Supplementary Materials for "IsoDOT Detects Differential RNA-isoform Usage with respect to a Categorical or Continuous Covariate with High Sensitivity and Specificity"

A Calculation of effective length

An RNA-seq fragment is a segment of RNA to be sequenced. Usually only part of an RNA-seq fragment is sequenced: one end or both ends, hence single-end sequencing or paired-end sequencing. All the discussions in this section are for paired-end reads, though the extension to single-end reads is straightforward. The minimum fragment size is the read length, denoted by d. This happens when the two reads of a fragment completely overlap. We impose an upper bound for the fragment length based on prior knowledge of the experimental procedure and denote the upper bound by l_M. Then the fragment length l satisfies $d \leq l \leq l_M$. We denote the distribution of the fragment length for sample i by $\varphi_i(l)$, which can be calculated using observed read alignment information.

For the i-th sample, the effective length of exon j of r_j base pairs (bps) is

$$\eta_{i,\{j\}} = f(r_j, d, l_M, \varphi_i) = \begin{cases} 0 & \text{if } r_j < d \\ \sum_{l=d}^{\min(r_j, l_M)} \varphi_i(l)(r_j + 1 - l) & \text{if } r_j \geq d \end{cases}.$$

If $r_j < d$, the exon is shorter than the shortest fragment length, and thus the effective length of this exon is 0. In other words, no RNA-seq fragment is expected to overlap and only overlap with this exon. If $r_j \geq d$, the effective length is $r_j + 1 - l$, i.e., there are $r_j + 1 - l$ distinct RNA-seq fragments that can be sequenced from this exon (Figure 1). Then $\sum_{l=d}^{\min(r_j, l_M)} \varphi_i(l)(r_j + 1 - l)$ is summation across all likely fragment lengths, weighted by the probability of having fragment length l.

In the following discussions, to simplify the notation, we skip the subscript of i. For two exons j and k ($j < k$) of lengths r_j and r_k, which are adjacent in the transcript, the effective length for the fragments that cover both exons is

$$\eta_{\{j,k\}} = f(r_j + r_k, d, l_M, \varphi) - \eta_{\{j\}} - \eta_{\{k\}}. \quad (1)$$

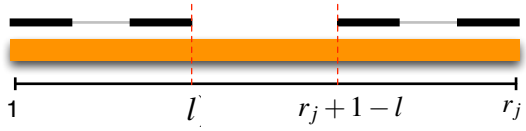

Figure 1: An illustration of effective length calculation for an exon of r_j bps and RNA-seq fragment of l bps. The orange box indicates the exon, and the black lines above the orange box indicate two RNA-seq fragments, while each RNA-seq fragment is sequenced by a paired-end read. There are $r_j + 1 - l$ distinct choices to select an RNA-seq fragment of l bps from this exon, and thus the effective length is $r_j + 1 - l$.

For three exons j, h, and k ($j < h < k$) of lengths r_j, r_h and r_k, which are adjacent in the transcript, the effective length for the fragments that cover all three exons is

$$\eta_{\{j,h,k\}} = f(r_j + r_h + r_k, d, l_M) - \eta_{\{j,h\}} - \eta_{\{h,k\}} - \eta_{\{j,(h),k\}} - \eta_{\{j\}} - \eta_{\{h\}} - \eta_{\{k\}},$$

where $\eta_{\{j,(h),k\}}$ is the effective length in the scenario that the transcript covers consecutive exons j, h, and k, whereas the observed paired-end read only covers exons j and k.

$$\eta_{\{j,(h),k\}} = \begin{cases} 0 & \text{if } (r_j, r_h, r_k) \in R_1 \\ \sum_{l=2d+r_h}^{\min(r_j+r_h+r_k, l_M)} \varphi(l)\delta_l & \text{otherwise} \end{cases}$$

where $R_1 = \{(r_j, r_h, r_k) : \ r_j < d \text{ or } r_k < d \text{ or } r_h + 2d > l_M\}$, and $\delta_l = \min(r_j, l - r_h - d) - \max(d, l - r_h - r_k) + 1$. The above formula is derived by the following arguments. Let l_j and l_k be the lengths of the parts of the fragment that overlaps with exon j and k, respectively. Given l, the restriction of l_j and l_k are $l = l_j + l_k + r_h$, $d \le l_j \le r_j$, and $d \le l_k \le r_k$, and thus the range of l_j is $\max(d, l - r_h - r_k) \le l_j \le \min(r_j, l - r_h - d)$. For more than 3 consecutive exons, the effective lengths can be calculated using recursive calls to the above equations.

In practice, a few sequence fragments may be observed even when the effective length is zero, which may be due to sequencing errors. To improve the robustness of our method, we modify the design matrix $\mathbf{X}$ by adding a pre-determined constant `eLenMin` to each element of $\mathbf{X}$.

B Selection of candidate isoforms

For each gene, we select a set of candidate isoforms given the fragment counts at each exon set. We define a start exon as an exon that is only connected to downstream exons and an end exon as an exon that is only connected to upstream exons. An initial set of start and end exons can be identified simply by examining the observed exon sets.

Next, we seek to find more start and end exons by identifying break points where the read-depth of two adjacent exons are different. Specifically, suppose the gene of interest has h exons. Let $y_{\{k\}}$ be the number of fragments overlapping the kth exon of this gene. We apply a Pearson chi-squared test to assess whether the frequency distribution of $y_{\{k-1\}}$ and $y_{\{k\}}$ is significantly different from theoretical expectation based on their effective lengths. For $k = 2$, ..., h, there are $h - 1$ possible break points, which correspond to $h - 1$ p-values: $\texttt{pB}_1$, ..., $\texttt{pB}_h$. We order those possible break points by the corresponding p-values in ascending order and select the top

$$\min\left(\texttt{maxBreaks},\ \sum_{k=2}^{h} I(\texttt{pB}_k < \texttt{pvalBreaks})\right)$$

break points, where $I(\cdot)$ is an indicator function, `maxBreaks` and `pvalBreaks` are two pre-set parameters. `maxBreaks` is the maximum number of break points, with default value 5, and `pvalBreaks` is a p-value cutoff, with default value 0.05. If the k-th break points is selected, the $(k-1)$th exon is added to the set of start exons and the kth exon is added to the set of end exons. After identifying all possible start and end exons, we can construct all isoforms that have consecutive exons.

For each exon set, we assign a p-value to quantify whether it is expressed. Suppose there are n_T fragments for the gene of interest and among them n_j fragments are from the jth exon set. Then the expression p-value is $\texttt{pE}_j = \texttt{pbinom}(n_j, n_T, l_j/l_T)$, where $\texttt{pbinom}(\cdot, n, \pi)$ is cumulative binomial distribution function with n trials and probability of success π, l_j is the effective length of the jth exon set, and l_T is the total effective length of this gene. We claim the jth exon set is expressed if

$$\texttt{pE}_j \geq \texttt{pvalExpress} \text{ and } \frac{n_j/l_j}{n_T/l_T} > \texttt{foldExpress},$$

where `pvalExpress` and `foldExpress` are two pre-set parameters, with default values 0.01 and 1/5, respectively.

Finally, we select all the expressed exon sets that harbor at least on exon-skipping event, and order them by the $\texttt{pE}_j$ in a descending order. Then for each of these ordered exon sets, we construct new RNA-isoforms by inserting this exon set into each existing isoform if this exon set is compatible with the isoform. We stop adding more isoforms if either

$$q/m > \texttt{pMaxRel} \text{ or } q > \texttt{pMaxAbs}$$

where q is the number of isoforms, `pMaxRel` and `pMaxAbs` are pre-set parameters with default values 10 and 2000, respectively. In oder words, we allow the number of isoforms to be at most 10 times the number of exon sets and the total number of isoforms to be at most 2000. Users can change these default values. Our penalized regression can handle the situation `pMaxRel`=100 and `pMaxAbs`=100,000; however it may significantly reduce the computational efficiency.

C Model fitting of the penalized negative binomial regression

Let $f(y_i; \mu_i, \phi)$ be the density function for a negative binomial distribution with mean μ_i and dispersion parameter ϕ (hence variance $\mu_i + \phi\mu_i^2$):

$$f(y_i; \mu_i, \phi) = \frac{\Gamma(y_i + 1/\phi)}{y_i!\Gamma(1/\phi)} \left(\frac{1}{1+\phi\mu_i}\right)^{1/\phi} \left(\frac{\phi\mu_i}{1+\phi\mu_i}\right)^{y_i}. \tag{2}$$

As $\phi \to 0$, $f(y_i; \mu_i, \phi)$ converges to Poisson distribution with mean μ_i. While all the following discussions focus on negative binomial distribution, they can be easily extended to Poisson situation and we omit the details here. Using negative binomial distribution, the log likelihood is

$$l(\mathbf{y}, \boldsymbol{\mu}, \phi) = \sum_{i=1}^{n} \left[\log\left(\frac{\Gamma(y_i + 1/\phi)}{y_i!\Gamma(1/\phi)}\right) + y_i \log\left(\frac{\phi\mu_i}{1+\phi\mu_i}\right) - \frac{1}{\phi}\log(1+\phi\mu_i)\right], \tag{3}$$

where $\mathbf{y} = (x_1, ..., x_n)^\mathsf{T}$, $\boldsymbol{\mu} = (\mu_1, ..., \mu_n)^\mathsf{T}$, and n indicates sample size. We further assume $\mu_i = \sum_{j=1}^{J} x_{ij} b_j$, where $b_j \geq 0$, and maximize the penalized log likelihood

$$l(\mathbf{y}, \boldsymbol{\mu}, \phi) - \sum_{j=1}^{p} q(b_j), \tag{4}$$

where $q(b_j) = \lambda \log(b_j + \tau)$, and λ and τ are two tuning parameters. In contrast to conventional penalized GLM, we employ a non-canonical link function, does not use an

intercept, and impose a set of constraints that $b_j \geq 0$ for $j = 1, 2, ..., J$. We maximize the likelihood by iteratively updating regression coefficients b_j and dispersion parameter ϕ. Following Friedman et al. [1], we approximate the likelihood part in equation (4) by a quadratic approximation:

$$l_Q(\mathbf{y}, \boldsymbol{\mu}, \phi) = -\sum_{i=1}^{n} w_i \left(y_i - \sum_{j=1}^{p} x_{ij} b_j \right)^2,$$

where $w_i = 1/\left(\hat{\mu}_i + \phi \hat{\mu}_i^2\right)$, and $\hat{\mu}_i$ is the estimate of μ_i in the previous iteration. Then to solve $\mathbf{b}$, we just need to solve the following penalized least squares problem.

$$\sum_{i=1}^{n} w_i \left(y_i - \sum_{j=1}^{p} x_{ij} b_j \right)^2 + \sum_{j=1}^{p} q(b_j),$$

subject to the constraints of $b_j \geq 0$ for $1 \leq j \leq J$. We employ a modified Iterative Adaptive Lasso (IAL) algorithm [2] to solve this problem. Given estimates of b_j $(1 \leq j \leq p)$, ϕ can be re-estimated by maximizing the conditional likelihood of ϕ.

