Stat Biosci
DOI 10.1007/s12561-012-9057-6

Bias Correction in RNA-Seq Short-Read Counts Using Penalized Regression

David Dalpiaz · Xuming He · Ping Ma

Received: 21 November 2011 / Accepted: 2 February 2012

Abstract RNA-Seq produces tens of millions of short reads. When mapped to the genome and/or to the reference transcripts, RNA-Seq data can be summarized by a very large number of short-read counts. Accurate transcript quantification, such as gene expression calculation, relies on proper correction of sequence bias in the RNA-Seq short-read counts. We use a linear model for the sequence bias, which is much more flexible than the popular Poisson model. We fit the model using a penalized regression method, which allows for a significant dimension reduction. The algorithm is scalable for modeling RNA-Seq data. We demonstrate the excellent performance of our proposed method by applying it to real examples. The methods are implemented in open-source code, which is available in the R package `lmbc`.

Keywords RNA-Seq · Next-generation sequencing · Gene expression · LASSO · Regularization · Penalized likelihood

1 Introduction

With the rapid development of second-generation sequencing technologies, RNA-Seq has become a popular tool for transcriptome analysis [8, 9, 14]. It produces digital signals and offers the chance to detect novel transcripts by obtaining tens of millions

D. Dalpiaz · P. Ma (✉)
Department of Statistics, University of Illinois at Urbana-Champaign, Champaign, IL, USA
e-mail: pingma@illinois.edu

X. He
Department of Statistics, University of Michigan, Ann Arbor, MI, USA

P. Ma
Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana, IL, USA

Published online: 22 February 2012

of short reads. When mapped to the genome and/or to the reference transcripts, RNA-Seq data can be summarized by a tremendous number of short-read counts. The huge number of short-read counts enables researchers to make transcript quantification in ultra-high resolution.

A number of researchers have worked on transcript quantification, in particular, the gene expression calculation, using the short-read counts. Mortazavi et al. [8] develop a simple method, in which the expression level of a transcript is quantified as reads per kilobase of the transcript per million mapped reads to the transcriptome (RPKM). A variant, FPKM is developed in [11]. These analysis methods assume, explicitly or implicitly, a naive constant-rate Poisson model, which often fits the data poorly. Recent work found that short-read counts have significant sequence bias, e.g., GC-rich regions tend to have larger read counts than AT-rich regions, see [3], which makes simple transcript quantification methods like RPKM questionable. Thus, more elaborate statistical models that can effectively remove the sequence bias of the short-read counts are highly desirable to make transcript quantification more accurate. Li et al. [7] and Bullard et al. [1] developed Poisson regression models with variable rates for modeling the short-read counts. However, the short-read counts data are observed to be overdispersed, which renders the Poisson model inadequate. Moreover, Poisson model-fitting using the iterative re-weighted least squares is computationally expensive with large data. Because of the inadequate fit of the Poisson model, Li et al. [7] also attempted a regression tree model, MART [4, 5], which provides a much better fit. However, the price paid is that as an algorithmic approach, the MART model does not enjoy the nice interpretation of the Poisson model and it is hard to make statistical inference based on the method.

To surmount these challenges, in this paper, we develop a model-based bias correct approach, in which we model linearly the sequence bias of logarithm-transformed read counts as a function of the surrounding dinucleotide configurations. The linear model enjoys the easy interpretation and many readily available inference tools. We fit the model using a distance-weighted penalized regression method, which enables effective dimension reduction. The LARS algorithm is employed for model-fitting, which provides efficient and fast computation. We demonstrate the excellent performance of our proposed method by applying it to real examples. The methods are implemented in open-source code, which is available in the R package `lmbc`.

2 Methods

2.1 Dinucleotide Linear Model

Let n_{ij} denote the counts of reads that are mapped to the genome starting at the jth nucleotide of the ith gene, where $i = 1, 2, \ldots, G$, $j = 1, \ldots, L_i$. As observed in [7], the read counts in each nucleotide in the same gene are highly heterogeneous, and highly correlated across tissues, see Fig. 1 of [7].

