A sparsity-based approach for detecting sequence bias in RNA-seq

Abstract

Shotgun RNA sequencing is a widespread technology useful for both the mapping and relative quantification of cellular transcript expression. A multitude of studies have investigated efficient and accurate methods for regularizing expression estimates in the presence of bias yet significant coverage variability persists. To investigate this a sparse linear modelling framework was adopted and effects were estimated via the `SVEN` algorithm on those stretches of nucleotides indicating sufficient residual coverage variability. Findings indicate a relative over-abundance of fragments with high C content and an under-abundance of fragments with high G content, possibly suggesting unaccounted for sequence effects.

1 Introduction

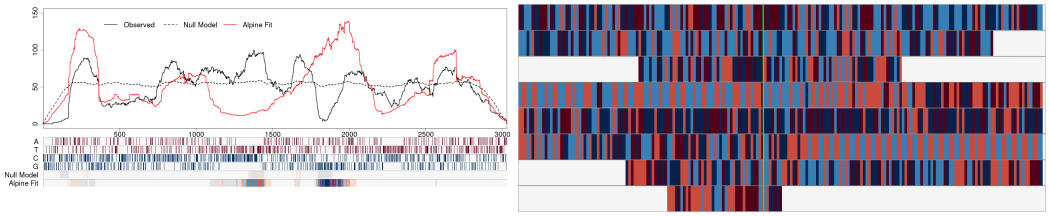
Shotgun sequencing of RNA fragments has proven a useful technique for both the mapping and quantification of transcripts comprising a cell [Wang et al., 2009]. By assuming the relative number of reads aligned to pre-specified gene annotations along a transcriptome is associated with the true cellular gene expression, researchers are able to assess differential mean expression across condition [Mortazavi et al., 2008]. These analyses have confirmed known and discovered previously unknown associations between, for example, the gene expression profiles of cancerous cells versus healthy cells. Furthermore, gains in sequencing efficiency coupled with persistent reductions in protocol cost have made the use of RNA-seq technology widespread.

Similar to microarray technology however, distributional non-uniformity is prevalent in the resultant mapped read coverage, much of which is likely a consequence of fragment features. This in turn biases estimates, yielding false-positives and unreliable results. [Roberts et al., 2011, Benjamini et al., 2012, Love et al., 2015]. In the case of multiple isoform transcripts, failure to account for bias effects can drastically alter estimated expression as entire exons may be incorrectly estimated; thus downstream inference on expression levels would be based on an incomplete or incorrect calculation having assigned expression to the wrong gene isoform (see Appendix Figure 1). Multiple studies, including those listed previously, have investigated this inconsistency and shown that adjusting for several sources of technical bias, such as fragment length and GC-bias effects due to PCR amplification, greatly reduces the residual variability and in turn strengthens inferential consistency between laboratory and sample.

Yet despite these advances, a great deal of unaccounted for variability in read coverage persists. The left panel of Figure 1 illustrates this by plotting the sum coverage computed along one single-isoform transcript. The solid black line is the realized fragment coverage and the red line is the predicted coverage after controlling for several known biases using state-of-the-art software [Love et al., 2015]. The dashed line represents the mean and may be interpreted as the mean transcript expression relative

*Paper submission for MIT 9.520.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.



(a) Coverage (b) Extracted Sequences

Figure 1

to some in-sample reference level. It is important to note how significantly the predicted coverage and realized coverage differ. This suggests the presence of unaccounted for or incorrectly estimated bias effects, yet the problem of how to identify meaningful effects given the large feature space associated with sequencing data and the categorical nature of nucleotides makes this difficult to investigate.