Specifically, the implementation includes the following four levels of loops:

- OUTER LOOP: Iterate across all combinations of tuning parameter λ_n and τ_n.
- MIDDLE LOOP 1: This corresponds to the loop of iteratively update b_j $(1 \leq j \leq p)$ and ϕ. For each given ϕ, we carry out the next two nested loops to estimate b_j and then re-estimate ϕ.
- MIDDLE LOOP 2: This corresponds to the loop of IRLS (Iterative Re-weighted Least Squares). Update the quadratic approximation l_Q using current estimate of b_j $(1 \leq j \leq p)$ and ϕ.
- INNER LOOP: Run the modified IAL to re-estimate b_j $(1 \leq j \leq p)$ on the penalized weighted least squares problem.

The modified IAL algorithm is as follows. It is different from the IAL [2] in that the regression coefficients need to be non-negative and we remove the step of estimating residual variance to improve the computational efficiency and robustness.

1. INITIALIZATION: initialize b_j with zero's or estimate from previous IRLS iteration, and initialize $\kappa_j = (b_j + \tau)/\lambda$, where $1 \leq j \leq p$.

2. Iterative Updates:

   (a) For $j = 1, ..., p$, update b_j,

   $$\begin{cases} b_j = \bar{b}_j - 1/\kappa_j & \text{if } \bar{b}_j > 1/\kappa_j \\ b_j = 0 & \text{otherwise} \end{cases},$$

   where

   $$\bar{b}_j = \left(\sum_{i=1}^{n} x_{ij}^2 \right)^{-1} \sum_{i=1}^{n} x_{ij} \left(y_i - \sum_{k \neq j} x_{ik} b_k \right).$$

   (b) Update κ_j: $\kappa_j = (b_j + \tau)/\lambda$.

This IAL algorithm is converged if the coefficient estimates $\hat{b}_1, ..., \hat{b}_p$ have little change.

Tuning parameter selection is a crucial step for any penalization method. We select the tuning parameters λ and τ by a two-dimensional grid search. By default, we search across 10 values of λ and 3 values of τ, which are 30 tuning parameter combinations. Larger λ and smaller τ leads to stronger penalty, and thus we first choose the ratio λ/τ's so that they are uniformly distributed in log scale and the largest λ/τ is large enough to penalize all coefficients to 0. Then τ's are chosen so that they are uniformly distributed in log scale with largest τ being 0.1. Finally for each ratio $r = \lambda/\tau$ and for each τ, λ can be calculated as $r\tau$. Simulations show that the results are similar if we carry out grid search for 500 or 50 tuning parameter combinations.

Through the two-dimensional grid search, we choose the combination of λ and τ that minimizes BIC or extended BIC [3] if $n > p$ or $n \leq p$, respectively. If we only study the expression of known RNA isoforms, p is often smaller than n (e.g., see Supplementary Figure 4), and thus BIC is used. In contrast, if we detect de novo RNA isoforms, p is often larger than n (e.g., see Supplementary Figure 5), and thus extended BIC is used. For hypothesis testing of the isoform usage, the rule (BIC or extended BIC) is chosen based on the alternative model and the same rule is applied to the null model. More specifically, BIC is defined as

$$\texttt{BIC} = -2l(\hat{\Theta}) + s \log(n),$$

where $l(\hat{\Theta})$ is log likelihood given parameter estimates $\hat{\Theta}$, s is the number of non-zero coefficients after variable selection, and n is sample size. Following Chen and Chen (2012) [3], the extended BIC is defined as

$$\texttt{extBIC} = -2l(\hat{\Theta}) + s\log(n) + 2\gamma \log p,$$

where $0 \leq \gamma < 1 - 1/(2\kappa)$ given $p = O(n^{\kappa})$. In our simulation and real data studies, since we restrict the number of covariates $p \leq 10n$, we set $\kappa = 1$ and choose $\gamma = 1/2$. Tuning parameter selection is an active research area and we do not claim our approach is optimal. However our hypothesis-testing framework rely on parametric bootstrap to resample RNA-seq read counts to calculate p-values. This resampling-based p-value calculation is robust to bias due to suboptimal tuning parameters because any bias that influences the null distribution of the test statistic can be captured through resampling. On the other hand, optimal tuning parameter selection may improve the power of our method.

D Mouse haploperidol treatment experiment

Ethics Statement. All animal work was conducted in compliance with the "Guide for the Care and Use of Laboratory Animals" (Institute of Laboratory Animal Resources, National Research Council, 1996) and approved by the Institutional Animal Care and Use Committee of the University of North Carolina.

Animals. The mice used in this study were N=2 inbred C57BL/6J females (one placebo treated, one drug treated) and N=2 (129S1Sv/ImJ x PWK/PhJ)F1 females (one placebo treated, one drug treated). All animals were bred at UNC from mice that were less than 6 generations removed from founders acquired from the Jackson Laboratory (Bar Harbor, ME). Animals were maintained on a 14 hour light, 10 hour dark schedule with lights on at 0600. The housing room was maintained at 20-24C with 40-50% relative humidity. Mice were housed in standard 20cm $\times$ 30cm ventilated polysulfone cages with laboratory grade Bed-O-Cob bedding. Water and Purina Prolab RMH3000 were available ad libitum. A small section of PVC pipe was present in each cage for enrichment.

Drug treatment. Slow release haloperidol pellets (3.0 mg/kg/day; Innovative Research of America; Sarasota, FL)[4] were implanted subcutaneously with a trocar at 8 weeks of age. Blood plasma was collected via tail nick for a drug concentration assay after 30 days of exposure to haloperidol. Steady-state concentration of haloperidol within the clinically relevant range (10-50 nanomoles/L, nM, or 3.75-19 ng/ml)[5] was achieved for both drug treated animals (C57BL/6J: 19nM, (129S1Sv/ImJ x PWK/PhJ)F1: 24 nM).

Tissue collection. Mice were sacrificed at 12 weeks of age (following 30 days of drug or placebo treatment) by cervical dislocation without anesthesia to avoid its confounding effects on gene expression. Mice were removed from their home cages at 9:00 AM and sacrificed between 10:00 AM and 12:00 PM. Whole brain was rapidly collected, snap frozen in liquid nitrogen, and pulverized to a fine powder using a BioPulverizer unit (BioSpec Products, Bartlesville, OK).

RNA extraction. Total RNA was extracted from $\sim$25 mg of tissue powder using automated instrumentation (Maxwell 16 Tissue LEV Total RNA Purification Kit, Promega, Madison, WI). RNA concentration was measured by fluorometry (Qubit 2.0 Fluorometer, Life Technologies Corp., Carlsbad, CA) and RNA quality was verified using a microfluidics

platform (Bioanalyzer, Agilent Technologies, Santa Clara, CA).

RNAseq methods. A multiplex library containing all four samples was prepared using the Illumina (San Diego, CA) TruSeq mRNA Sample Preparation Kit v2 with unique indexed adapters (GCCAAT, ACAGTG, CTTGTA, CAGATC). One microgram of total RNA per sample was used as input and the resulting libraries were quantitated using fluorometry. An Illumina HiSeq 2000 instrument was used to generate 100bp paired-end reads (2x100) in one lane of a flow cell.

For the C57BL/6J inbred mice, the mm9 reference was used for alignment. For the F1 animals, we developed a customized RNAseq alignment pipeline tailored to this experiment. Our approach considered these mice as diploid and included two separate alignments that were subsequently merged. This has the advantage of incorporating all known strain-specific genetic variants into the alignment reference sequence to improve alignment quality and to minimize bias caused by differences in genetic distance between the parental genomes to the reference sequence. First, reads from the F1 hybrids were aligned to the appropriate 'pseudogenomes' representing each of the parental genomes using TopHat[6] (v1.4, default parameters including segment length 25 bp, 2 mismatches allowed per segment, 2 mismatches total allowed per 100 bp read, and maximum indel of 3 bases). Pseudogenomes are approximations constructed by incorporating all known SNPs and indels into the C57BL/6 genome (mm9)[7]. We included all variants reported by a large-scale sequencing effort that included 129S1Sv/ImJ and PWK/PhJ[8] (June 2011 release). Second, we mapped coordinates from the pseudogenome aligned reads to mm9 coordinates. This involved updating the alignment positions and rewriting the CIGAR strings of each aligned read[9]. This was necessary as indels alter the pseudogenome coordinates relative to mm9. Third, we annotated each aligned read to indicate the numbers of maternal and paternal alleles (SNPs and indels) observed in a given read and its paired-end mate. Considering the paired-end mates allowed the use of more paired-end reads for ASE. Finally, alignments to maternal and paternal pseudogenomes were merged by computing the proper union of the separate alignments (i.e., the two alignments were combined such that a read aligning to the same position in both alignments was counted once).

E Massively Parallel Computing for IsoDetector

In practice, the penalized estimation in IsoDetector is performed on a grid of λ and τ and the best estimate is chosen according to certain model selection criterion such as BIC. For hypothesis testing purpose, this tuning process has to be done on thousands of bootstrap samples for each gene, which incurs formidable computation burden in real applications. Massively parallel computing based on graphical processing units (GPUs) provides a promising solution. However the coordinate descent based algorithm does not particularly suit the massively parallel computing architecture. Here we propose an algorithm based on the minorization-maximization (MM) principle. Like EM algorithm, MM algorithm always increases the objective value and thus is numerically stable. Furthermore, MM algorithm tends to separate variables, making massively parallel computing feasible in high dimensional optimization problems [10].