Figure 1 of [7] also suggests that the read counts in a nucleotide might have bias associated with its genomic position, which can be determined by the neighborhood nucleotide composition. Thus Li et al. [7] considered associating the read

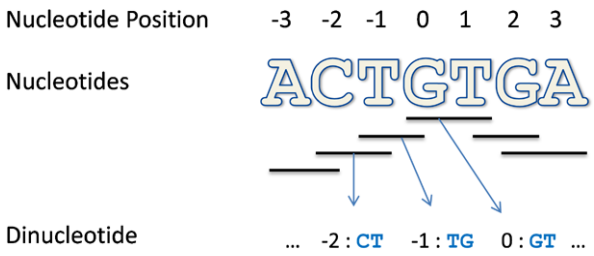

Fig. 1 An illustration of the neighborhood overlapping dinucleotide composition. A read is mapped to the genome starting at position 0. Upstream positions are labeled as negative and downstream positions are labeled as positive

counts with neighborhood single nucleotide composition by additive models. In this paper, we develop a linear model relating the short-read counts in a nucleotide with its neighborhood overlapping dinucleotide compositions, through which the nucleotide interactions are naturally built in. We assume that the log transformed count of reads, $y_{ij} = \log(n_{ij} + 1)$, depends on K_u nucleotide pairs immediately upstream and K_d nucleotide pairs immediately downstream the read, denoted as $b_{ij,-K_u}, b_{ij,-(K_u-1)}, \ldots, b_{ij,(K_d-1)}, b_{ij,K_d}$, see Fig. 1, through the following linear model:

$$y_{ij} = \alpha + v_i + \sum_{k=-K_u}^{K_d} \sum_{h \in \mathcal{H}} \beta_{kh} I(b_{ijk} = h) + \epsilon_{ij}, \tag{2.1}$$

where $\mathcal{H} = \{CC, GG, TT, AC, CA, AG, GA, AT, TA, CG, GC, CT, TC, GT, TG\}$ (AA was used as baseline), α is the grand mean, v_i is the main effect of gene i, under the constraint $\sum v_i = 0$, $I(b_{ij,k} = h)$ equals 1 if the kth dinucleotide of the surrounding sequence is h, and 0 otherwise, β_{kh} is the coefficient of the effect of dinucleotide h occurring in the kth position, and $\epsilon_{ij} \sim N(0, \sigma^2)$. Let $K = K_u + K_d$. The constant 1 is added to the original counts to account for positions with zero reads mapped. This linear model uses $15K + G$ parameters to model the sequence bias of read counts. It is worth noting that the trinucleotide composition may be considered in the model, but the large number of parameters, i.e., 63 parameters for each trinucleotide position, incurs a rapid surge of the computation costs for model-fitting.

2.2 Model-Fitting and the Distance-Weighted Penalized Regression

In practice, the number of upstream nucleotides K_u and K_d in model (2.1) need to be specified. One way is to assign sufficiently large numbers to K_u and K_d so that the related dinucleotides are all included in the model. However, if we set $K_u = 40$ and $K_d = 40$ (thus $K = 80$), we will have roughly 1200 dinucleotide coefficients β_{kh} to estimate. With such a huge number of coefficients, many of which are redundant, the calculations are unstable and error-prone. To alleviate the computational cost and to stabilize the algorithm, we use a penalized regression method to determine the number of nucleotides adaptively. Since the number of overlapping dinucleotides K corresponds to K single nucleotides, our penalized regression directly searches for an optimal number of single nucleotides. Among penalized regression methods, the L_1 penalized likelihood procedure is very effective since the L_1 penalty encourages shrinkage of irrelevant predictors to be exactly zero. The standard L_1 penalty uses the

same weights for different predictors. However, the predictors in our model are dinucleotides, and it is observed in [7] that the impact of the nucleotides on the modeling read counts becomes smaller as the nucleotides get further away from the mapped reads. We thus consider a distance-weighted L_1 penalty [16] in our algorithm so different nucleotides are penalized according to their relative distance to the mapped reads.

