Supplementary material for: Identifying differentially expressed transcripts from RNA-seq data with biological variation

Peter Glaus[1], Antti Honkela[2] and Magnus Rattray[3]

[1]School of Computer Science, University of Manchester, UK
[2]Helsinki Institute for Information Technology HIIT,
Department of Computer Science, University of Helsinki, Finland
[3]Department of Computer Science and Sheffield Institute of Translational Neuroscience,
The University of Sheffield, UK

1 Methods

1.1 Alignment probabilities

We present the alignment probability computation for the case of paired-end reads. For single reads, the terms related to fragment or insert length distribution and the other paired read disappear.

For a given transcript $I_n = m; m \in \{1, \dots, M\}$, the probability of observing a pair of reads $(r_n^{(1)}, r_n^{(2)})$ is determined by the probability of the read being sequenced from a specific strand s at a specific position p with a specific insert length l and the probability of reporting the reads after sequencing the sequences $(seq_{mlps}^{(1)}, seq_{mlps}^{(2)})$,

$$P(r_n^{(1)}, r_n^{(2)} | I_n = m) = P(l|m)P(p|l,m)P(s|m)P(r_n^{(1)}|seq_{mlps})P(r_n^{(2)}|seq_{mlps}) \ . \tag{1}$$

Unless a strand specific sequencing protocol is used, the probability of observing a read from either strand is the same, $P(s|m) = 1/2$, and can be ignored. The fragment length distribution $P(l|m)$ is assumed to be log-normal with its parameters given by the user or estimated from read pairs with only a single transcript alignment.

The probability of sequencing a given position is in general

$$P(p|I_n = m, l) = \frac{b_m(p)}{\sum_{p=1}^{l_m - l_r + 1} b_m(p)}. \tag{2}$$

where $b_m(p)$ denotes bias for a particular position p on transcript m. For a constant $b_m(p)$ corresponding to a uniform read distribution, this reduces to $P(p|m) = 1/(l_m - l_r + 1)$ which only depends on the lengths of the transcript l_m and the read l_r.

We calculate the probability of observing a sequence based on the read's quality base scores and mismatches.

The *Phred* score can be converted into probability of base-calling error $p_{\text{err},i}$. The final sequence probability is now obtained as

$$P(r_n^{(j)}|seq_{mps}) = \prod_{i \in \text{matches}} (1 - p_{\text{err},i}) \prod_{i \in \text{mismatches}} p_{\text{err},i}, \tag{3}$$

where the probability of error for a given base i is based on the Phred score $p_{\text{err},i} = 10^{-\text{Phred}_i/10}$.

1.1.1 Bias estimation

Our model can easily incorporate a correction for position and sequence specific biases. One example of such a model is presented by Roberts *et al.* (2011) for correcting the fragmentation bias. Under this model, we have

$$b_m(p) = b_m^{s,5}(e_5) b_m^{s,3}(e_3) b_m^{p,5}(e_5) b_m^{p,3}(e_3), \tag{4}$$

where $b_m^{s,5}(e_5)$ and $b_m^{s,3}(e_3)$ are the sequence specific biases for 5' and 3' ends of the fragment, respectively, and $b_m^{p,5}(e_5)$ and $b_m^{p,3}(e_3)$ are the corresponding positional biases.

We use separate variable length Markov models to capture the bias for each end. The structure of this model is the same as that of Roberts *et al.* (2011), presented in Figure 2 of the supplementary methods. For the sequence bias these are

$$b_m^{s,5}(e_5) = \prod_{n=1}^{21} \frac{\psi_{n,\pi_n}^{5,R}}{\psi_{n,\pi_n}^{5,U}}, \tag{5}$$

which are based on 21 probabilities ψ_{n,π_n}^{5} from 8 bases before and 12 bases after the read starting position. Here $\psi^{5,R}$ refers to the biased and $\phi^{5,U}$ to a uniform model, n is a node or a position, π_n are the parents of node n and ψ_{n,π_n}^{5} is the probability of base X at node (or position) n given the bases observed on parent nodes π_n. The model has 744 parameters in all, with each node having 0, 1 or 2 parents as in the model of Roberts *et al.* (2011). The parameters are estimated from empirical frequencies using reads with a single alignment. For a read r aligning to transcript m we increase appropriate probabilities $\psi^{5,R}$ by $1/\theta_m$, where θ_m is an initial coarse expression estimate obtained by running BitSeq with uniform read distribution model beforehand. In the contrasting uniform model for all $K = l_m - l_r + 1$ possible positions of read of length l_r, the appropriate probabilities $\psi^{5,U}$ are increased by $\frac{1}{\theta_m K}$. The model $b_m^{s,3}(e_3)$ is similar.

In addition to the sequence-specific bias, there is a model for positional bias within the transcript. This is

$$b_m^{p,5}(e_5) = \frac{\omega_{l_m, e_5/l_m}^{R}}{\omega_{l_m, e_5/l_m}^{U}}, \tag{6}$$

where $\omega_{l,p}$ is the probability for starting position within transcript of length l on position p. The probabilities are modelled within 5 transcript length bins and 20 bins of relative position. The probabilities are again estimated from empirical frequencies of reads with single alignments taking into account expression θ.