I address the problem by extracting stretches along each single-isoform transcript indicating sufficient miscalculation, post-bias regularization. The resultant sequences are encoded with a binary response indicating over-/under-estimation and the nucleotide sequence effect size is assessed under a sparse linear modelling framework. It is hypothesized that certain nucleotide sequences drastically affect a fragment's probability in being amplified during the PCR step of RNA sequencing and the sparse modelling framework will highlight any sequence patterns by estimating significant nucleotide effects as part of the active set. The work contributes in two areas: The results of the sparsity analysis suggest a previously unknown sequence effect arising from stretches of C's and G's causing pronounced spikes and drops in coverage. Second, to resolve the issue of the sparse feature space, an Elastic Net regularized linear model is fit by implementing the `SVEN` algorithm in R, the code of which may be made available to the research community [Zhou et al., 2014]. The paper proceeds as follows: preparation of the reads and model definitions are provided in Section 2. Section 3 presesnts the main results including an assessment of sparsity. Section 4 concludes with a brief discussion.

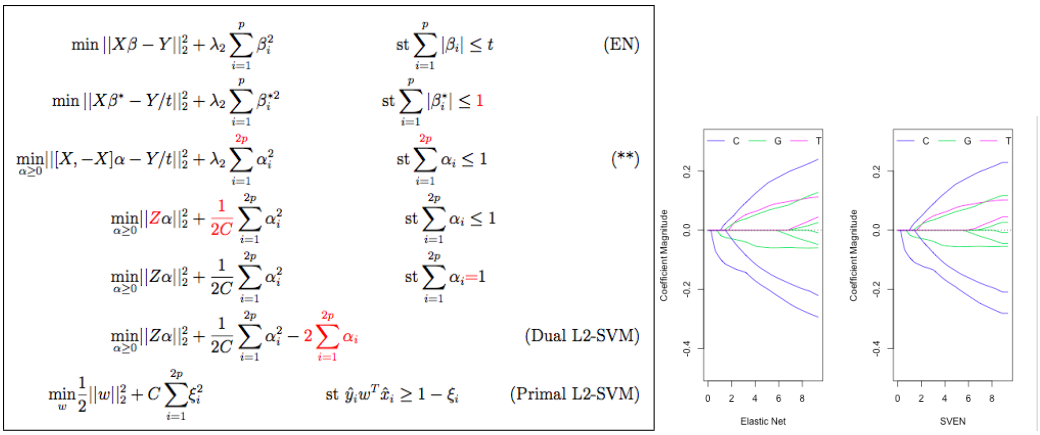
2 Methods

2.1 Preparation of RNA sequencing reads

The GEUVADIS Project provides high quality RNA-seq reads in conjunction with the 1000 Genomes Project [Lappalainen et al., 2013]. Three lymphoblastoid cell lines were downloaded, each of which were derived from the Tuscany population in Italy, and the raw RNA fragment counts were aligned to known transcript annotations along the HG19 reference human genome using the STAR sequence aligner [Dobin et al., 2013]. Only those fragments mapped uniquely to suitable single-isoform transcripts were retained for the subsequent analysis.

The resultant fragments were then split in two datasets conditionally upon transcript such that no transcript was present in both datasets. The Alpine package in R was then utilized to estimate the following known bias parameters on each dataset independently: fragment length, fragment GC-content, fragment relative position along the transcript, and fragment start bias modelled with random hexamer priming [Love et al., 2015]. To the author's knowledge these biases comprise the set of known RNA-seq bias parameters. These parameters were estimated under a poisson GLM framework and subsequently used to predict read coverage by applying the estimated coefficients *across* dataset fold. The resultant fragment-level predictions were aggregated and the sum coverage along each transcript was computed. It should be noted that one would obtain the plot in Figure 1 by plotting this series for a single transcript.

Predicted coverage was then subtracted from realized coverage and the resulting residuals were smoothed using LOESS non-parametric regression. The nucleotide sequence for those regions along the transcript with suitably large smoothed residuals were extracted. A large positive difference suggests a spike in realized coverage relative to the bias-regularized estimates and conversely, a large negative difference suggests a drop in realized coverage. Those extracted sequences from the transcript presented in the coverage plot in Figure 1 are seen on the righthand side of the figure.