Algorithm

(1) We first fit a single nucleotide model with distance-weighted penalty,

$$\sum_{i=1}^{G}\sum_{j=1}^{L_i}\left\{y_{ij}-\alpha-v_i-\sum_{k=-K_u}^{K_d}\sum_{h\in\mathcal{H}^*}\beta^*_{kh}I\big(b^*_{ijk}=h\big)\right\}^2+\lambda\sum_{k=-K_u}^{K_d}\sum_{h\in\mathcal{H}^*}w_k\big|\beta^*_{kh}\big|, \tag{2.2}$$

where b^*_{ijk} is the nucleotide composition of the kth nucleotide away from the jth nucleotide in ith gene, λ is the tuning parameter and $\mathcal{H}^*=\{C,G,T\}$ (A was used as baseline). The weight $w_k>0$ will be chosen to be proportional to a certain power of distance between nucleotide j and the kth nucleotide in the surrounding sequence [16, 17], i.e., $w_k=(|k|+1)^\gamma$. We use the LARS/Lasso algorithm (a.k.a. the homotopy algorithm) to find the solutions for all values of λ. Even though the solution path for all values of λ can be effectively computed, it is still highly desirable that one solution is given for a carefully chosen value of λ. To choose a value of λ with a good balance of goodness-of-fit of the model and model parsimony, we minimize the Bayesian Information Criterion (BIC).

(2) Based on the parameters of the penalized fit in step 1, we then select the new sequence endpoints $K^*_u=\min\{k:\beta^*_{kh}\neq 0\ \forall h\in\{C,G,T\}\}$, and $K^*_d=\max\{k:\beta^*_{kh}\neq 0\ \forall h\in\{C,G,T\}\}$.

(3) With the selected K^*_u and K^*_d and the dinucleotide expansion, we fit the model (2.1) using the least squares,

$$\sum_{i=1}^{G}\sum_{j=1}^{L_i}\left\{y_{ij}-\alpha-v_i-\sum_{k=K^*_u}^{K^*_d}\sum_{h\in\mathcal{H}}\beta_{kh}I\left(b_{ijk}=h\right)\right\}^2, \tag{2.3}$$

where $\mathcal{H}=\{CC,GG,TT,AC,CA,AG,GA,AT,TA,CG,GC,CT,TC,GT,TG\}$.

This model will then be used to estimate gene expression levels.

3 Results and Discussion

3.1 Datasets

Our model was fitted to three genome-wide RNA-Seq datasets. These datasets will be referred to as Wold, Burge and Grimmond as in [7]. The Wold data, which come from [8], consist of 79, 76, and 70 million reads, which are of length 25, generated by

Illumina's Solexa. The 79, 76, and 70 million reads each correspond to a subdataset from brain tissue, liver tissue, and skeletal muscle tissue, respectively. Like Li et al. [7], when fitting the data, we will use the top 100 genes according to RPKM. So for the brain, liver and muscle Wold datasets, we are considering 146828, 171776 and 143570 nucleotides for each datasets' 100 genes, respectively. The Burge data, which come from [12], consist of three subdatasets which will be referred to as G1, G2 and G3. G1 consists of adipose, brain and breast tissue. G2 consists of colon, heart and liver tissue. G3 consists of lymph node, skeletal muscle and testes tissue. Each has reads ranging from 61 to 77 million. The datasets G1, G2, and G3 each consider 157614, 125056 and 103394 nucleotides, respectively. The Burge data were also generated from Illumina's Solexa with reads of length 32. Lastly, the Grimmond data, of [2], were generated from ABI's SOLiD with an original read length of 35. (Some are truncated into 30 or 25 nucleotides to ensure high quality.) The data consist of two subdatasets, each consisting of 16 million reads from each of two cell lines, which will be referred to as EB (embryoid bodies) and ES (undifferentiated mouse embryonic stem cells). The EB subdataset's top 100 genes considers 51751 nucleotides and the ES subdataset uses 64966 nucleotides. We use the read counts data for the top 100 genes as prepared by Li et al. [7].