1.2 Effective length computation

For the purpose of reporting normalized measure such as RPKM, $\boldsymbol{\theta}$, the relative expression of fragments, has to be normalized by the amount of reads or fragments that can be produced by a unit of transcript. When assuming uniform read distribution of single-end reads, this would be $l_m - l_r + 1$ as the number of starting positions for a read of length l_r. For paired-end reads, the effective length of a transcript has to account for fragment length distribution as well,

$$l_m^{(eff)} = \sum_{l_f=1}^{l_m} p(l_f|m) * (l_m - l_f + 1). \tag{7}$$

With the use of read distribution with bias correction, we learn more about the distribution of fragments and thus can use this information when computing the effective length. In this case, the effective length takes into account bias weight for every position of the transcript,

$$l_m^{(eff+bias)} = \sum_{l_f=1}^{l_m} p(l_f|m) \sum_{p=1}^{l_m - l_f + 1} b_m(p) \tag{8}$$

As we show later in Section 2.2 of this Supplementary material, using the bias corrected effective length can substantially improve the accuracy of our method.

1.3 Gibbs sampling in expression estimation (Stage 1)

We apply a collapsed Gibbs sampler for Stage 1 estimation by marginalising out the expression level and noise level parameters $\boldsymbol{\theta}$ and θ^{act} and iteratively resampling the isoform assignments I_n of each read given the assignments of other reads $I^{(-n)}$. The full update rules for the sampler are

$$\begin{aligned}
P(I_n|I^{(-n)}, R) &= \mathrm{Cat}(I_n|\boldsymbol{\phi}^{*}_{\boldsymbol{n}}), \qquad (9)\\
\phi^*_{n0} &= P(r_n|\text{noise})(\beta^{act} + C_0^{(-n)})/Z_n^{(\phi^*)},\\
m \neq 0; \phi^*_{nm} &= P(r_n|I_n)(\alpha^{act} + C_+^{(-n)}) \tfrac{(\alpha^{dir}+C_m^{(-n)})}{(M\alpha^{dir}+C_+^{(-n)})} / Z_n^{(\phi^*)},\\
C_m^{(-n)} &= \textstyle\sum_{i\neq n} \delta(I_i = m),\\
C_+^{(-n)} &= \textstyle\sum_{i\neq n} \delta(I_i > 0)\ ,
\end{aligned}$$

with $Z_n^{(\phi^*)}$ being a constant normalising $\boldsymbol{\phi_n}^*$ to sum up to 1, and $\alpha^{dir} = 1, \alpha^{act} = 2, \beta^{act} = 2$.

As an alternative, it is also possible to use a regular Gibbs sampler alternating between sampling I_n and $\boldsymbol{\theta}$. The corresponding update rules are

$$\begin{aligned}
P(I_n|\boldsymbol{\theta}, \theta^{act}, R) &= \mathrm{Cat}(I_n|\boldsymbol{\phi_n}), \qquad (10)\\
\phi_{n0} &= P(r_n|\text{noise})(1 - \theta^{act})/Z_n^{(\phi)},\\
m \neq 0; \phi_{nm} &= P(r_n|I_n)\theta_m\theta^{act}/Z_n^{(\phi)},\\
P(\boldsymbol{\theta}|\boldsymbol{I}, \theta^{act}, R) &= \mathrm{Dir}(\boldsymbol{\theta}|(\alpha^{dir} + C_1, \ldots, \alpha^{dir} + C_M)), \qquad (11)\\
P(\theta^{act}|\boldsymbol{I}, \boldsymbol{\theta}, R) &= \mathrm{Beta}(\theta^{act}|\alpha^{act} + N - C_0, \beta^{act} + C_0), \qquad (12)\\
C_m &= \textstyle\sum_{n=1}^{N} \delta(I_n = m).
\end{aligned}$$

This approach is usually less efficient in practice, though.

1.4 Differential Expression model (stage 2)

The Differential Expression (DE) model is shown in Figure 3 of the main paper. We consider data from conditions $c = 1 \ldots C$ with number of replicates for each condition denoted $R_1, \ldots, R_C$. We fit the model to each transcript m independently using "pseudo-data" $y_m^{(cr)} = \log \theta_m^{(cr)}$ which is created from MCMC samples from Stage 1. One sample of $\theta_m^{(cr)}$ is drawn for each (r, c) combination to create a pseudo-data vector $\boldsymbol{y}_m$ of length $\sum_{c=1}^{C} R_c$. Inference is carried out independently for each pseudo-data vector and the results are then combined as described in the main text. This allows the technical error from Stage 1 to be propagated through the model. Since the model is conjugate then the inference for each pseudo-data vector is exactly tractable and no further MCMC is required to sample the condition means.