(a) The SVEN algorithm (b) Regularization Path Equivalence

Figure 2

The collected nucleotide sequences across all single-isoform transcripts in a given sample were extracted and represent N observations, X_i, with a corresponding binary outcome, $Y_i \in \{-1, 1\}$, indicating a drop or spike in coverage, respectively. Each observation was encoded by centering around the nucleotide with the maximal absolute smoothed residual and expanding outwards. The index of this "centered" nucleotide will be referred to as position 0. That is to say, each X_i is a vector of length $P = n_i^L + 1 + n_i^R$, where n_i^L is the number of nucleotides leading up to the absolute maximum (position 0) and n_i^R is the number of nucleotides following, with the constraint that $n_i^L + n_i^R + 1$ is no larger than the length of the corresponding sequence. In the experiments that follow, n^L and n^R are both capped at 100, thus sequences shorter than 201 base-pairs are encoded with a padding of *NA* at the start and end to ensure a vector of length P. Each observation was then encoded numerically using 1-hot encoding (i.e. a binary variable for each A, T, C, G thus expanding the vector by a factor of 3 plus a reference level).

2.2 Reducing the Elastic Net to a squared hinge-loss SVM

The Elastic Net (EN) regularized linear model was fit by implementing `SVEN`, an algorithm which augments the data matrix, trains a soft-margin linear support vector machine (SVM) minimizing the squared hinge loss, and scales the resultant coefficients back to the original magnitude of the EN [Zhou et al., 2014]. The approach is particularly suited for $P >> N$ and $N >> P$, cases in which leveraging the SVM primal-dual relationship is advantageous, and is thus also easily parallelizable by implementing parallel SVM solvers. In fact, the data augmentation step is of order $O(NP)$, fitting an SVM in the primal formulation is $O(N^3)$, and in the dual $O(P^3)$, for the worst possible cases. The time complexity for the EN solved via coordinate descent is more difficult to calculate but for comparison the LARS algorithm used to optimize the standard Lasso is of $O(P^3 + NP^2)$ [Friedman et al., 2010, Efron et al., 2004].

The lefthand panel of Figure 2 demonstrates the mathematical correspondence between the two formulations. The equivalance starts from the Elastic Net formulation in line 1 and augments the objective and constraints to the formulation of the L2 soft-margin SVM in the dual. X is the $N \times P$ design matrix whereas Z is the augmented $N \times 2P$ matrix. Line 3 makes use of slack variables to tighten the L1 constraint on the weight coefficients $\beta^* \in \mathbb{R}^P$. This may be represented symbolically as $\beta^*_{\blacklozenge} = \beta^*_{\lozenge} = [I_P, -I_P]\beta^*_{\triangle}$, where $\triangle$ represents a positively-valued, tight L1-norm on unit support, $\lozenge$ represents a tight L1-norm on unit support, and $\blacklozenge$ represents the L1-norm on the unit support under an inequality constraint [Jaggi, 2014]. Other major results are highlighted in red and it should be noted that the resultant weights in the dual $\alpha \in \mathbb{R}^{2P}$, in the primal $w \in \mathbb{R}^N$, and the augmented primal "observations" $\hat{x}_i \in \mathbb{R}^N$ for $i = 1, \ldots, 2P$, $\hat{y}_i = -1$ for $i = 1, \ldots, P$, and $\hat{y}_i = 1$ for $i = P + 1, \ldots, 2P$. The artificial outcome variable $\hat{y}_i$ is an artifact of the reformulation performed on line 3. The righthand panel of Figure 2 verifies the regularization path equivalence for 10 randomly selected parameters. The EN was fit with `glmnet` and the `SVEN` algorithm was implemented in R

[Friedman et al., 2010]. It is assuring to verify that the support vectors may be rescaled to yield exactly those values obtained from fitting EN directly.