3.2 Tuning the Algorithm

Our algorithm requires several tuning parameters. In this section, we present some results of various choices we attempted for the parameters. As an assessment of goodness-of-fit, we calculated the Bayesian Information Criterion (BIC),

$$\text{BIC} = -2\text{loglike} + (15K + G)\log\left(\sum_{i=1}^{G} L_i\right) \tag{3.1}$$

where the log-likelihood of the fitted model is

$$\text{loglike} = \sum_{i=1}^{G}\sum_{j=1}^{L_i}\left\{\frac{1}{2}\log\frac{1}{2\pi\hat{\sigma}^2} - \frac{(\log(n_{ij}+1) - \widehat{\log(n_{ij}+1)})^2}{2\hat{\sigma}^2}\right\} \tag{3.2}$$

where $\widehat{\log(n_{ij}+1)}$ is the fitted value of the model. $\hat{\sigma}^2$ is the estimated σ^2 using the residual sum of squares.

3.2.1 Determining Weights for Penalized Regression

In the penalized regression, γ, the power of the weights, $w_k = (|k|+1)^\gamma$ needs to be determined. By fixing $\gamma = 1, 2, \ldots, 10$ one-at-a-time, we calculated BIC based on the resulting surrounding sequence, which is presented in Table 1. By inspecting Table 1, we note that cubic and quartic weights led to the best results most frequently. For simplicity, we opt to use the cubic weight ($\gamma = 3$) as our final choice when determining a surrounding sequence as it frequently preforms very well.

Table 1 Bayesian Information Criterion (BIC) for linear model with various penalty weights, γ. BICs are scaled with respect to the sample size of the dataset

	γ	1	2	3	4	5	6	7	8	9	10
Wold	Brain	2.38	2.39	2.39	2.39	2.41	2.39	2.41	2.59	2.66	2.87
	Liver	2.57	2.56	2.55	2.56	2.58	2.56	2.58	2.74	2.81	3.00
	Muscle	2.76	2.74	2.74	2.74	2.76	2.74	2.76	2.88	2.93	3.10
Burge	G1	2.83	2.82	2.82	2.82	2.83	2.83	2.86	2.95	2.99	3.08
	G2	3.07	3.07	3.05	3.05	3.06	3.04	3.07	3.15	3.19	3.29
	G3	2.97	2.95	2.94	2.94	2.96	2.94	2.97	3.07	3.12	3.22
Grimmond	EB	3.52	3.52	3.52	3.52	3.55	3.52	3.55	3.59	3.65	3.77
	ES	3.31	3.26	3.26	3.26	3.29	3.27	3.31	3.35	3.40	3.49

Table 2 The resulting surrounding sequence lengths upstream and downstream of the reads

Dataset	Subdataset	Upstream	Downstream
Wold	Brain	13	14
	Liver	17	12
	Muscle	13	23
Burge	G1	21	20
	G2	22	32
	G3	25	31
Grimmond	EB	24	25
	ES	25	27

3.2.2 Determining Surrounding Sequence

In our algorithm, the distance-weighted penalized regression results in a sparse set of parameters β_{kh}. Since each nucleotide position has three associated β_{kh}, we need to translate the sparse set of parameters β_{kh} into a sparse set of surrounding nucleotides. In the second step of our algorithm, we select the K_u as the upmost k with all $\beta_{kh} \neq 0$ for $h \in \{C, G, T\}$, and K_d as the downmost k with all $\beta_{kh} \neq 0$ for $h \in \{C, G, T\}$. To examine the effectiveness of this strategy, we compare it with an alternative strategy which selects K_u as the upmost k with at least one of $\beta_{kh} \neq 0$ for $h \in \{C, G, T\}$, and K_d as the downmost k with at least one of $\beta_{kh} \neq 0$ for $h \in \{C, G, T\}$. This alternative strategy results in a larger K_u and K_d, however after refitting with the dinucleotide expansion, the goodness-of-fit does not improve enough to justify the increased number of parameters. This suggests that the strategy used in step 2 is appropriate. Table 2 presents the sequence lengths determined by our algorithm.