1.4.1 Parameter estimation for each transcript

The condition means are denoted $\boldsymbol{\mu_m} = (\mu_m^{(1)}, \ldots, \mu_m^{(C)})$ and we are interested in inferring the posterior distribution over the means given one pseudo-data vector $\boldsymbol{y_m}$. The model is defined as,

$$\begin{aligned}
y_m^{(cr)} &\sim \mathrm{Norm}(\mu_m^{(c)}, 1/\lambda_m^{(c)})\\
\mu_m^{(c)} &\sim \mathrm{Norm}(\mu_m^{(0)}, 1/(\lambda_0\lambda_m^{(c)}))\\
\lambda_m^{(c)} &\sim \mathrm{Gamma}(\alpha_G, \beta_G)
\end{aligned}$$

with hyper-parameters $\lambda_0, \alpha_G, \beta_G$ which are estimated from groups of transcripts with similar mean expression across conditions. The hyper-parameter $\mu_m^{(0)}$ is fixed at the empirical mean transcript expression across conditions.

$$\begin{aligned}
p(\boldsymbol{\mu}_m, \boldsymbol{\lambda}_m|\boldsymbol{y}_m) &\propto p(\boldsymbol{y}_m|\boldsymbol{\mu}_m, \boldsymbol{\lambda}_m)p(\boldsymbol{\mu}_m)p(\boldsymbol{y}_m)\\
&\propto \prod_{c=1}^{C} p(\mu_m^{(c)})p(\lambda_m^{(c)}) \prod_{r=1}^{R_c} p(y_m^{(cr)}|\mu_m^{(c)}, \lambda_m^{(c)})\\
&\propto \prod_{c=1}^{C} \mathrm{Gamma}(\lambda_m^{(c)}|a_c, b_c)\mathrm{Norm}\left(\mu_c \,\middle|\, \frac{\lambda_0\mu_m^{(0)} + Syc}{\lambda_0 + R_c}, \frac{1}{\lambda_m^{(c)}(\lambda_0 + R_c)}\right)\\
a_c &= \alpha_G + \frac{R_c}{2}\\
b_c &= \beta_G + \frac{1}{2}\left(\lambda_0{\mu_m^{(0)}}^2 + S^2yc - \frac{(\lambda_0\mu_m^{(0)} + Syc)^2}{\lambda_0 + R_c}\right)
\end{aligned}$$

where Syc denotes $\sum_{r=1}^{R_c} y_m^{(cr)}$ and S^2yc denotes $\sum_{r=1}^{R_c} {y_m^{(cr)}}^2$.

1.4.2 Hyper-parameter estimation across transcript groups

For hyper-parameter estimation we consider a set of transcripts $m = 1 \ldots M'$ in a group g of transcripts with similar expression. We set $\lambda_0 = 2.0$ and split transcripts into 200 groups of similar expression based on μ_0, the mean transcript expression over all replicates. We have pseudo-data samples $y_m^{(cr)}$ for each transcript and we are interested in hyperparameters α and β, where β is the rate of Gamma distribution. The model is defined as,

$$\begin{aligned} y_m^{(cr)} &\sim \text{Norm}(\mu_m^{(c)}, 1/\lambda_m^{(c)}) \\ \mu_m^{(c)} &\sim \text{Norm}(\mu_m^{(0)}, 1/(\lambda_m^{(c)}\lambda_0)) \\ \lambda_m^{(c)} &\sim \text{Gamma}(\alpha, \beta) \\ P(\alpha,\beta) &\sim \text{Uniform}(0,\infty) \end{aligned}$$

The hyper-parameter posterior distribution is given by,

$$\begin{aligned} P(\alpha,\beta|\boldsymbol{y}) &\propto P(\alpha,\beta)P(\boldsymbol{y}|\alpha,\beta) \\ &\propto \prod_{m=1}^{M'}\prod_{c=1}^{C} P(\boldsymbol{y}_m^c|\alpha,\beta) \\ &\propto \prod_{m=1}^{M'}\prod_{c=1}^{C} \int d\lambda_m^{(c)} p(\lambda_m^{(c)}|\alpha,\beta) \int d\mu_m^{(c)} P(\mu_m^{(c)}|\lambda_m^{(c)}) \prod_{r=1}^{R_c} P(y_m^{(cr)}|\lambda_m^{(c)},\mu_m^{(c)}) \\ &\propto \prod_{m=1}^{M'}\prod_{c=1}^{C} \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\Gamma(\alpha+R_c)}{\left(\beta + \frac{1}{2}\left(\lambda_0 {\mu_m^{(0)}}^2 + S^2yc - \frac{(\lambda_0\mu_m^{(0)}+Syc)^2}{\lambda_0+R_c}\right)\right)^{\alpha+R_c}} . \end{aligned}$$

This distribution is not in a standard form and we use Metropolis-Hastings Random walk MCMC to sample α and β. We then use lowess smoothing across groups to estimate the mean hyper-parameter for each transcript according to its empirical mean expression level across conditions.