Consider the log likelihood of a negative binomial model with response vector $\mathbf{y} \in \mathbb{N}^n$ and design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$

$$\ell(\mathbf{b}, \phi | \mathbf{y}, \mathbf{X}) = \sum_{i=1}^{n} \left[\log \frac{(\phi^{-1})_{(y_i)}}{y_i!} + y_i \log(\phi \mathbf{x}_i^{\mathsf{T}} \mathbf{b}) - (y_i + \phi^{-1}) \log(1 + \phi \mathbf{x}_i^{\mathsf{T}} \mathbf{b}) \right],$$

where ϕ is the overdispersion parameter of negative binomial distribution, and $(\phi^{-1})_{(y_i)}$ denotes the rising factorial $\prod_{k=0}^{y_i - 1} (\phi^{-1} + k) = \phi^{-1}(\phi^{-1} + 1)...(\phi^{-1} + y_i - 1)$. We assume that entries of $\mathbf{X}$ are nonnegative which is true for isoform estimation problem. To simultaneously achieve isoform selection and estimation, IsoDetector relies on log penalized estimation due to its attractive properties. In particular we seek to maximize the penalized objective function

$$f(\mathbf{b}, \phi) = \ell(\mathbf{b}, \phi \mid \mathbf{y}, \mathbf{X}) - \sum_{j=1}^{p} \lambda \log(b_j + \tau), \tag{5}$$

where λ and τ are two tuning parameters, subject to the nonnegativity constraint $b_j \geq 0$.

The derivation of MM algorithm for maximizing (5) relies on simple inequalities [11]. The strategy is to minorize term by term. By concavity of log function,

$$\log(\mathbf{x}_i^{\mathsf{T}} \mathbf{b}) = \log \left(\sum_j x_{ij} b_j^{(t)} \right) \geq \sum_j w_{ij}^{(t)} \log b_j + c^{(t)},$$

where the superscript t indicates iteration number and $c^{(t)}$ is a constant irrelevant to optimization, and

$$w_{ij}^{(t)} = \frac{x_{ij} b_j^{(t)}}{\sum_j x_{ij} b_j^{(t)}}.$$

By the convexity of negative log function, we apply supporting hyperplane inequality to obtain minorizations

$$\begin{aligned} -\log(1 + \phi \mathbf{x}_i^\mathsf{T} \mathbf{b}) &\geq -\frac{\phi \mathbf{x}_i^\mathsf{T} \mathbf{b}}{1 + \phi \mathbf{x}_i^\mathsf{T} \mathbf{b}^{(t)}} + c^{(t)} \\ -\log(b_j + \tau) &\geq -\frac{b_j}{b_j^{(t)} + \tau} + c^{(t)}. \end{aligned}$$

Combining above pieces, we obtain an overall minorization function to the objective function (5)

$$\begin{aligned} & g(\mathbf{b}|\mathbf{b}^{(t)}, \phi^{(t)}) \\ = & \sum_{i=1}^n \left[y_i \sum_j w_{ij}^{(t)} \log b_j - \left(\frac{y_i + (\phi^{(t)})^{-1}}{1 + \phi^{(t)} \mathbf{x}_i^\mathsf{T} \mathbf{b}^{(t)}} \right) \phi^{(t)} \sum_j x_{ij} b_j \right] - \lambda \sum_j \frac{b_j}{b_j^{(t)} + \tau} + c^{(t)} \\ = & \sum_j \left[\left(\sum_i y_i w_{ij}^{(t)} \right) \log b_j - \left(\sum_i \frac{x_{ij}(y_i \phi^{(t)} + 1)}{1 + \phi^{(t)} \mathbf{x}_i^\mathsf{T} \mathbf{b}^{(t)}} + \frac{\lambda}{b_j^{(t)} + \tau} \right) b_j \right] + c^{(t)}. \end{aligned}$$

It is easy to check that

$$\begin{aligned} g(\mathbf{b}|\mathbf{b}^{(t)}, \phi^{(t)}) &\leq f(\mathbf{b}, \phi) \quad \text{for all } \mathbf{b} \\ g(\mathbf{b}^{(t)}|\mathbf{b}^{(t)}, \phi^{(t)}) &= f(\mathbf{b}^{(t)}, \phi^{(t)}). \end{aligned}$$

Therefore maximizing the minorizing function $g(\mathbf{b}|\mathbf{b}^{(t)}, \phi^{(t)})$ always increases the objective function

$$f(\mathbf{b}^{(t+1)}, \phi^{(t)}) \geq g(\mathbf{b}^{(t+1)}|\mathbf{b}^{(t)}, \phi^{(t)}) \geq g(\mathbf{b}^{(t)}|\mathbf{b}^{(t)}, \phi^{(t)}) = f(\mathbf{b}^{(t)}, \phi^{(t)}).$$

Setting derivative of $g(\mathbf{b}|\mathbf{b}^{(t)}, \phi^{(t)})$ to zero yields an extremely simple update

$$b_j^{(t+1)} = \frac{\sum_i y_i w_{ij}^{(t)}}{\sum_i \frac{x_{ij}(y_i \phi^{(t)} + 1)}{1 + \phi^{(t)} \mathbf{x}_i^\mathsf{T} \mathbf{b}^{(t)}} + \frac{\lambda}{b_j^{(t)} + \tau}}, \quad j = 1, \ldots, p, \tag{6}$$

involving only trivial algebra. All parameters b_j are separated and thus updated simultaneously, matching perfectly with the massively parallel architecture of GPUs. The nonnegativity constraints $b_j \geq 0$ are also preserved in the update (6). Whenever $b_j^{(0)}$ are positive, all subsequent iterates $b_j^{(t)}$ will always be nonnegative. Effects of the tuning parameters λ and τ are clear: large λ and small τ cause more shrinkage and vice versa. Furthermore, the log penalty penalizes small b_j more heavily than large b_j, a desired property the lasso penalty lacks.

The update (6) always increases the objective value. However, it does not update the overdispersion parameter ϕ. In practice, we update ϕ after every a few (e.g., five) updates of $\mathbf{b}$ by (6). Updating of ϕ can be done by either Newton's steps or by invoking MM algorithm again. Both are simple because it is a smooth univariate optimization problem. For brevity the details are omitted here.

F Supplementary Figures

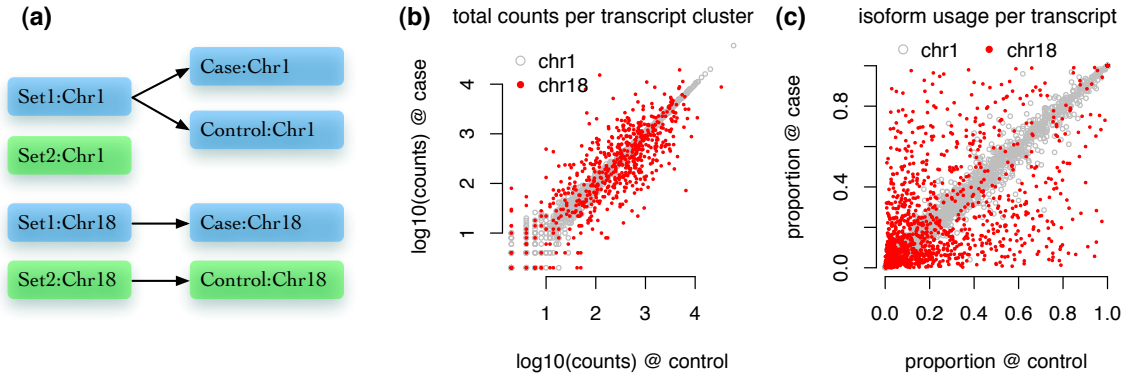

Figure 2: A summary of simulated data. We first simulated ∼2 million 76+76bps paired-end reads for data set 1 and data set 2, based on the transcriptome annotation of chromosome 1 and 18 of mouse genome. The expression of any gene/transcript are independent between data set 1 and data set 2. Then as illustrated in **(a)**, a case and a control sample were generated as follows. For chromosome 1, the sequence fragments of simulation set 1 were randomly split into the case and control samples. For chromosome 18, half of the sequence fragments from set 1 were selected for case and half of the sequence fragments from set 2 were selected for control. Therefore, comparing case and control, all the transcripts in chromosome 1 were equivalently expressed and all the transcripts in chromosome 18 were differentially expressed, either in terms of total expression **(b)** or isoform usage **(c)**. **(a)** Comparison of the total number of fragments per transcript cluster between the case and the control samples. **(c)** Comparison of isoform usage of each transcript between the case and the control samples. Here isoform usage is quantified by the ratio of the number of sequence fragments of one transcript over the total number of fragments of the corresponding transcript cluster.

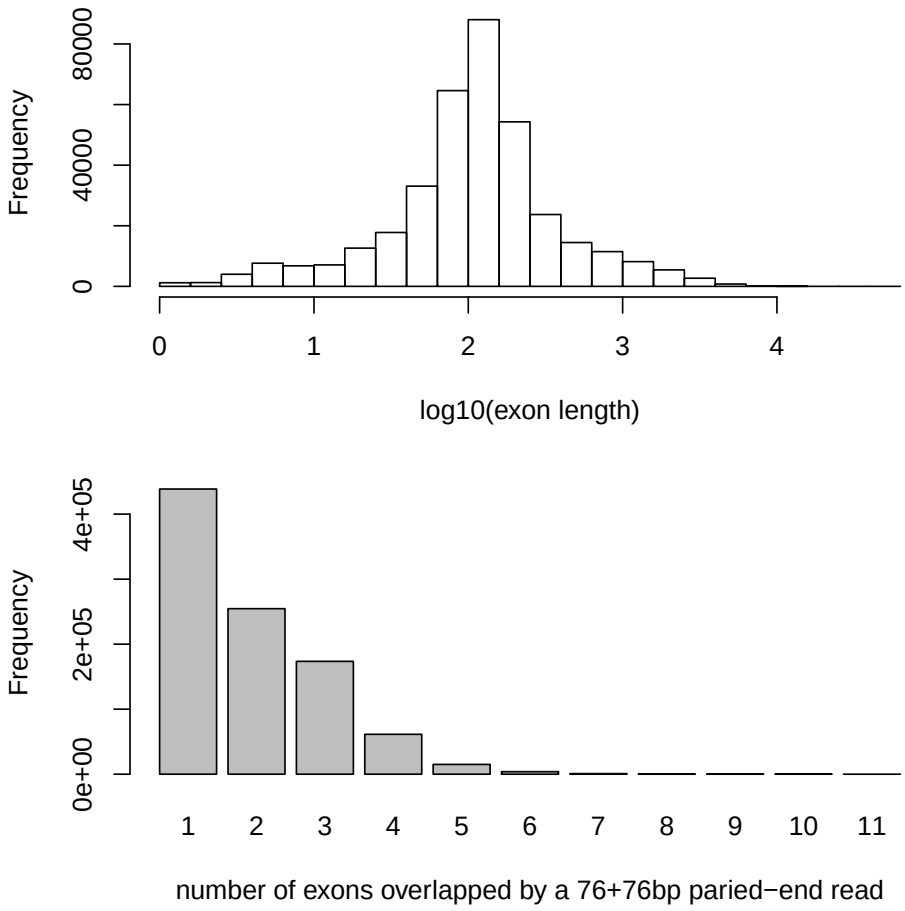

Figure 3: Upper panel shows the distribution of the lengths of non-overlapping exons. Lower panel shows the distribution of the number of exons overlapped by each paired end read. A paired-end read overlaps an exon if at least one base pair of either end of the read overlap with the exon. About 46%, 27%, and 18% of the reads overlap with only one, two, or three exons.