An interesting side effect from utilizing this concordance arises by considering the augmentation that has taken place. Notably, the observations selected as the support vectors in the dual formulation are exactly those non-zero variables estimated in the EN active set. It is worth contrasting the formulated SVM in `SVEN` against the typical use case of the SVM to highlight this implication. In the latter, one provides the SVM algorithm a matrix of M rows and Q variables and is returned a set of $m \leq M$ nonzero weights, each of which correspond to those observations which were selected to the active set, i.e. the support vectors. In other words, sparsity acts along the rows of the data matrix and inner products are computed between observations. On the other hand, when one fits the standard EN (or equivalently fits an SVM with a linear kernel through `SVEN`) for a matrix of size $N \times P$, they are returned a set $p \leq P$ nonzero weights corresponding to *features* in the active set. Thus sparsity acts along the columns of the data matrix.

Now consider the addition of a linear kernel matrix under the square loss objective function and ignore the regularization constants from the EN for notational ease. This minimization may be expressed as $\min_{w \in \mathbb{R}^N} \sum_{i=1}^{N} ||K(x, x_i)w_i - y_i||_2^2$, and inner products are computed between rows such that sparsity selects $n \leq N$ support vectors ($w_i > 0$). Selection of the kernel function determines the dimensionality of the feature space and may be greater than P in the cases of the polynomial and Gaussian kernels. Addition of a kernel to the SVM case operates similarly; the minimization in the dual is expressed as $\min_{\alpha \in \mathbb{R}^M} \sum_{i=1}^{M} ||K(\hat{x}, \hat{x}_i)\alpha_i||_{\mathcal{H}}^2$, and sparsity selects $m \leq M$ support vectors ($\alpha_i > 0$). Recall $\hat{x}_i \in \mathbb{R}^Q$ and $K(\hat{x}, \cdot)$ is a $M \times M$ kernel matrix.

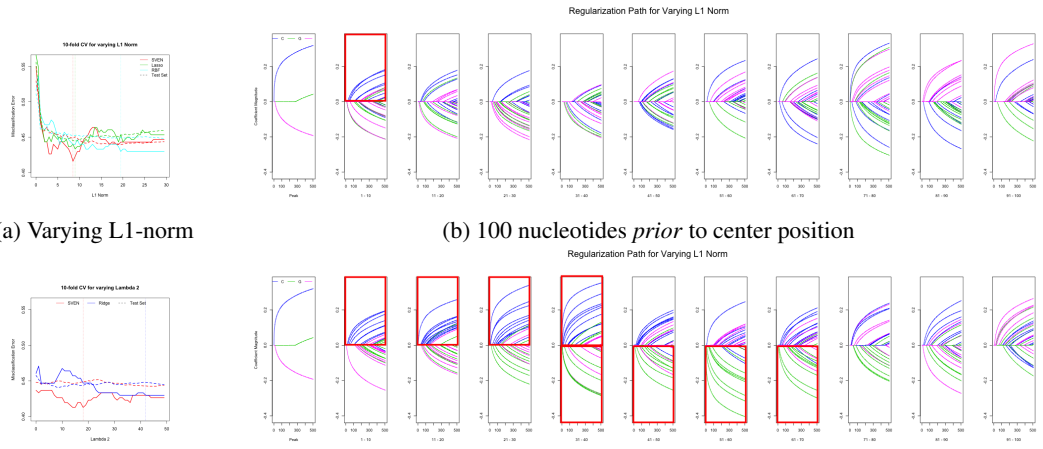
To see the relationship back to the kernelized `SVEN` algorithm note that $N = Q$, $2P = M$, and expand $\hat{x}_i = X_i^T - y$, with $X_i^T, y \in \mathbb{R}^N, i = 1, \ldots, 2P$ (X_i^T represents the i^{th} column of the original design matrix X and W.L.O.G. assume y was scaled by t). Making use of the feature map representation we have that $\Phi(\hat{x}_i) = \Phi(X_i^T) - \Phi(y)$ and this is indeed calculable since $K(\hat{x}, \hat{x})_{i,j} = \langle \Phi(\hat{x}_i), \Phi(\hat{x}_j) \rangle = \langle \Phi(X_i^T) - \Phi(y), \Phi(X_j^T) - \Phi(y) \rangle = K(X_i^T, X_j^T) - K(X_i^T, y) - K(X_j^T, y) + K(y, y)$. Thus the kernelized `SVEN` solves the dual formulation $\min_{\alpha \in \mathbb{R}^{2P}} \sum_{i=1}^{2P} ||K(\hat{x}, \hat{x}_i)\alpha_i||_{\mathcal{H}}^2 = \min_{\alpha \in \mathbb{R}^{2P}} \sum_{i=1}^{2P} ||\Phi(X_i^T)\alpha_i - \Phi(y)||_{\mathcal{H}}^2$, where $K(\cdot, \cdot)$ is the $2P \times 2P$ kernel matrix and $\Phi(\cdot)$ the associated feature map. By considering the data augmentation in tandem with the primal SVM formulation it is therefore implied that the kernelized `SVEN` algorithm computes inner products between rows corresponding to the original $2P$ features in a potentially $\geq N$-dimensional feature space (depending upon the kernel selection) with the original outcome variable mapped to the same kernel space. In what follows this topic will be explored further by considering several different model formulations. Misclassification error will be assessed with a sparsity analysis and brief discussion concluding.