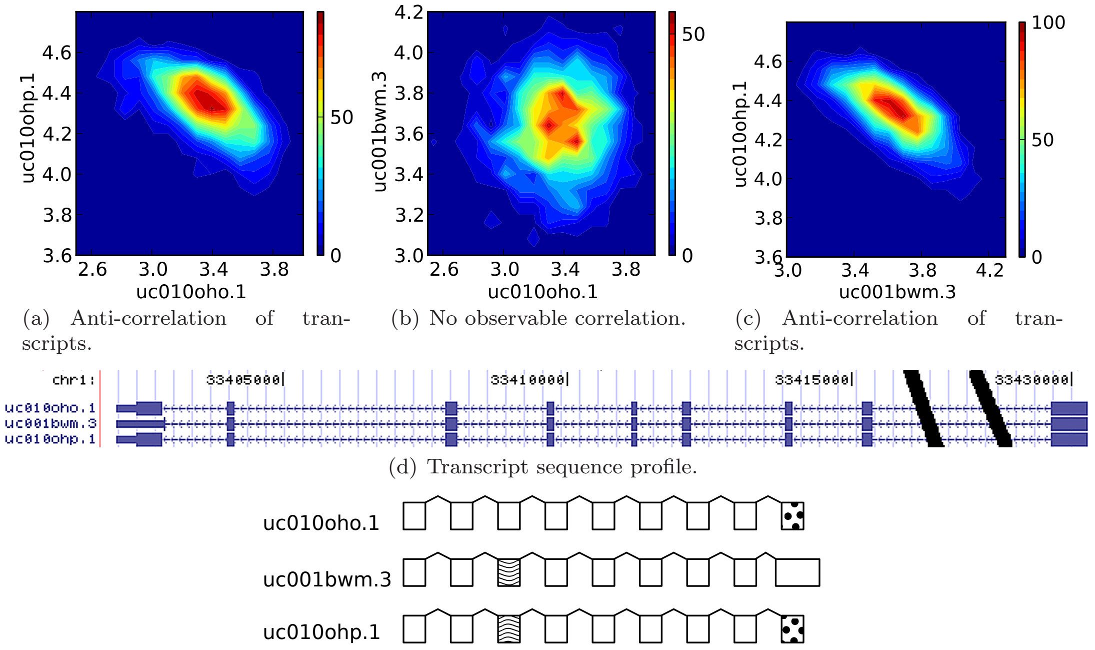
2 Results

2.1 Transcript expression inference

In the main text (Figure 4) we illustrate the correlation present in the expression posterior distribution for transcripts that share a large proportion of transcribed sequence. Here we provide all three pairwise plots for the transcripts uc010oho.1, uc010ohp.1 and uc001bwm.3, which are the only transcripts of gene Q6ZMZ0 in the UCSC Known Genes annotation. The expression samples of uc001bwm.3 and uc010ohp.1 (Figure 1(c)) are also negatively correlated. This means that the model is not able to decide from which transcript some of the reads originated and the posterior distribution captures all viable assignments. The transcript sequence profile in Figure 1(d) clearly demonstrates the similarity of the transcripts that causes higher uncertainty when inferring the transcript expression levels.

2.2 Read distribution bias correction

We compared four different methods for expression estimation which include bias correction options for non-uniform read distribution. The extended results are presented in Table 1, where we report the R^2 correlation of 893 transcript expression estimates with the TaqMan qRT-PCR results. We used every method to analyse each of the three technical replicates separately and then used the average expression level for the comparison. As was already stated in the main paper, the newest stable version of Cufflinks does provide the lowest correlation. We resorted to using the version 0.9.3 which was used in the paper presenting the bias correction method adopted by BitSeq (Roberts *et al.*, 2011). The results for MMSEQ assuming non-uniform read distribution were produced using effective lengths computed by BitSeq bias correction algorithm.



(a) Anti-correlation of transcripts.

(b) No observable correlation.

(c) Anti-correlation of transcripts.

(d) Transcript sequence profile.

(e) Splice variant model.

Figure 1: In plots (a), (b) and (c) we show the posterior transcript expression density for pairs of transcripts from the same gene. This is a density map constructed using the MCMC expression samples for these three transcripts. In (d) we show transcript sequence profile obtained from the UCSC genome browser. (e) highlights the differences between individual splice variants. The sequencing data is from miRNA-155 study published by Xu *et al.* (2010).