This data-driven method of selecting a surrounding sequence gives different results from that in [7]. We find shorter surrounding sequences are needed for the Wold datasets, but larger surrounding sequences for the Burge and Grimmond data.

Table 3 Log-likelihoods for linear models with single nucleotide expansion and the fit with the dinucleotide expansion. Both fit with surrounding sequences from Table 2

Dataset	Subdataset	Single nucleotide	Dinucleotide
Wold	Brain	1.28	1.172
	Liver	1.36	1.26
	Muscle	1.42	1.34
Burge	G1	1.44	1.38
	G2	1.54	1.48
	G3	1.50	1.42
Grimmond	EB	1.77	1.67
	ES	1.66	1.55

3.2.3 Dinucleotide Composition

We also compare the linear model using neighborhood single nucleotide composition with the linear model using our neighborhood overlapping dinucleotide compositions. Table 3 presents the (negative) log-likelihoods for the two models. We can see the (negative) log-likelihood of the linear model with dinucleotide composition improves upon that with single nucleotide composition. This improvement can also be seen through an increase in R^2 (data not shown). For example in the Wold Brain data, R^2 is increased by 25% from the model with single nucleotides to that with dinucleotide.

3.3 Comparison of Linear Models with the Existing Models

We now compare our linear model with the Poisson and MART models in [7]. As a direct comparison of goodness-of-fit, we consider the log-likelihoods of our linear model and the Poisson model. (The MART is an algorithmic method, the log-likelihood cannot be calculated.) Since the read counts are clearly overdispersed, we also fit a scaled (overdispersed) Poisson model as a fair comparison.

For the Poisson model, the log-likelihood is

$$\text{loglike} = \sum_{i=1}^{G} \sum_{j=1}^{L_i} \left\{ \frac{n_{ij}}{\sigma} \log\left(\frac{1}{\sigma} \frac{\hat{n}_{ij}}{\sigma} \right) - \frac{\hat{n}_{ij}}{\sigma} - \log \frac{\hat{n}_{ij}!}{\sigma} \right\} \tag{3.3}$$

where $\hat{n}_{ij}$ is the fitted value of the model. For the scaled Poisson model σ is the dispersion parameter estimated by a quasi-likelihood method [13]. For the unscaled Poisson model, σ is taken to be 1. The predicted counts using linear model and Poisson model for gene TNNc2 in Wold data are given in Fig. 2.

The resulting likelihood of the models for each dataset were summarized in Table 4. Even after adjusting for the dispersion, we see that the linear model outperforms the scaled Poisson model in terms of log-likelihoods.

In addition to goodness-of-fit, the computational costs of our linear and existing models are also compared. Table 5 summarize the total runtime for each method.

When recording the runtime of the linear model, we include the time to determine the surrounding sequence and the time to refit the model with the dinucleotide expansion. For the MART model, we use a predetermined surrounding sequence and

