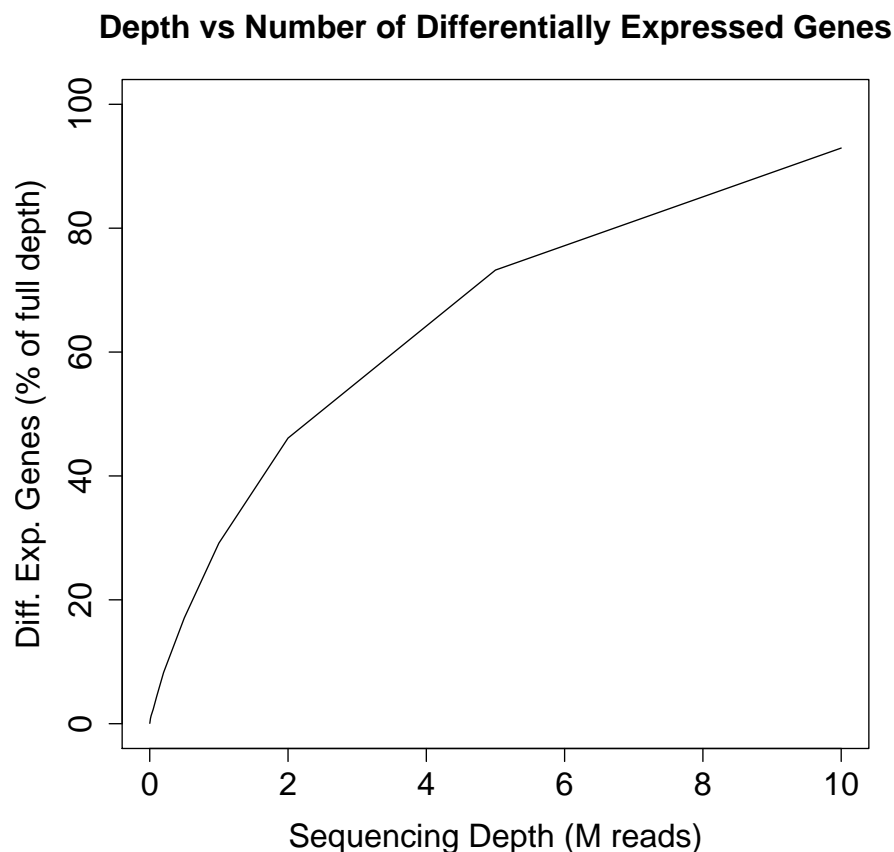


Figure 1.7: Decreasing sequencing depth per sample decreases the number of tags called as differentially expressed. This occurs because the total number of genes which can be examined for differential expression decreases in a similar fashion.

**Placeholder
Figure!!!**

Figure 1.8: Median tag-wise expression vs sequencing depth

Placeholder
Figure!!!

A yellow rectangular box with a red border containing the text "Placeholder Figure!!!".

Figure 1.9: Common biological coefficient of variation vs sequencing depth.

Bibliography

- Kumar, R, Ichihashi, Y, Kimura, S, Chitwood, DH, Headland, LR, Peng, J, Maloof, JN, and Sinha, NR (2012). A high-throughput method for Illumina RNA-Seq library preparation. *Frontiers in Plant Genetics and Genomics* 3, p. 202. DOI: [10.3389/fpls.2012.00202](https://doi.org/10.3389/fpls.2012.00202) (cit. on pp. [6](#), [7](#), [11](#)).
- Robinson, MD and Oshlack, A (Mar. 2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* 11.3. PMID: 20196867, R25. DOI: [10.1186/gb-2010-11-3-r25](https://doi.org/10.1186/gb-2010-11-3-r25) (cit. on p. [16](#)).
- Van Verk, MC, Hickman, R, Pieterse, CM, and Van Wees, SC (Apr. 2013). RNA-Seq: revelation of the messengers. *Trends in Plant Science* 18.4, pp. 175–179. DOI: [10.1016/j.tplants.2013.02.001](https://doi.org/10.1016/j.tplants.2013.02.001) (cit. on pp. [9](#), [10](#)).

Chapter 2

Appendix

Notes:

- Code listings, where included, are illustrative. Full source code of all software developed is large (over 5000 lines of code), and will be distributed as a gzipped tar archive. The latest code for all pipelines, scripts, is available online. See Appendix ?? and ??