Method ver.	Read distribution	Average	Rep. 1	Rep. 2	Rep. 3
BitSeq 0.4	uniform *	0.758	0.757	0.758	0.759
BitSeq 0.4	uniform †	0.767	0.767	0.766	0.767
BitSeq 0.4	bias corrected *	0.756	0.755	0.756	0.757
BitSeq 0.4	bias corrected †	0.765	0.764	0.764	0.765
BitSeq 0.4	bias corrected ‡	**0.801**	0.801	0.795	0.804
Cufflinks 0.9.3	uniform	0.750	0.747	0.751	0.751
Cufflinks 0.9.3	bias corrected	**0.805**	0.801	0.805	0.808
Cufflinks 1.3.0	uniform	0.533	0.513	0.533	0.547
Cufflinks 1.3.0	bias corrected	0.684	0.685	0.691	0.644
RSEM 1.1.14	uniform	0.763	0.762	0.762	0.764
RSEM 1.1.14	bias corrected	**0.763**	0.762	0.762	0.764
MMSEQ 0.9.18	uniform	0.761	0.760	0.760	0.762
MMSEQ 0.9.18	bias corrected	**0.799**	0.799	0.793	0.802

Table 1: Evaluation of transcript expression inference algorithms using the SRA012427 RNA-seq data and TaqMan qRT-PCR expression measures for 893 matching transcripts. Reported values are Pearson R^2 correlation coefficient of the 893 transcripts' expression estimates and qRT-PCR results, best correlation of a method using averaged expression is highlighted. For each method we present values for average expression taken from three replicates as well as for each technical replicate separately. BitSeq was used with three different versions of expression length normalisation: $*$ – using actual transcript length, $\dagger$ – using effective length accounting for fragment length distribution, $\ddagger$ – using effective length accounting for fragment length and read distribution bias.

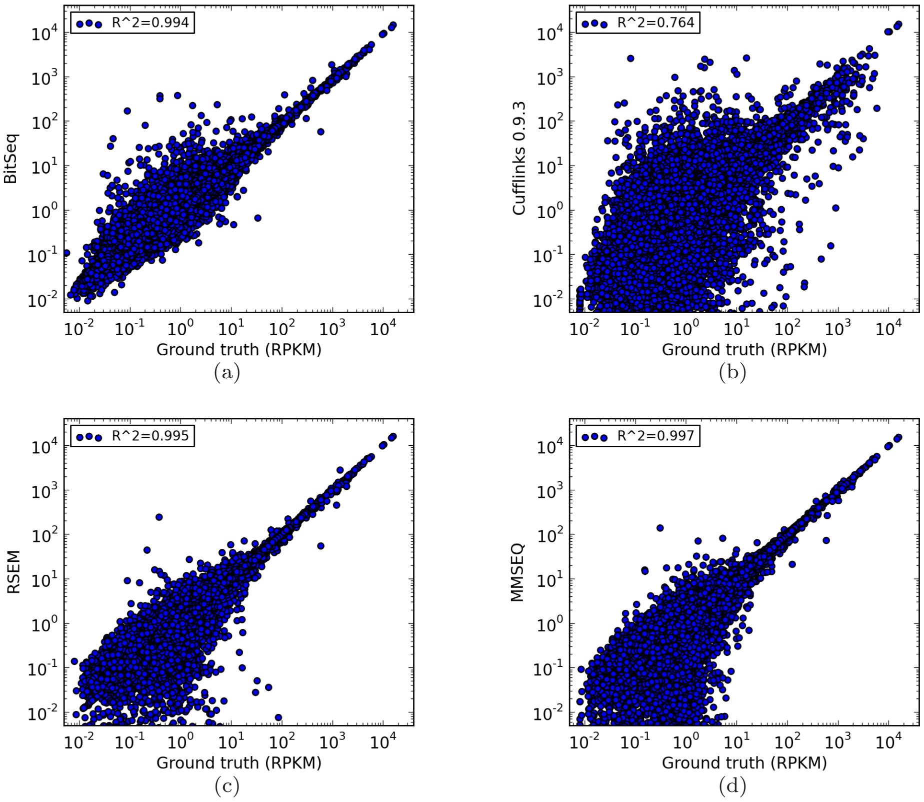


Figure 2: Comparison of expression estimates using 10M simulated paired-end reads with known expression. The expression estimates were converted into RPKM for each transcript and compared against ground truth using Log-Log plot. We calculated Pearson R^2 correlation coefficient for transcripts with at least one generated paired-end read.The figures show (a) BitSeq, (b) Cufflinks v0.9.3, (c) RSEM, and (d) MMSEQ.

For BitSeq, the major benefit of the bias correction algorithm comes from the effective transcript length normalisation. Relative expression of fragments used by BitSeq can be converted into relative expression of transcripts or into RPKM measure by adjusting the expression by effective length (see Supplementary Section 1). In Table 1 we compare three different approaches for length normalisation. In the first approach ($*$), the expression is adjusted by the length of a transcript. The second approach (†) uses effective length taking into account the paired-end read fragment length distribution and the number of all positions from which a fragment could originate. The best result for this dataset yields the last approach (‡), which uses effective length dependent on the fragment length distribution as well as read distribution bias weights (see Equation 8). More careful investigation of this process is required, however it is limited by small number of RNA-seq datasets with known underlying expression, especially when using paired-end reads.