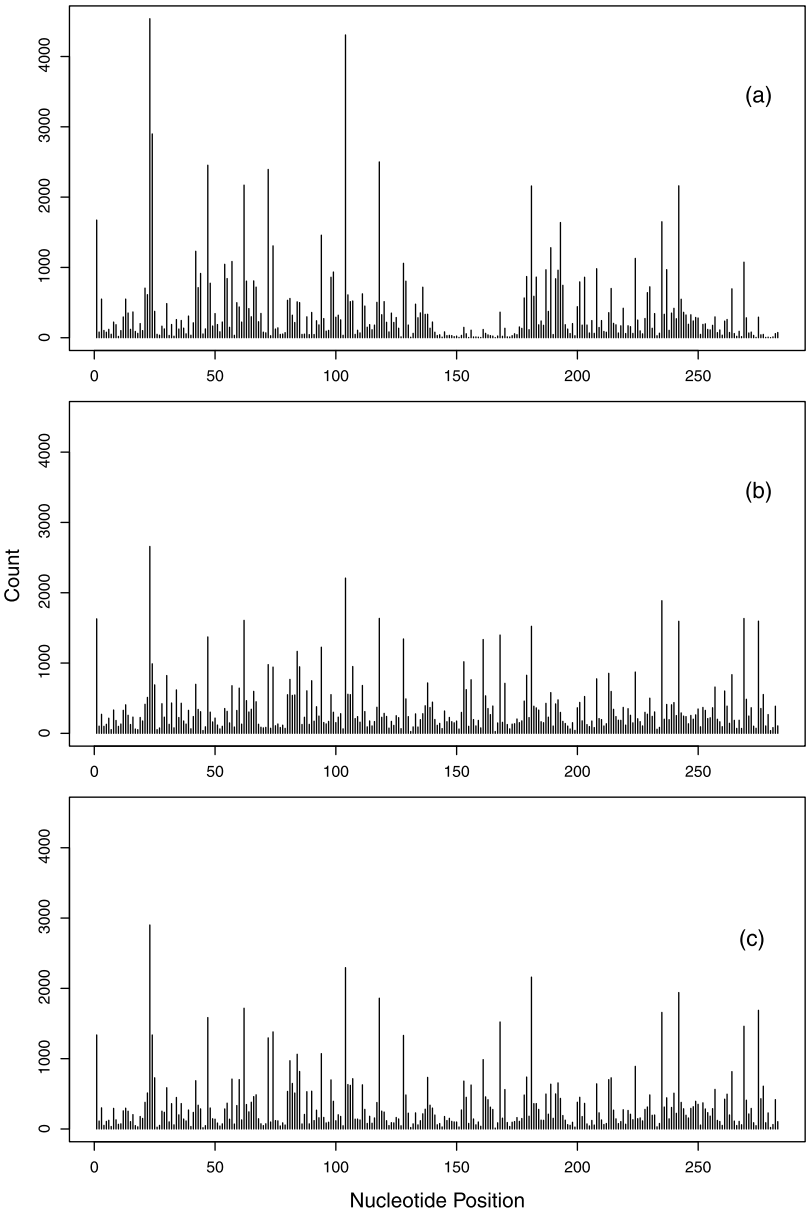

Fig. 2 Predicted counts for gene Tnnc2. (**a**) True read counts for the Tnnc2 gene of the Wold muscle data. (**b**) Predicted counts for the linear model for the Tnnc2 gene. (**c**) Predicted counts for the Poisson model for the Tnnc2 gene

use the default parameters for fitting as suggested by Li et al. [7]. From Table 5, we can clearly see that there is a significant run time difference between the Poisson and the linear models. This observation is most notable when fitting with the dinu-

Table 4 Log-likelihoods for Poisson and linear models. Likelihoods are scaled with respect to the sample size of the dataset. The Poisson models are fit with a surrounding sequence of 40. The surrounding sequences used for the linear model are the surrounding sequences found in Table 2

Dataset	Subdataset	Poisson	Scaled Poisson	Linear
Wold	Brain	6.54	3.69	1.17
	Liver	13.00	4.43	1.26
	Muscle	17.00	4.61	1.34
Burge	G1	9.32	3.97	1.38
	G2	16.8	4.64	1.48
	G3	15.6	4.54	1.42
Grimmond	EB	89.95	6.38	1.67
	ES	34.79	5.43	1.55

Table 5 Runtime for the linear, Poisson and MART models. CPU time (in seconds) for fitting the models obtained on a PC with an Intel Xeon E5540 processor and 24 Gbytes of RAM running openSUSE 11.4 and R 2.12.1. When fitting the linear model, the time used to determined the surrounding sequence is included

Dataset	Subdataset	Linear	Poisson	MART
Wold	Brain	313	777	2751
	Liver	394	900	3314
	Muscle	220	513	1733
Burge	G1	325	1478	1843
	G2	344	2317	527
	G3	176	648	1273
Grimmond	EB	90	312	618
	ES	125	594	768

cleotide expansion. As an example for the Wold brain dataset, the linear model (with sequence selection) runs roughly three times faster than the Poisson model, which uses an iterative re-weighted least squares algorithm.