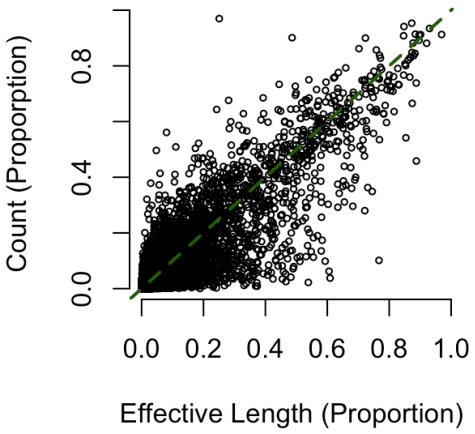

Figure 4: The relation between the effective length of an exon set (divided by the total effective length of the transcript cluster to which the exon belongs) and the proportion of RNA-seq fragments mapped to this exon set in our simulated data. The correlation between them is 0.88. Because different transcript clusters have different expression abundance, we compared read count and effective length as the proportions over the corresponding transcript cluster. Note that the effective length is calculated while assuming all the exons in an exon set are contiguous, which may not be true. Therefore the results here can only be viewed as an approximation.

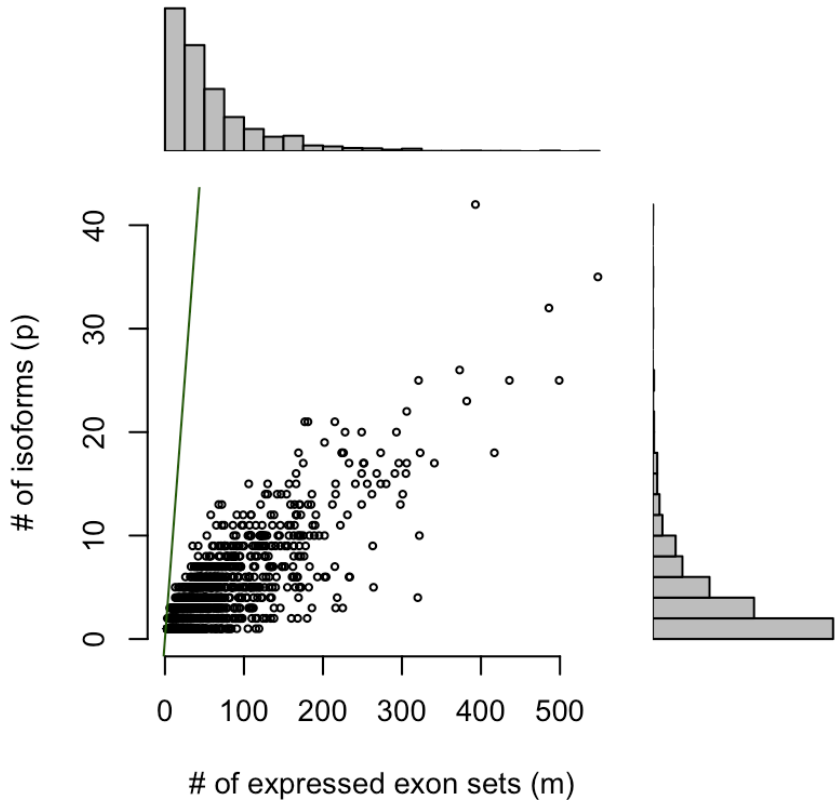

Figure 5: An illustration of the dimension of the isoform selection problem when we use known transcriptome annotation. For each transcript cluster, we consider a variable selection problem where sample size n is the number of expressed exon sets, and the number of covariates p is the number of (candidate) isoforms. The solid line indicates $p = n$.

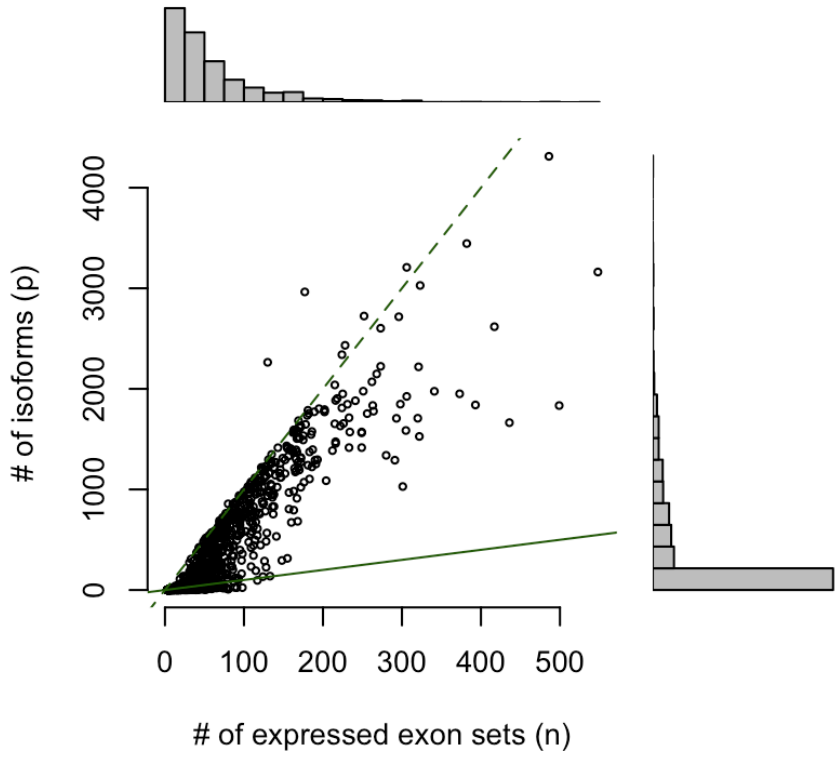

Figure 6: An illustration of the dimension of the isoform selection problem when there is no isoform annotation. For each gene (or transcript cluster), we consider a variable selection problem where sample size n is the number of expressed exon sets, and the number of covariates p is the number of (candidate) isoforms. The solid line indicates $p = n$, and the broken line indicates $p = 10n$. In our implementation, we choose the number of candidate isoforms so that $p \leq 10n$ approximately. Users can loose this restriction with price of increasing computational cost. Our experience is that IsoDetector runs well for $p \leq 100n$.

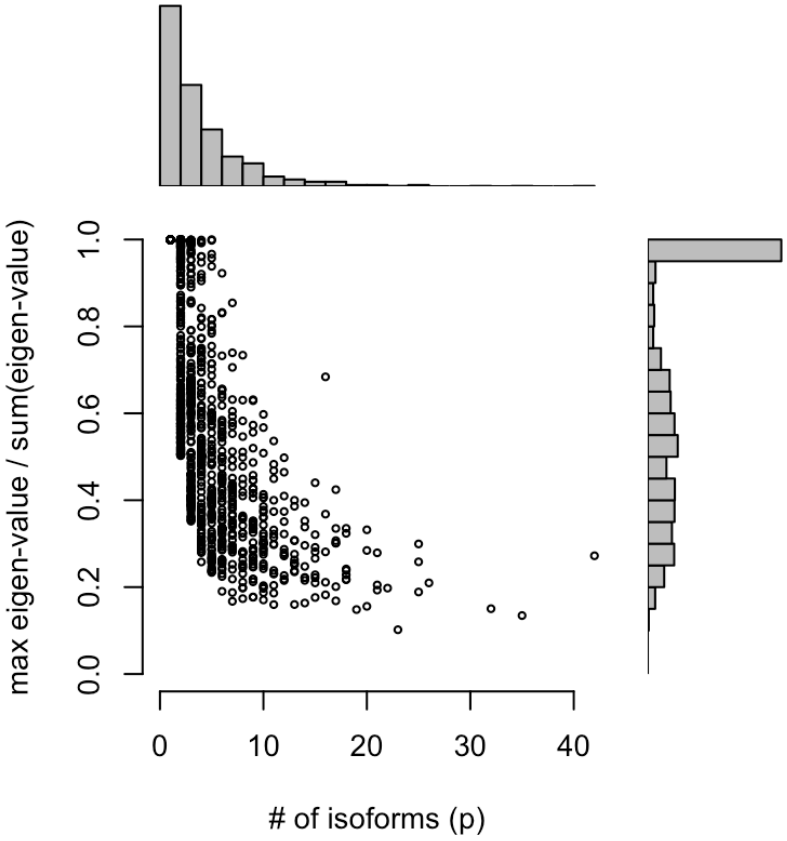

Figure 7: An illustration of the correlation among the isoforms of each transcript cluster when we use known isoform annotation. Each point indicates a transcript cluster where x-axis is the number of isoforms and y-axis is the proportion of variance explained by the first principal component.