2.3 Model formulations

As noted above the soft margin L2-SVM may be shown equivalent to fitting the Elastic Net through data augmentation and scaling. This will yield $3 \times p$ coefficients where $p = (n_{\max}^L + n_{\min}^R + 1)$ minimizing the objective function provided on line 1 of the lefthand panel of Figure 2. Recall the multiplication by a factor of three is due to the 1-hot encoding of the data with a reference category.

As $\lambda_2 \to 0$ the EN reduces to the Lasso and similarly, as $c \to \infty$, the soft margin SVM reduces to a hard margin SVM [Jaggi, 2014]. For this reason a hard margin SVM was also implemented and again the support vectors were transformed back to the scaled equivalence from a standard Lasso. Alternatively, as $\lambda_1 \to 0$, the EN reduces to the ridge regression specification and the corresponding SVM is an L2-SVM constrained to the unit ball, a formulation of little practicality. In this case the `SVEN` algorithm is inefficient; one only need to solve a system of linear equations involving a single PSD matrix inversion. In the results that follow the ridge specification was fit in this manner.

The `SVEN` algorithm implements a linear kernel in the SVM step and allows for the interpretation of support vectors as the non-zero coefficients from the EN fitting. In fact, this also suggests a correspondence between the dimensionality of the kernel space and the number of observations N of the original dataset (pre-data augmentation). Thus when utilizing the linear kernel one is fitting an SVM by computing the inner product in a feature space of dimension N. If on the other hand a Gaussian kernel is used, the interpretation becomes that of training an SVM in an infinitely-dimensional feature space and would therefore imply $N \to \infty$. In this case, however, the features



(a) Varying L1-norm

(b) 100 nucleotides *prior* to center position

(c) Varying λ_2

(d) 100 nucleotides *after* center position

Figure 3

are linearly independent but not necessarily orthogonal, implying N are no longer independent and identically distributed random observations. Further, this formulation also assumes any higher moments of both X and the outcome Y are estimable. As noted by Jaggi (2014), it does indeed seem to make the regression problem more difficult by estimating higher moments of Y. Despite this caveat however, the implication of adding rows to the design matrix and increasing N via the kernel trick is quite intuitive and results from modifying the `SVEN` algorithm to use a Gaussian kernel are included in the prediction results below.