3.4 Estimating Gene Expression Levels

Using our linear model, we have two methods to estimate gene expression levels. First, to estimate the gene expression for gene i, we can use $\hat{\alpha} + \hat{v}_i$ from the estimated model. As an alternative, we can also estimate the gene expression by bias-removed read counts $\sum_{j=1}^{L_i} n_{ij}/W_i$ where

$$W_i = \sum_{j=1}^{L_i} \exp\left(\hat{\alpha} + \sum_{k=1}^{K} \sum_{h \in \mathcal{H}} \hat{\beta}_{kh} I(b_{ijk} = h)\right) \tag{3.4}$$

which is the sum of the sequence bias across all the nucleotide positions of gene i.

Because there is no gold standard to validate the gene expression estimates, we opt to correlate our estimates with the estimates using MART model in [7]. We find that both methods are highly correlated with the results in [7] using non-linear MART model with the sum of sequencing preferences. Our second method, using the sequence bias, does slightly better than the first method based on the estimated $\hat{\alpha} + \hat{v}_i$.

Table 6 Spearman rank correlations between MART and linear model gene expression estimates

Dataset	Subdataset	Correlation
Wold	Brain	0.992
	Liver	0.994
	Muscle	0.993
Burge	G1	0.968
	G2	0.956
	G3	0.972
Grimmond	EB	0.974
	ES	0.980

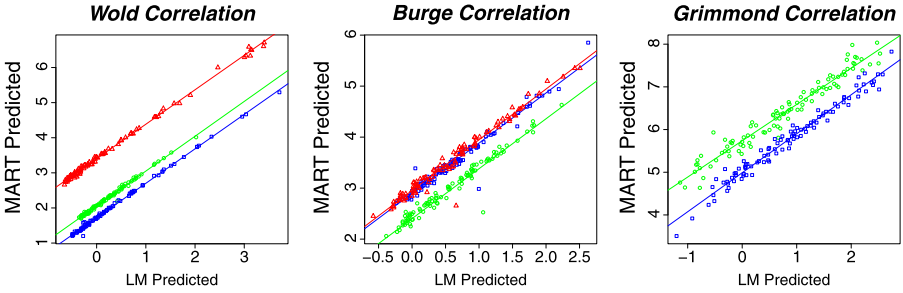

Fig. 3 Gene expression estimates of the MART and linear model. Estimates of the expression are plotted for the linear and MART models fit on each subdataset of the Wold, Burge and Grimmond data. Subdatasets are differentiated by color and point shape. Plotted on a log scale

Table 6 shows the Spearman rank correlation between the linear model and the MART model for each dataset using the sequence bias for gene expression estimation. Figure 3 compares the gene expression estimates of the MART method and the linear model.

Our work suggests that we can use the estimates from the linear model in place of the MART model. Since their results are very similar, the linear model may be a better choice due to its significantly lower computation time and easily interpretable parameters.

4 Conclusion

We propose a linear model for the sequence bias in RNA-seq read counts data using the neighborhood overlapping dinucleotides. We develop a penalized least squares algorithm for model-fitting. Fitting the linear model using a penalized least squares approach, we use weights to penalize parameters which are further away from the read in the surrounding sequence. We then use a data-driven method to determine an appropriate number of dinucleotides in the neighborhood sequence. Finding the sparse set of surrounding sequence which captures as much variation of read counts as

possible results in a great savings in computational cost, especially compared with a computationally intense method such as MART. We also find that the gene expression estimates from our model are highly correlated with the estimates from the non-linear model MART.