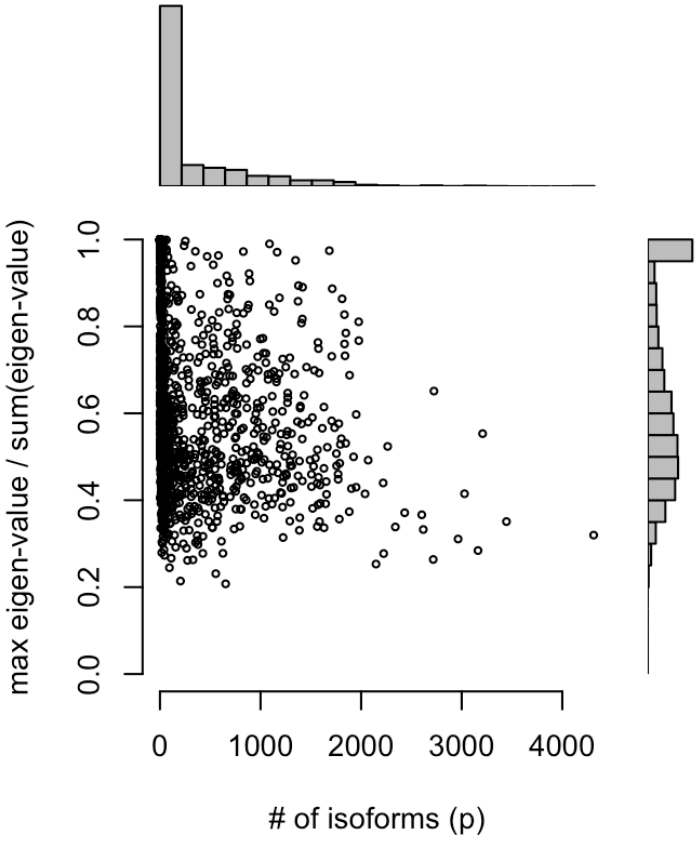

Figure 8: An illustration of the correlation among the isoforms of each transcript cluster when there is no isoform annotation. Each point indicates a transcript cluster where x-axis is the number of isoforms and y-axis is the proportion of variance explained by the first principal component.

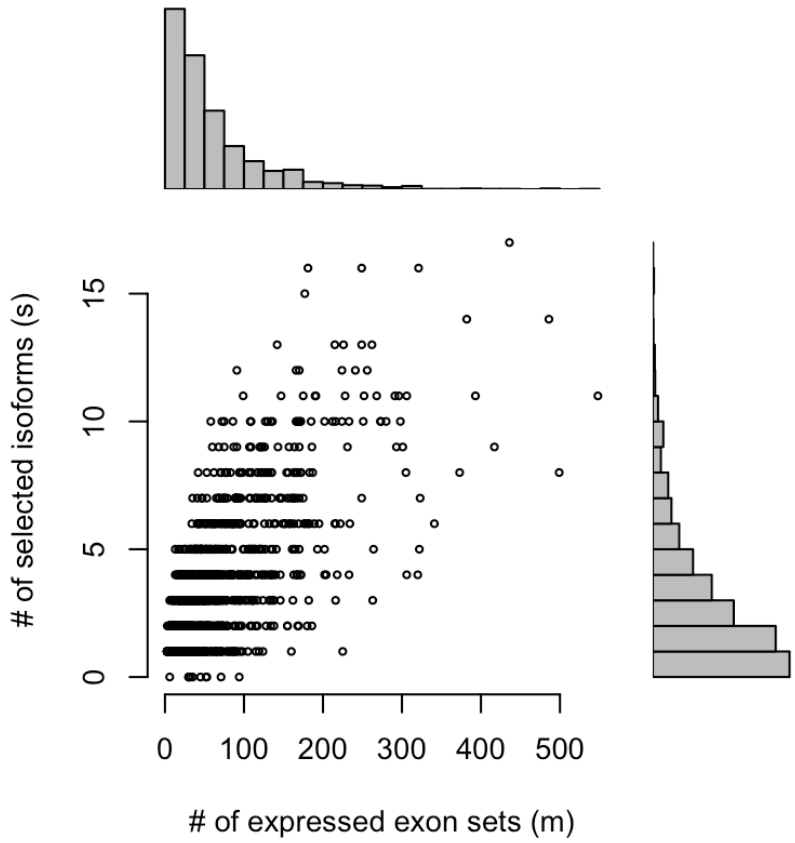

Figure 9: An illustration of the number of isoforms selected by IsoDetector when we choose the candidate isoforms based on the known isoform annotation.

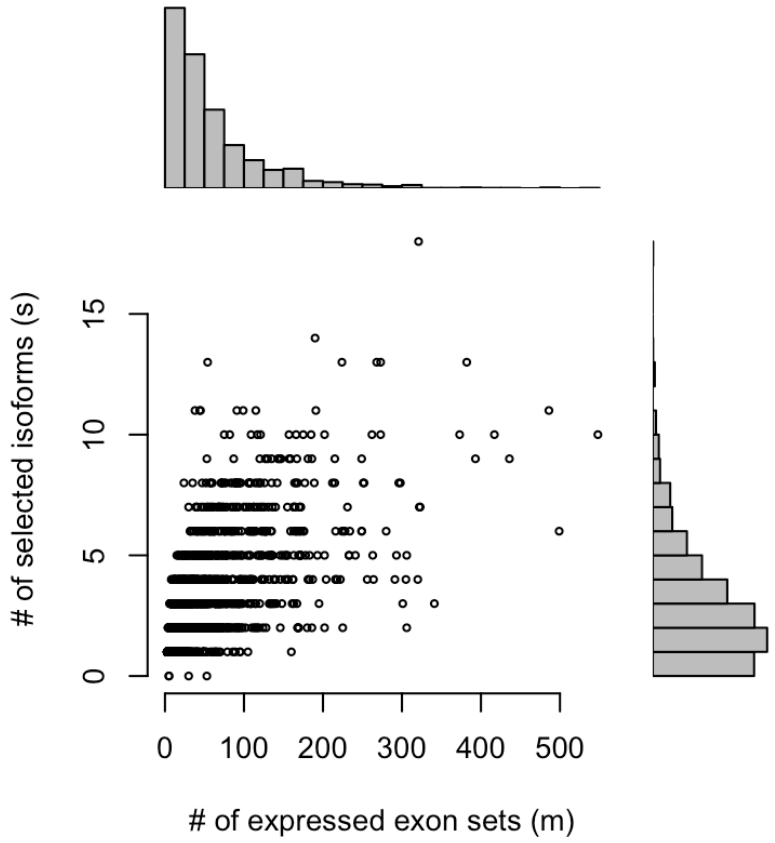

Figure 10: An illustration of the number of isoforms selected by IsoDetector when we choose the candidate isoforms without using any isoform annotation.

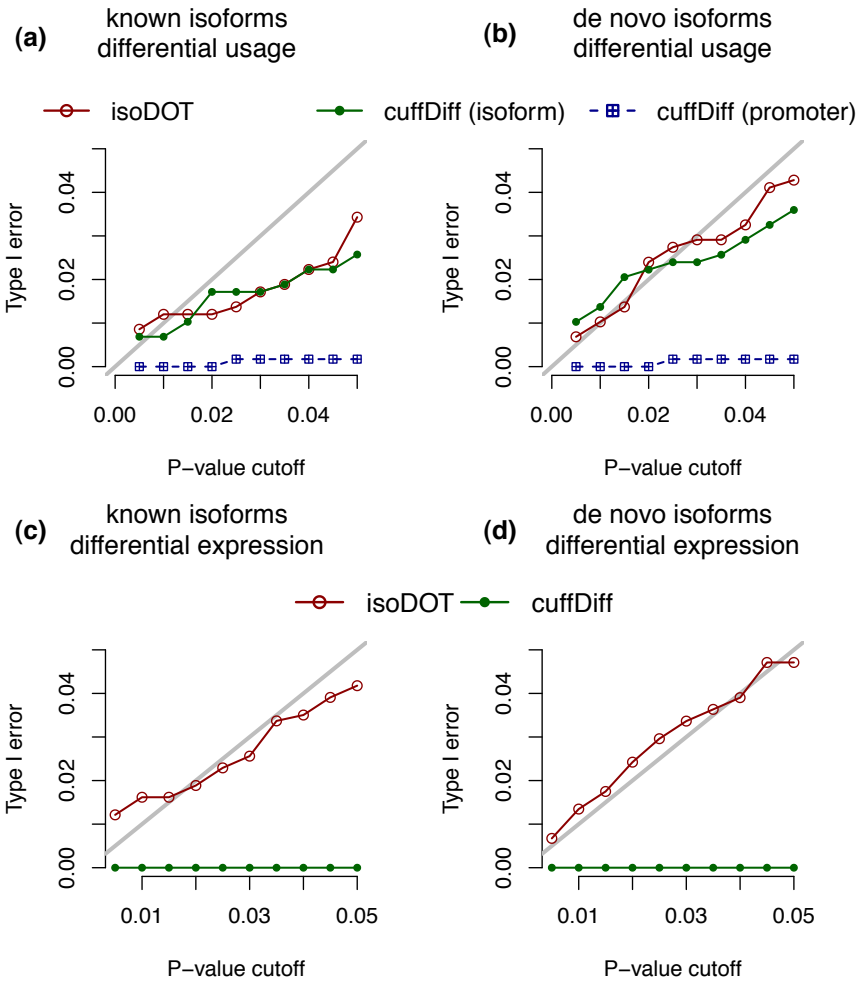

Figure 11: Compare the type I error of IsoDOT and Cuffdiff2 for detesting genes with differential isoform usage (a-b) or differential expression (c-d), while transcriptome annotation is known (a,c) or not (b,d). For the case of differential isoform usage, cufflinks provides results for "isoform" and "promoter", where the former is for isoform sharing a TSS, and the latter is for differential usage of TSSs. For the "isoform" case, we have collapsed the p-values of multiple tests of a gene by taking minimum, thus it leads to an over-estimate of type I error.

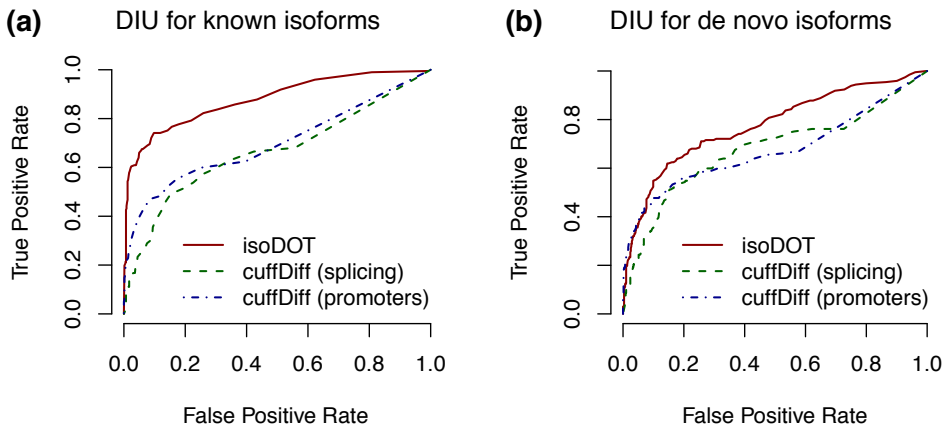

Figure 12: Compare the results of IsoDOT and Cuffdiff2 by ROC curves. The ROC curves compare two methods across a wide range of p-value cutoffs. If one method has a calibration issue (e.g., p-value is larger than it should be) but still ranks the genes correctly, it would perform well judged by ROC curve. The results shown here demonstrate that the IsoDOT still performs better than Cuffdiff2 even if we allow the results of Cuffdiff2 to be calibrated.