3 Results

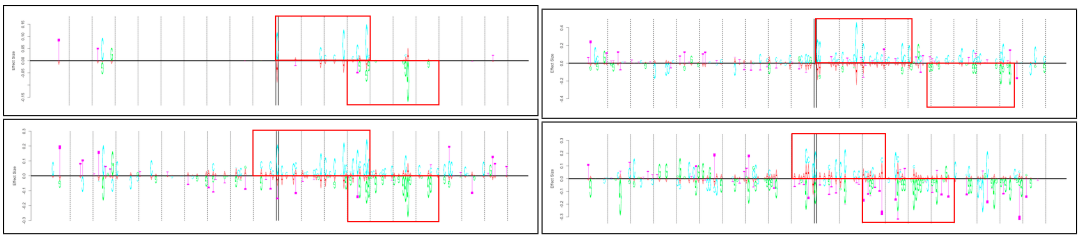
3.1 Model performance

Model performance is evaluated by performing 10-fold cross evaluation and computing the mean validation error. Test set error is calculated with the test set simply being the second within-sample partition. That is, the reads from a given sample were conditionally split based on transcript prior to estimating the bias parameters in the data preparation (see Preparation of RNA sequencing reads). This intra-fold estimation and cross-fold prediction is done to mitigate overfitting within each sample. All models perform similar with the EN and the EN-RBF slightly outperforming the others. Notably, the plots indicate relative invariance to parameter selection for the test set. This suggests that even under high regularization, any sparse coefficients estimated do improve the prediction error relative to the null. As shown in the subsequent section, regardless the value for the regularization parameters, a general pattern in sequence effect may be noticed.

3.2 Assessment of sparsity

A key benefit of fitting the EN regularized linear model is the estimated coefficients will be sparse in the sense that null-effect coefficients will be set to zero. The regularization paths for varying λ_1 are presented in the righthand panels of Figure 3 for sample ID:188088. The colors indicate nucleotide (C=blue, G=green, T=purple) and the panels separate the coefficients by location. The top row of panels is for those coefficient effects corresponding to nucleotides positioned prior to the position 0 nucleotide (the nucleotide with the maximal absolute residual deviance on which the sequence was centered). The bottom row of panels is alternatively for those nucleotides positioned after position 0. As one traces from left to right across panels, one is observing the locational effects of different nucleotide sequences. Thus the first panel on the left corresponds to position 0, the next panel to the right positions 1-5, then positions 6-10, etc.

The red boxes drawn on the panels in Figure 3 highlight the sequence effects due to C and G nucleotides. The high concentration of blue (C) positive effects suggests an over-abundance of realized coverage relative to what is expected for those stretches along the transcript containing



(a) sample:188052; L1:13(top),85(bottom) (b) sample:188145(top),188088(bottom); L1:85

Figure 4

a high concentration of the base C. Recalling the method in which the data were prepared, one may conclude that there is an over-abundance of mapped fragments containing stretches of the base C. Similarly, there appears an under-abundance of mapped fragments containing high G content, albeit this effect appears dependent upon location[2]. Furthermore, these effects do not appear to be dependent upon the values of the regularization parameters.

3.3 Sparse sequence effects

The visualizations in Figure 4 illustrate the estimated sequence effects for two values of the L1-norm constraint on the lefthand side and for two different samples (datasets) on the righthand side. The position 0 nucleotides are indicated by the solid black vertical lines and the dashed lines represent location effects due to moving 10 nucleotides in either direction from position 0. Red boxes highlight trends.

Of note is the abundance of postively-valued coefficients corresponding to the nucleotide base C clustering around, and specifically after, position 0. Similarly, we note an opposite and slightly smaller effect for G occurring at base locations further from position 0. Importantly, as the L1-norm increases (relaxing the regularization constraint), the trend of positively-valued C effects and negatively-valued G effects is consistent across sample. As noted previously, these also do not appear dependent upon selection of the regularization parameters.

These results suggest nucleotide sequences composed of stretches of C are over-represented among sequences of highly-residual coverage and sequences composed primarily of G are under-represented. This may suggest that GC-bias arising from PCR amplification may need to adjust differentially for fragments with high G versus high C content. Alternatively, the bases G and C are complementary and differential effects for complements lack biological motivation and/or cause.