2.3 Assessing transcript expression inference using simulated data

We used simulated dataset of 10M paired-end reads to examine the expression estimation accuracy of BitSeq and compared it against other three popular methods. The reads were generated with expression levels based on the estimates from the Xu et al. dataset with a fragment size distribution $l_f \sim \mathrm{LogNorm}(5.32, 0.12)$, inferred from the SRA012427 dataset. We used the UCSC NCBI37/hg19 knownGene annotation transcripts to generate the read fragments. First we compared the overall expression accuracy against the ground truth RPKM (Figure 2) computing the Pearson R^2 correlation coefficient. The coefficient was calculated for transcripts with at least one read generated (46841 transcripts). In this comparison MMSEQ ($R^2 = 0.997$) has the highest correlation with RSEM ($R^2 = 0.995$) and BitSeq ($R^2 = 0.994$) being closely behind. Unfortunately we again have to report poor results for the latest version of Cufflinks ($R^2 = 0.202$) with the version 0.9.3 still performing worse than the other three methods ($R^2 = 0.764$).

In the within-gene expression comparison (Table 2), we calculated the relative proportion of transcripts' RPKM within its gene, with two cutoffs for relevant transcripts. The first taking

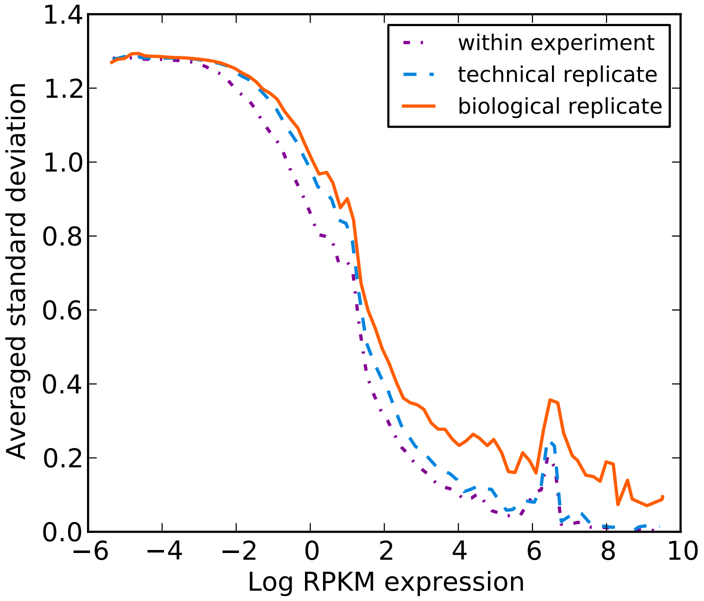


Figure 3: Comparison of standard deviation of posterior samples within single dataset and combined datasets of technical replicates and biological replicates, with log RPKM expression on the x-axis and standard deviation of log RPKM expression on y-axis. The standard deviation is a sliding average over groups of transcripts with similar expression in order to highlight its dependents on the expression.

into account transcripts for which their gene has at least 10 reads in the ground truth (45662 transcripts) and the second considering only transcripts for which the gene has at least 100 reads (33757 transcripts). BitSeq performs the best for the narrow range of transcripts with RSEM and MMSEQ having comparable results. For the less stringent criteria, BitSeq still retains very good correlation with the ground truth while the performance of the other two methods deteriorates. As we are using the same dataset, both versions of Cufflinks provide poor correlation when compared to other three methods.

	BitSeq	Cufflinks 1.3.0	Cufflinks 0.9.3	RSEM	MMSEQ
above 10 reads	**0.945**	0.214	0.724	0.876	0.886
above 100 reads	**0.963**	0.181	0.773	0.946	0.948

Table 2: The R^2 correlation coefficient of estimated within-gene relative expression and ground truth. The correlation was calculated for two groups, first one containing transcripts of genes with at least 10 reads and the second one containing transcripts of genes with at least 100 reads according to the ground truth.

Adding expression from all transcripts of a given gene yields an absolute gene expression measure, for which we compared genes with at least one read generated (15973 genes). All methods provide very good accuracy with MMSEQ producing the best correlation (see Table 3).

BitSeq	Cufflinks	Cufflinks 0.9.3	RSEM	MMSEQ
0.994	0.330	0.823	0.996	**0.998**

Table 3: The R^2 correlation coefficient of estimated absolute gene expression and ground truth. Genes with at least one generated read were considered.

2.4 Biological variance of RNA-seq data

We used RNA-seq data from the microRNA target identification study (Xu *et al.*, 2010) to test and compare the differential expression analysis method used in BitSeq Stage 2. This dataset contains technical as well as biological replicates for each studied condition allowing assessment of the effects of biological variation. Similarly to previous results (Anders and Huber, 2010; Oshlack *et al.*, 2010), we observe significant biological variation within conditions. Figure 3 shows the standard deviation of transcript expression level posterior MCMC samples as a function of the mean expression level of the transcript. We compare the standard deviation for samples from within one experiment, between two technical replicates and between two biological replicates. In order to calculate the standard deviation between replicates we took the square root of variance which was estimated by computing root mean square distance between samples. Plotted values are averaged for a sliding window of similarly expressed transcripts. The MCMC sample variation captures the intrinsic estimation variance in the "within-experiment" case. The technical variance includes a contribution due to re-sequencing the same biological sample while the biological variance includes a contribution due to repeating the experiment.