Category	Term	Count	List Total	Pop Hits	Pop Total	Bonferroni
GOTERM_CC_FAT	GO:0043005~neuron projection	12	71	245	12504	1.78E-05
GOTERM_CC_FAT	GO:0042995~cell projection	14	71	575	12504	2.66E-03
GOTERM_CC_FAT	GO:0030424~axon	6	71	107	12504	5.53E-02

Figure 13: DAVID functional category enrichments for DIU genes (with transcriptome annotation).

Category	Term	Count	List Total	Pop Hits	Pop Total	Bonferroni
GOTERM_BP_FAT	GO:0006470~protein amino acid dephosphorylation	5	75	114	13588	9.18E-01
GOTERM_BP_FAT	GO:0016311~dephosphorylation	5	75	141	13588	9.95E-01
GOTERM_MF_FAT	GO:0004721~phosphoprotein phosphatase activity	5	74	152	13288	8.95E-01
GOTERM_MF_FAT	GO:0004725~protein tyrosine phosphatase activity	4	74	101	13288	9.85E-01
GOTERM_MF_FAT	GO:0016791~phosphatase activity	5	74	238	13288	1.00E+00

Figure 14: DAVID functional category enrichments for DIE genes (with transcriptome annotation).

Category	Term	Count	List Total	Pop Hits	Pop Total	Bonferroni
GOTERM_MF_FAT	GO:0030955~potassium ion binding	5	65	118	13288	3.77E-01
GOTERM_BP_FAT	GO:0006813~potassium ion transport	5	68	160	13588	9.97E-01
GOTERM_MF_FAT	GO:0005249~voltage-gated potassium channel activity	4	65	99	13288	9.00E-01
GOTERM_BP_FAT	GO:0015672~monovalent inorganic cation transport	6	68	303	13588	1.00E+00
GOTERM_MF_FAT	GO:0031420~alkali metal ion binding	5	65	206	13288	9.64E-01
GOTERM_MF_FAT	GO:0022843~voltage-gated cation channel activity	4	65	128	13288	9.90E-01

Figure 15: DAVID functional category enrichments for DIU genes (without transcriptome annotation).

Category	Term	Count	List Total	Pop Hits	Pop Total	Bonferroni
GOTERM_BP_FAT	GO:0016477~cell migration	6	76	240	13588	9.99E-01
GOTERM_BP_FAT	GO:0051674~localization of cell	6	76	284	13588	1.00E+00
GOTERM_BP_FAT	GO:0048870~cell motility	6	76	284	13588	1.00E+00
GOTERM_BP_FAT	GO:0006928~cell motion	6	76	367	13588	1.00E+00

Figure 16: DAVID functional category enrichments for DIE genes (without transcriptome annotation).

cuffdiff_knownIso_promoter_top100_DAVID

Category	Term	Count	List Total	Pop Hits	Pop Total	Bonferroni
GOTERM_MF_FAT	GO:0003677~DNA binding	21	76	1781	13288	2.7E-01
GOTERM_BP_FAT	GO:0045449~regulation of transcription	21	69	2227	13588	9.5E-01
GOTERM_BP_FAT	GO:0006350~transcription	18	69	1772	13588	9.6E-01
GOTERM_MF_FAT	GO:0003700~transcription factor activity	11	76	776	13288	8.9E-01
GOTERM_BP_FAT	GO:0006355~regulation of transcription, DNA-dependent	15	69	1465	13588	1.0E+00
GOTERM_BP_FAT	GO:0051252~regulation of RNA metabolic process	15	69	1488	13588	1.0E+00

cuffdiff_knownIso_splicing_top100_DAVID

Category	Term	Count	List Total	Pop Hits	Pop Total	Bonferroni
GOTERM_BP_FAT	GO:0006650~glycerophospholipid metabolic process	4	67	88	13588	1.0E+00
GOTERM_BP_FAT	GO:0046486~glycerolipid metabolic process	4	67	129	13588	1.0E+00
GOTERM_BP_FAT	GO:0030384~phosphoinositide metabolic process	3	67	63	13588	1.0E+00
GOTERM_BP_FAT	GO:0006644~phospholipid metabolic process	4	67	163	13588	1.0E+00
GOTERM_BP_FAT	GO:0019637~organophosphate metabolic process	4	67	176	13588	1.0E+00

cuffdiff_unknownIso_promoter_top100_DAVID

Category	Term	Count	List Total	Pop Hits	Pop Total	Bonferroni
GOTERM_MF_FAT	GO:0005083~small GTPase regulator activity	7	75	223	13288	2.7E-01
GOTERM_BP_FAT	GO:0051336~regulation of hydrolase activity	6	69	196	13588	8.0E-01
GOTERM_MF_FAT	GO:0005099~Ras GTPase activator activity	4	75	65	13288	7.0E-01
GOTERM_BP_FAT	GO:0043087~regulation of GTPase activity	4	69	93	13588	1.0E+00
GOTERM_BP_FAT	GO:0046578~regulation of Ras protein signal transduction	5	69	181	13588	1.0E+00
GOTERM_MF_FAT	GO:0030695~GTPase regulator activity	7	75	361	13288	9.6E-01

cuffdiff_unknownIso_splicing_top100_DAVID

Category	Term	Count	List Total	Pop Hits	Pop Total	Bonferroni
GOTERM_BP_FAT	GO:0000902~cell morphogenesis	6	63	309	13588	1.0E+00
GOTERM_BP_FAT	GO:0048858~cell projection morphogenesis	5	63	202	13588	1.0E+00
GOTERM_BP_FAT	GO:0030030~cell projection organization	6	63	319	13588	1.0E+00
GOTERM_BP_FAT	GO:0032990~cell part morphogenesis	5	63	212	13588	1.0E+00
GOTERM_BP_FAT	GO:0032989~cellular component morphogenesis	6	63	351	13588	1.0E+00
GOTERM_BP_FAT	GO:0048812~neuron projection morphogenesis	4	63	176	13588	1.0E+00

Figure 17: DAVID functional category enrichments for the results from Cuffldiff.

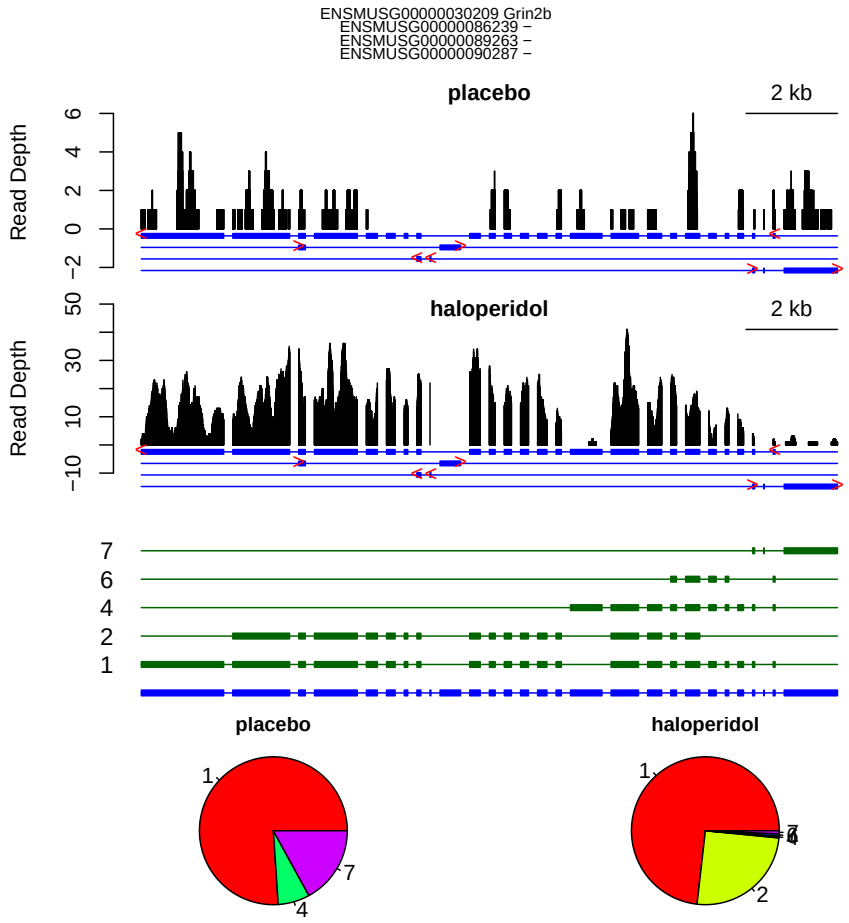

Figure 18: Differential isoform usage of gene Grin2b between two C57BL/6 mice with haloperidal or placebo treatment. Note Grin2b belongs to a transcript cluster with four genes. However, the other three genes are short and contribute little if any signal of differential isoform usage.

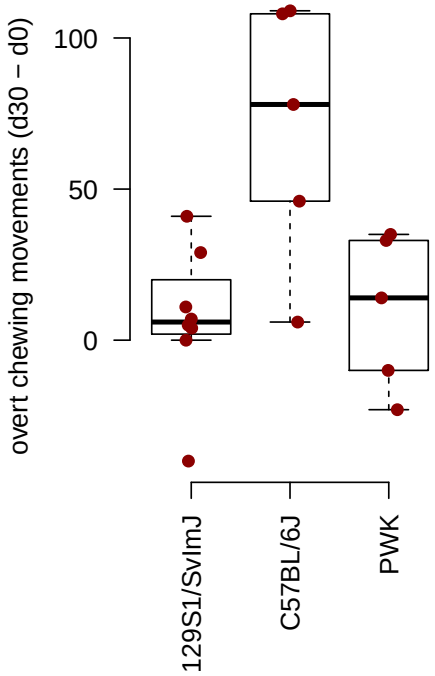

Figure 19: Change of overt chewing movements for three inbred strains between day 0 and day 30 after haloperidol treatment. See Crowley et al. [12] for more details of the experiment and the results of other phenotypic outcomes.