After we submitted this paper, we were made aware of some recent work in correcting bias in RNA-seq. In particular, Zheng et al. [15] directly corrects the gene expression estimates through nonparametric regression on all potential bias factors. Zheng et al. [15] focuses on gene level bias whereas our paper focuses on base level (nucleotide) bias. Roberts et al. [10] develops a Markov model with 744 parameters to model both gene level and base level biases. Hu et al. [6] develops a Poisson mixed effect model to model the base level bias one-gene-at-a-time. The computation of the latter two methods is expensive. On the other hand, the fragment bias considered in those papers may be integrated into our method to further improve the accuracy of transcript quantification.

Acknowledgements This research was supported by the National Science Foundation grant DMS 0800631. This research was partially supported by the National Science Foundation CAREER award DMS 1055815, and the Office of Science (BER), US Department of Energy to PM, National Institutes of Health grant 1R01 GM080503-03 to XH. The authors thank an Editor and an anonymous reviewer for constructive comments and suggestions, which led to a better presentation.

References

1. Bullard JH, Purdom E, Hansen KD, Dudoit S (2010) Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. BMC Bioinform 11:94
2. Cloonan N, Forrest ARR, Kolle G, Gardiner BBA, Faulkner GJ, Brown MK, Taylor DF, Steptoe AL, Wani S, Bethel G, Robertson AJ, Perkins AC, Bruce SJ, Lee CC, Ranade SS, Peckham HE, Manning JM, McKernan KJ, Grimmond SM (2008) Stem cell transcriptome profiling via massive-scale mRNA sequencing. Nat Methods 5(7):613–619
3. Dohm JC, Lottaz C, Borodina T, Himmelbauer H (2008) Substantial biases in ultra-short read data sets from high-throughput dna sequencing. Nucleic Acids Res 36:e105
4. Friedman JH (2001) Greedy function approximation: A gradient boosting machine. Ann Stat 29(5):1189–1232
5. Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378
6. Hu M, Zhu Y, Taylor JM, Liu JS, Qin ZS (2012) Using Poisson mixed-effects model to quantify transcript-level gene expression in RNA-Seq. Bioinformatics 28:63–68
7. Li J, Jiang H, Wong WH (2010) Modeling non-uniformity in short-read rates in RNA-Seq data. Genome Biol 11:R50
8. Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nat Methods 5(7):621–628
9. Nagalakshmi U, Wang Z, Waern K, Shou C, Raha D, Gerstein M, Snyder M (2008) The transcriptional landscape of the yeast genome defined by RNA sequencing. Science 320(5881):1344–1349
10. Roberts A, Trapnell C, Donaghey J, Rinn J, Pachter L (2011) Improving RNA-Seq expression estimates by correcting for fragment bias. Genome Biol 12(3):R22
11. Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, van Baren MJ, Salzberg SL, Wold B, Pachter L (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol 28:511–515
12. Wang ET, Sandberg R, Luo S, Khrebtukova I, Zhang L, Mayr C, Kingsmore SF, Schroth GP, Burge CB (2008) Alternative isoform regulation in human tissue transcriptomes. Nature 456:470–476
13. Wedderburn R (1974) Quasi-likelihood functions, generalized linear models, and the Gauss–Newton method. Biometrika 61:439–447

14. Wilhelm BT, Marguerat S, Watt S, Schubert F, Wood V, Goodhead I, Penkett CJ, Rogers J, Bahler J (2008) Dynamic repertoire of a eukaryotic transcriptome surveyed at single-nucleotide resolution. Nature 453(7199):1239–1243
15. Zheng W, Chung L, Zhao H (2011) Bias detection and correction in RNA-sequencing data. BMC Bioinform 12(1):290
16. Zhu Z, Liu Y (2009) Estimating spatial covariance using penalized likelihood with weighted L1 penalty. J Nonparametr Stat 21(7):925–942
17. Zou H (2006) The adaptive LASSO and its oracle properties. J Am Stat Assoc 101(476):1418–1429