We see that with higher expression the variation of the expression level estimation decreases as expected. At high expression levels the variance associated with technical replicates approaches the level of the within-experiment variance. On the other hand, the biological variance becomes relatively more significant in this regime. Without consideration of biological differences, high confidence of expression estimation of these transcripts will lead to false differential expression calls. It can also be observed that the within-experiment variance is a significant contribution to replicate variance (technical and biological) at lower expression levels. Therefore the intrinsic variance due to mapping ambiguity and limited read depth, as estimated by our MCMC expression estimation procedure, will provide useful information for assessing replicate variance in this low expression regime.

2.5 Assessing DE performance with simulated data

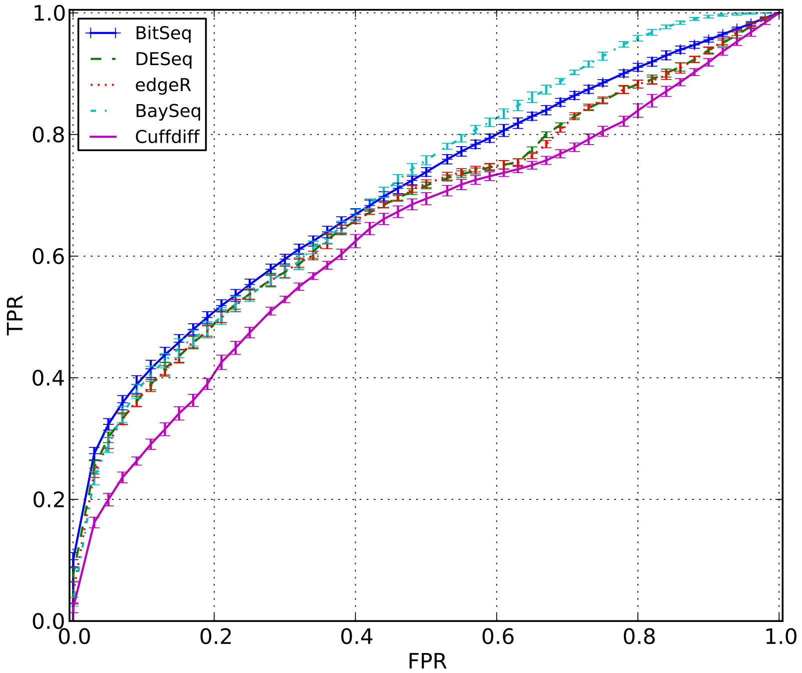
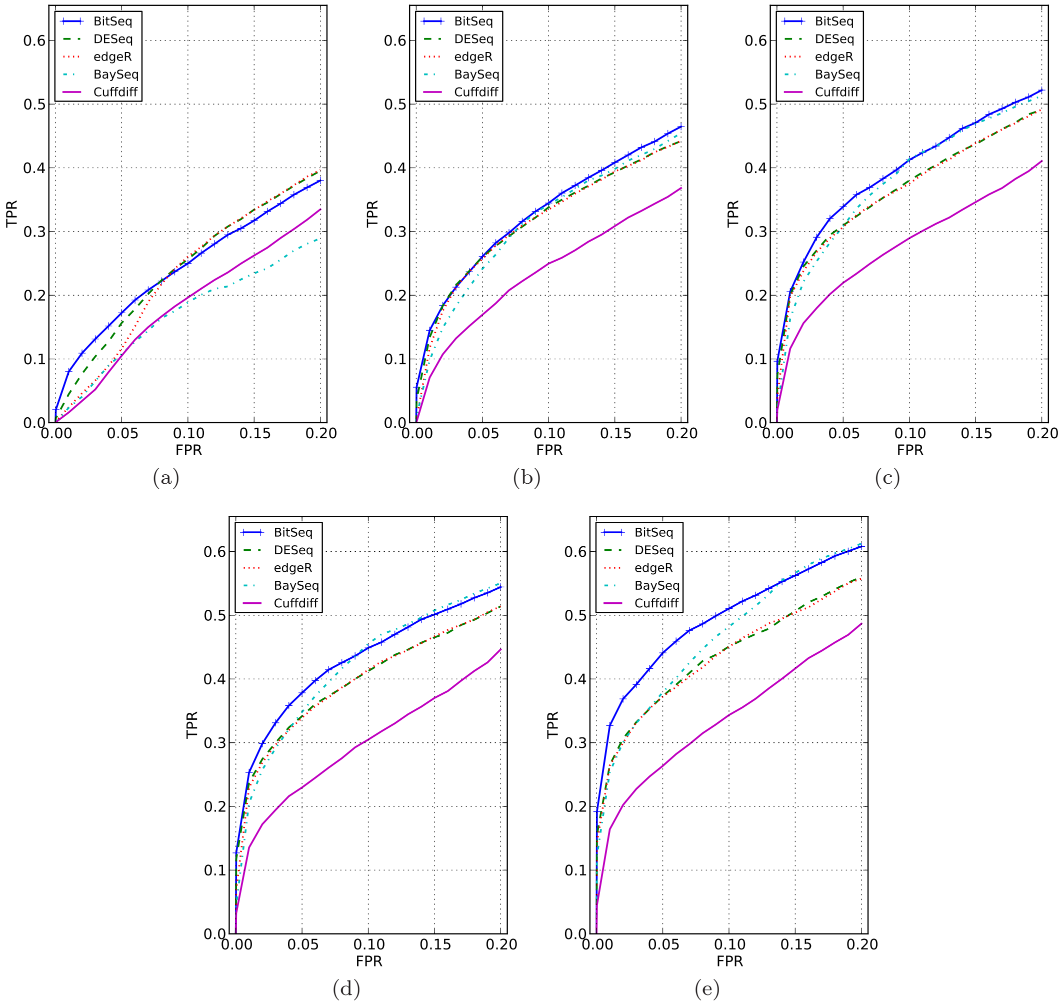