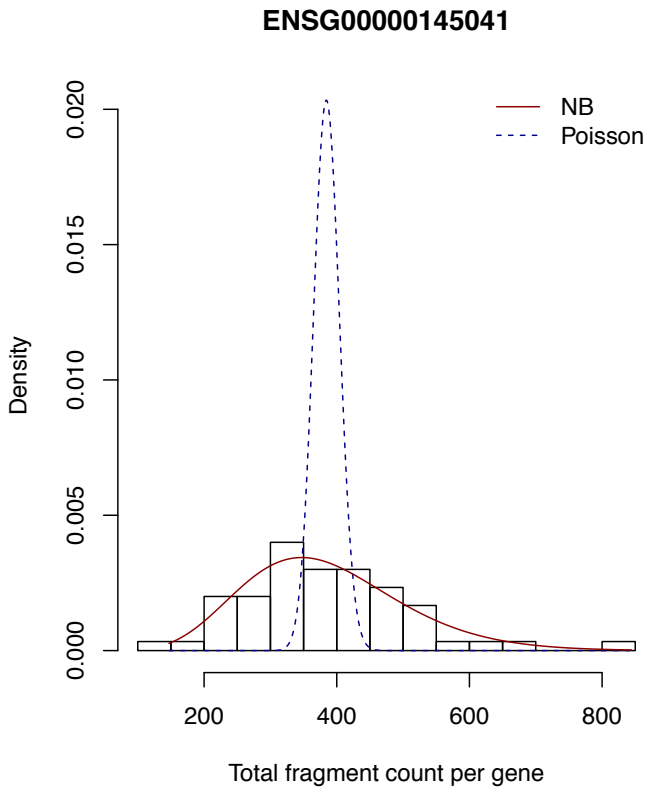

Figure 20: An example to show that negative binomial distribution can provide adequate fit to RNA-seq fragment count data whereas Poisson distribution assumption leads to severe underestimate of variance. The RNA-seq data used in this example are the RNA-seq fragment count for gene VPRBP (Vpr (HIV-1) binding protein, ensembl ID: ENSG00000145041) from 50 HapMap CEU samples [13]. The MLE of the two distributions were obtained using R function glm and glm.nb, respectively, after correction for read-depth.

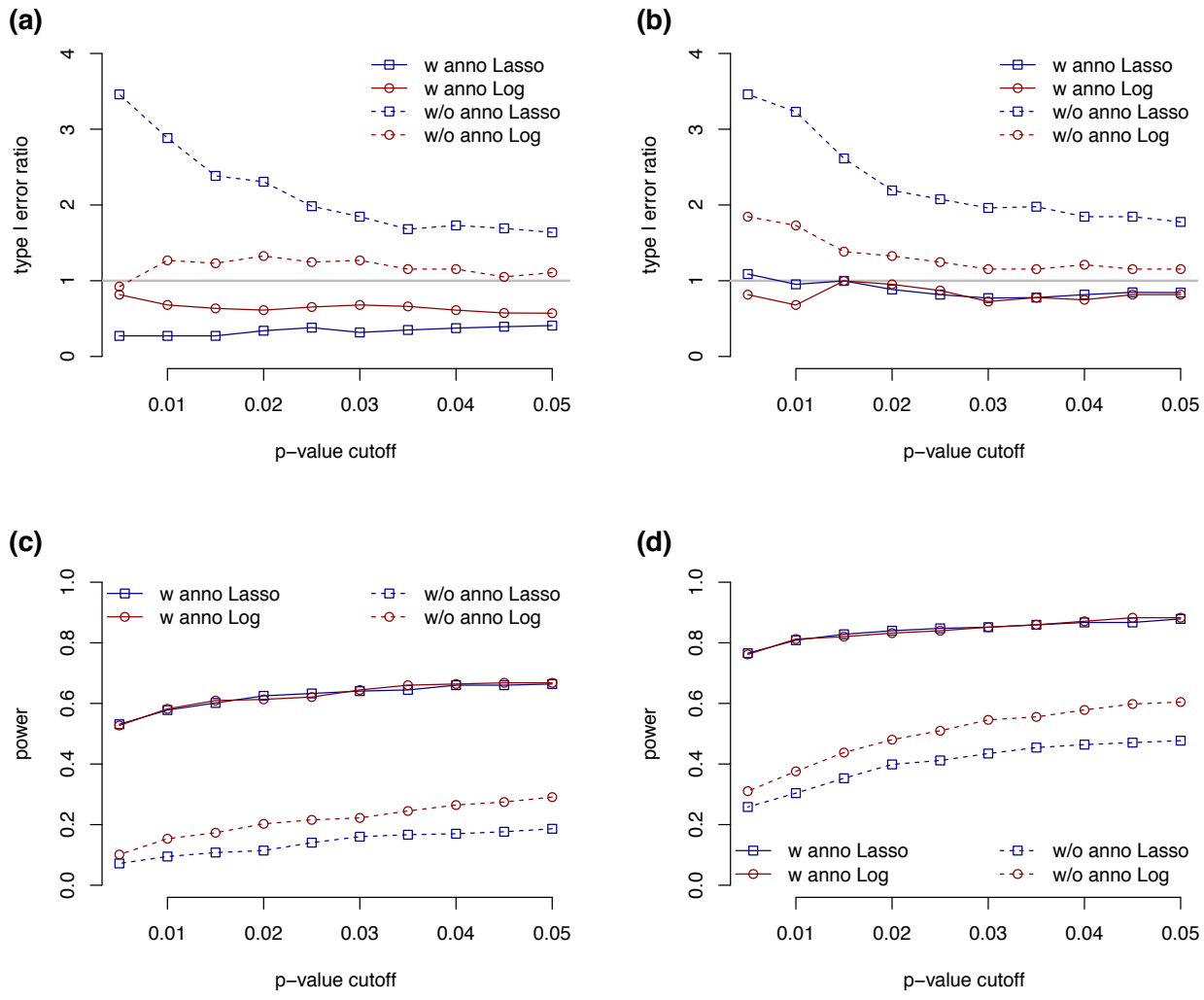

Figure 21: Compare type I error and power when we use the Lasso penalty or the Log penalty in IsoDOT. (a) Type I for DIU test. (b) Type I error for DIE test. (c) Power for DIU test. (d) Power for DIE test. In panel (a) and (b), the y-axis is type I error ratio, which is the ratio of observed type I error rate divided by the corresponding p-value cutoff (x-axis), which is the expected type I error rate.

Table 1: Read depth of the RNA-seq data in mouse haloperidol treatment study. Each sequence fragment was sequenced on both ends by 93-100bp. The first four rows show the information of four mince and the last four rows are for allele-specific RNA-seq reads from the two F1 mice.

Sample ID	Genetic background	Treatment	Total number of mapped reads	Number of fragments passed QC and mapped to exonic regions
BB1050	C57BL/6J	placebo	21,482,924	8,337,872
BB1068	C57BL/6J	haloperidol	27,178,749	10,486,170
CG0069	129×PWK	placebo	24,014,041	10,476,460
CG0077	129×PWK	haloperidol	20,365,336	8,871,864
CG0069	129 @ 129×PWK	placebo	4,667,545	1,953,335
CG0069	PWK @ 129×PWK	placebo	4,605,879	1,931,791
CG0077	129 @ 129×PWK	haloperidol	3,993,348	1,668,243
CG0077	PWK @ 129×PWK	haloperidol	3,957,371	1,654,705

Table 2: Top 100 genes identified from differential isoform usage (DU only) analysis comparing two C57BL/6J mice with haloperidol or placebo treatments.

Ensembl ID	**symbol**	**name**
ENSMUSG00000040537	Adam22	a disintegrin and metallopeptidase domain 22
ENSMUSG00000020431	Adcy1	adenylate cyclase 1
ENSMUSG00000049470	Aff4	AF4/FMR2 family, member 4
ENSMUSG00000061603	Akap6	A kinase (PRKA) anchor protein 6
ENSMUSG00000040407	Akap9	A kinase (PRKA) anchor protein (yotiao) 9
ENSMUSG00000069601	Ank3	ankyrin 3, epithelial
ENSMUSG00000071176	Arhgef10	Rho guanine nucleotide exchange factor (GEF) 10
ENSMUSG00000059495	Arhgef12	Rho guanine nucleotide exchange factor (GEF) 12
ENSMUSG00000002343	Armc6	armadillo repeat containing 6
ENSMUSG00000020788	Atp2a3	ATPase, Ca++ transporting, ubiquitous
ENSMUSG00000003604	Aven	apoptosis, caspase activation inhibitor
ENSMUSG00000048251	Bcl11b	B-cell leukemia/lymphoma 11B
ENSMUSG00000049658	Bdp1	B double prime 1, subunit of RNA polymerase III transcription initiation factor IIIB
ENSMUSG00000042460	C1galt1	core 1 synthase, glycoprotein-N-acetylgalactosamine 3-beta-galactosyltransferase, 1