4 Discussion

Much work has been done to investigate efficient and optimal methods for regularizing gene expression estimates coming from RNA-seq in the presence of bias. In this paper I have proposed a sparsity-based approach for discovering new and meaningful nucleotide sequence effects. The approach has the benefit of potentially discovering unknown sources of coverage bias and does not necessarily aim for efficiency. To address the issue of efficiency however, the `SVEN` algorithm was implemented in the programming language `R`. It is hoped potential package submission to CRAN will allow for researchers to implement the algorithm with greater ease. Due to balanced N and p in the samples considered the gains from utilizing SVM solvers were not attained. To achieve efficiency gains, the author hopes to extend the work to include 2-hot encoding in addition to the 1-hot encoding previously considered. 2-hot encoding would expand the number of columns of the data matrix by a factor of $4^2 = 16$ making the use of SVM solvers more appealing as well as potentially capturing nonlinear sequence effects. Several alternative formulations of the EN and the corresponding SVM were also presented.

Interestingly, the trends in the sparsity analysis highlight stretches of the base C as over-represented and stretches of the G as under-represented. From a biological viewpoint this may suggest that

[2]Fragment length bias is mitigated through estimation of an empirical density. For the samples considered fragments typically ranged between 80-200 bps in length.

methods for controlling GC-content bias could be expanded to account for G and C effects separately. Current protocol as suggested by Love et al. (2015) includes estimating a natural cubic spline for fragment-level GC-content and the authors noted a sharp under-representation for both high and low GC-content fragments. The results found herein suggest a differential effect for C and G, in which stretches of C are amplified while stretches of G are suppressed. Further, this trend is found to be consistent in all samples considered and is robust to the selection of regularization parameters. Despite these findings, however, due to the complementarity of the bases C and G, the molecular motivation is not clear. Further investigation of this sequence effect is necessary indeed.

References

Benjamini, Yuval, and Terence P. Speed. "Summarizing and correcting the GC content bias in high-throughput sequencing." *Nucleic acids research* (2012): gks001.

Dobin, Alexander, et al. "STAR: ultrafast universal RNA-seq aligner." *Bioinformatics* 29.1 (2013): 15-21.

Efron, Bradley, et al. "Least angle regression." *The Annals of statistics* 32.2 (2004): 407-499.

Friedman, Jerome, Trevor Hastie, and Rob Tibshirani. "Regularization paths for generalized linear models via coordinate descent." *Journal of statistical software* 33.1 (2010): 1.

Jaggi, Martin. "An equivalence between the lasso and support vector machines." *Regularization, Optimization, Kernels, and Support Vector Machines* (2014): 1-26.

Love, Michael I., John B. Hogenesch, and Rafael A. Irizarry. "Modeling of RNA-seq fragment sequence bias reduces systematic errors in transcript abundance estimation." *bioRxiv* (2015): 025767.

Lappalainen, Tuuli, et al. "Transcriptome and genome sequencing uncovers functional variation in humans." *Nature* 501.7468 (2013): 506-511.

Mortazavi, Ali, et al. "Mapping and quantifying mammalian transcriptomes by RNA-Seq." *Nature methods* 5.7 (2008): 621-628.

Roberts, Adam, et al. "Improving RNA-Seq expression estimates by correcting for fragment bias." *Genome biology* 12.3 (2011): 1.

Wang, Zhong, Mark Gerstein, and Michael Snyder. "RNA-Seq: a revolutionary tool for transcriptomics." *Nature reviews genetics* 10.1 (2009): 57-63.

Zhou, Quan, et al. "A reduction of the elastic net to support vector machines with an application to gpu computing." *arXiv preprint* arXiv:1409.1976 (2014).

5 Appendix

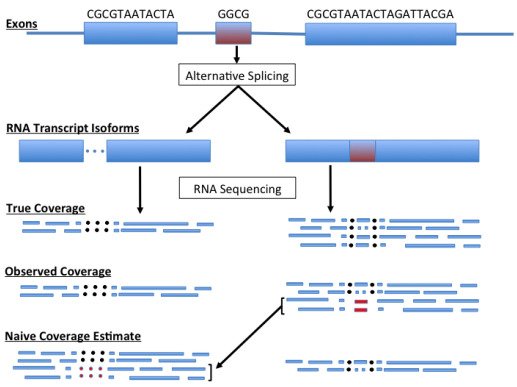


Figure 5: Appendix: Effects of bias misspecification