Figure 4: ROC curves averaged over 5 runs with standard deviation depicted by error bars. The curve was calculated for transcripts with average of at least one read in the ground truth. The fold change was uniformly distributed in the interval $(1.5, 3.5)$.

We carried out extensive assessment of DE analysis accuracy of BitSeq with comparison to other methods. The Cuffdiff method from Cufflinks package (Trapnell *et al.*, 2010) is the only other method designed for transcript level DE analysis that uses replicates and accounts for biological variation. We also included three popular methods which are primarily designed for gene level DE analysis (DESeq (Anders and Huber, 2010), edgeR (Robinson *et al.*, 2010), baySeq (Hardcastle and Kelly, 2010)), but given the lack of other options and their input being only the read count vectors, they could be considered for the transcript level analysis use case as well. Using expression estimates obtained by BitSeq Stage 1, we converted the relative expression of



(a)

(b)

(c)

(d)

(e)

Figure 5: Differential expression analysis of simulated data with various levels of fold change. The figures focus on the most relevant region with false positive rate above 0.2, and showing the y-axis up to true positive rate 0.65. The sub figures show data with fold change: 1.5 (a), 2.0 (b), 2.5 (c), 3.0 (d) and 5.0 (e).

fragments into read counts by simply multiplying it by the total number of aligned reads and used this as an input for the gene-level methods. For each of these methods we used default parameter settings according to the packages' vignettes.

The Figure 4 shows the same ROCs as Figure 6(a) in the main paper without the 0.2 cutoff. The evaluation is only for transcripts with at least one generated read on average with fold change being uniformly generated from the interval $(1.5, 3.5)$. In this figure, the error bars depict the standard deviation for the averaged curves showing consistent results through the experiments. We can see that BitSeq performs slightly better than the other methods with baySeq having higher true positive range in area with above 0.4 false positive range, however this area is not interesting from the application perspective.

In the very last figure (5), we compare the accuracy of these methods with respect to the fold change of differentially expressed transcripts. We again restrict the figures to the area with false positive rate below 0.2 which in our opinion is the most important in terms of applicability. Instead of using randomly selected fold change, all differentially expressed transcripts are either up-regulated or down-regulated by constant fold change. The increase of fold change clearly improves the performance of the methods as we expected. BitSeq and baySeq have consistently better results than the other methods except for the lowest fold change 1.5, in which baySeq has the lowest true positive rate and edgeR with DESeq outperform BitSeq in half of the spectrum.

In all of our DE experiments, Cuffdiff, despite being designed for transcript level analysis performs worse out of the 5 compared algorithm. This could be largely attributed to the expression estimation problem, however for DE analysis return to the older version (0.9.3) did not improve the results, possibly because of different DE model. Our data also shows that for most parts, the DESeq and edgeR methods produce very similar results in terms of accuracy. We have to note, that even though we tried to simulate the data in way to resemble real RNA-seq experiments, the data proved to be rather hard for all methods being compared.

References

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biol*, **11**(10), R106.

Hardcastle, T. J. and Kelly, K. A. (2010). baySeq: Empirical Bayesian Methods For Identifying Differential Expression In Sequence Count Data. *BMC Bioinformatics*, **11**(1), 422.

Oshlack, A., Robinson, M. D., and Young, M. D. (2010). From RNA-seq reads to differential expression results. *Genome Biol*, **11**(12), 220.

Roberts, A., Trapnell, C., Donaghey, J., Rinn, J. L., and Pachter, L. (2011). Improving RNA-Seq expression estimates by correcting for fragment bias. *Genome Biol*, **12**(3), R22.

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*, **26**(1), 139–40.

Trapnell, C. *et al.* (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat Biotechnol*, **28**(5), 516–520.

Xu, G. *et al.* (2010). Transcriptome and targetome analysis in MIR155 expressing cells using RNA-seq. *RNA*, pages 1610–1622.