Continued on next page

Table 2 – *Continued from previous page*

Ensembl ID	**symbol**	**name**
ENSMUSG00000039983	Ccdc32	coiled-coil domain containing 32
ENSMUSG00000033671	Cep350	centrosomal protein 350
ENSMUSG00000021097	Clmn	calmin
ENSMUSG00000060924	Csmd1	CUB and Sushi multiple domains 1
ENSMUSG00000048796	Cyb561d1	cytochrome b-561 domain containing 1
ENSMUSG00000017999	Ddx27	DEAD (Asp-Glu-Ala-Asp) box polypeptide 27
ENSMUSG00000037426	Depdc5	DEP domain containing 5
ENSMUSG00000024456	Diap1	diaphanous homolog 1 (Drosophila)
ENSMUSG00000045103	Dmd	dystrophin, muscular dystrophy
ENSMUSG00000041268	Dmxl2	Dmx-like 2
ENSMUSG00000039716	Dock3	dedicator of cyto-kinesis 3
ENSMUSG00000036270	Edc4	enhancer of mRNA decapping 4
ENSMUSG00000028760	Eif4g3	eukaryotic translation initiation factor 4 gamma, 3
ENSMUSG00000039167	Eltd1	EGF, latrophilin seven transmembrane domain containing 1
ENSMUSG00000004267	Eno2	enolase 2, gamma neuronal
ENSMUSG00000032314	Etfa	electron transferring flavoprotein, alpha polypeptide
ENSMUSG00000010517	Faf1	Fas-associated factor 1
ENSMUSG00000025262	Fam120c	family with sequence similarity 120, member C
ENSMUSG00000025153	Fasn	fatty acid synthase
ENSMUSG00000070733	Fryl	furry homolog-like (Drosophila)
ENSMUSG00000039801	Gm5906	RIKEN cDNA 2410089E03 gene
ENSMUSG00000031210	Gpr165	G protein-coupled receptor 165
ENSMUSG00000020176	Grb10	growth factor receptor bound protein 10
ENSMUSG00000030209	Grin2b	glutamate receptor, ionotropic, NMDA2B (epsilon 2)
ENSMUSG00000031584	Gsr	glutathione reductase
ENSMUSG00000006930	Hap1	huntingtin-associated protein 1
ENSMUSG00000029104	Htt	huntingtin
ENSMUSG00000009828	Ick	intestinal cell kinase
ENSMUSG00000023830	Igf2r	insulin-like growth factor 2 receptor
ENSMUSG00000042599	Jhdm1d	jumonji C domain-containing histone demethylase 1 homolog D (S. cerevisiae)
ENSMUSG00000024410	K100	RIKEN cDNA 3110002H16 gene
ENSMUSG00000016946	Kctd5	potassium channel tetramerisation domain containing 5
ENSMUSG00000063077	Kif1b	kinesin family member 1B
ENSMUSG00000027550	Lrrcc1	leucine rich repeat and coiled-coil domain containing 1
ENSMUSG00000028649	Macf1	microtubule-actin crosslinking factor 1
ENSMUSG00000036278	Macrod1	MACRO domain containing 1

Continued on next page

Table 2 – *Continued from previous page*

Ensembl ID	**symbol**	**name**
ENSMUSG00000008763	Man1a2	mannosidase, alpha, class 1A, member 2
ENSMUSG00000059474	Mbtd1	mbt domain containing 1
ENSMUSG00000020184	Mdm2	transformed mouse 3T3 cell double minute 2
ENSMUSG00000024294	Mib1	mindbomb homolog 1 (Drosophila)
ENSMUSG00000038056	Mll3	myeloid/lymphoid or mixed-lineage leukemia 3
ENSMUSG00000022889	Mrpl39	mitochondrial ribosomal protein L39
ENSMUSG00000033004	Mycbp2	MYC binding protein 2
ENSMUSG00000030739	Myh14	myosin, heavy polypeptide 14
ENSMUSG00000034593	Myo5a	myosin VA
ENSMUSG00000027799	Nbea	neurobeachin
ENSMUSG00000020716	Nf1	neurofibromatosis 1
ENSMUSG00000038495	Otud7b	OTU domain containing 7B
ENSMUSG00000021140	Pcnx	pecanex homolog (Drosophila)
ENSMUSG00000002265	Peg3	paternally expressed 3
ENSMUSG00000028085	Pet112l	PET112-like (yeast)
ENSMUSG00000039943	Plcb4	phospholipase C, beta 4
ENSMUSG00000032827	Ppp1r9a	protein phosphatase 1, regulatory (inhibitor) subunit 9A
ENSMUSG00000038976	Ppp1r9b	protein phosphatase 1, regulatory subunit 9B
ENSMUSG00000003099	Ppp5c	protein phosphatase 5, catalytic subunit
ENSMUSG00000039410	Prdm16	PR domain containing 16
ENSMUSG00000030465	Psd3	pleckstrin and Sec7 domain containing 3
ENSMUSG00000038764	Ptpn3	protein tyrosine phosphatase, non-receptor type 3
ENSMUSG00000053141	Ptprt	protein tyrosine phosphatase, receptor type, T
ENSMUSG00000068748	Ptprz1	protein tyrosine phosphatase, receptor type Z, polypeptide 1
ENSMUSG00000037098	Rab11fip3	RAB11 family interacting protein 3 (class II)
ENSMUSG00000027652	Ralgapb	Ral GTPase activating protein, beta subunit (non-catalytic)
ENSMUSG00000075376	Rc3h2	ring finger and CCCH-type zinc finger domains 2
ENSMUSG00000042453	Reln	reelin
ENSMUSG00000050310	Rictor	RPTOR independent companion of MTOR, complex 2
ENSMUSG00000020448	Rnf185	ring finger protein 185
ENSMUSG00000038685	Rtel1	regulator of telomere elongation helicase 1
ENSMUSG00000021313	Ryr2	ryanodine receptor 2, cardiac
ENSMUSG00000075318	Scn2a1	sodium channel, voltage-gated, type II, alpha 1
ENSMUSG00000028064	Sema4a	sema domain, immunoglobulin domain (Ig), transmembrane domain (TM) and short cytoplasmic domain, (semaphorin) 4A

Continued on next page

Table 2 – *Continued from previous page*

Ensembl ID	**symbol**	**name**
ENSMUSG00000005089	Slc1a2	solute carrier family 1 (glial high affinity glutamate transporter), member 2
ENSMUSG00000023032	Slc4a8	solute carrier family 4 (anion exchanger), member 8
ENSMUSG00000019769	Syne1	synaptic nuclear envelope 1
ENSMUSG00000062542	Syt9	synaptotagmin IX
ENSMUSG00000053580	Tanc2	tetratricopeptide repeat, ankyrin repeat and coiled-coil containing 2
ENSMUSG00000023923	Tbc1d5	TBC1 domain family, member 5
ENSMUSG00000039230	Tbcd	tubulin-specific chaperone d
ENSMUSG00000032186	Tmod2	tropomodulin 2
ENSMUSG00000009470	Tnpo1	transportin 1
ENSMUSG00000019820	Utrn	utrophin
ENSMUSG00000046230	Vps13a	vacuolar protein sorting 13A (yeast)
ENSMUSG00000045962	Wnk1	WNK lysine deficient protein kinase 1
ENSMUSG00000047694	Yipf6	Yip1 domain family, member 6
ENSMUSG00000020812		RIKEN cDNA 1810032O08
ENSMUSG00000053081		RIKEN cDNA 1700069B07
ENSMUSG00000072847		RIKEN cDNA A530017D24

Table 3: 23 genes with differential isoform usage (DU only p-value < 0.01) between the two alleles of the haloperidol treated F1(129×PWK) mouse, but no differential isoform usage (DU only p-value > 0.1) comparing the two alleles of the placebo treated F1(129×PWK) mouse.

Ensembl ID	**symbol**	**name**
ENSMUSG00000006638	Abhd1	abhydrolase domain containing 1
ENSMUSG00000005686	Ampd3	adenosine monophosphate deaminase 3
ENSMUSG00000004446	Bid	BH3 interacting domain death agonist
ENSMUSG00000022617	Chkb	choline kinase beta
ENSMUSG00000026816	Gtf3c5	general transcription factor IIIC, polypeptide 5
ENSMUSG00000031787	Katnb1	katanin p80 (WD40-containing) subunit B 1
ENSMUSG00000058740	Kcnt1	potassium channel, subfamily T, member 1
ENSMUSG00000039682	Lap3	leucine aminopeptidase 3
ENSMUSG00000026792	Lrsam1	RIKEN cDNA 4930555K19
ENSMUSG00000024085	Man2a1	mannosidase 2, alpha 1
ENSMUSG00000029822	Osbpl3	oxysterol binding protein-like 3
ENSMUSG00000021846	Peli2	pellino 2
ENSMUSG00000033628	Pik3c3	phosphoinositide-3-kinase, class 3
ENSMUSG00000005225	Plekha8	pleckstrin homology domain containing, family A (phosphoinositide binding specific) member 8
ENSMUSG00000026035	Ppil3	peptidylprolyl isomerase (cyclophilin)-like 3
ENSMUSG00000036202	Rif1	Rap1 interacting factor 1 homolog (yeast)
ENSMUSG00000001054	Rmnd5b	required for meiotic nuclear division 5 homolog B (S. cerevisiae)
ENSMUSG00000052656	Rnf103	ring finger protein 103
ENSMUSG00000027273	Snap25	synaptosomal-associated protein 25
ENSMUSG00000043079	Synpo	synaptopodin
ENSMUSG00000040389	Wdr47	WD repeat domain 47
ENSMUSG00000001017		RIKEN cDNA 2500003M10
ENSMUSG00000044600		RIKEN cDNA 9130011J15

References

[1] Friedman, J., Hastie, T., and Tibshirani, R. (2010) Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software,* **33**(1), 1–22.

[2] Sun, W., Ibrahim, J., and Zou, F. (2010) Genomewide Multiple-Loci Mapping in Experimental Crosses by Iterative Adaptive Penalized Regression. *Genetics,* **185**(1), 349.

[3] Chen, J. and Chen, Z. (2012) Extended BIC for small-n-large-p sparse glm. *Statistica Sinica,* **22**(2), 555.

[4] Fleischmann, N., Christ, G., Sclafani, T., and Melman, A. (2002) The effect of ovariectomy and long-term estrogen replacement on bladder structure and function in the rat. *The Journal of urology,* **168**(3), 1265–1268.

[5] Hsin-Tung, E. and Simpson, G. (2000) Medication-induced movement disorders. *Kaplan and Sadocks Comprehensive Text Book of Psychiatry. Baltimore, Maryland: Williams and Wilkins,* pp. 2265–2271.

[6] Trapnell, C., Pachter, L., and Salzberg, S. (2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics,* **25**(9), 1105.

[7] Church, D., Goodstadt, L., Hillier, L., Zody, M., Goldstein, S., She, X., Bult, C., Agarwala, R., Cherry, J., DiCuccio, M., et al. (2009) Lineage-specific biology revealed by a finished genome assembly of the mouse. *PLoS biology,* **7**(5), e1000112.

[8] Keane, T., Goodstadt, L., Danecek, P., White, M., Wong, K., Yalcin, B., Heger, A., Agam, A., Slater, G., Goodson, M., et al. (2011) Mouse genomic variation and its effect on phenotypes and gene regulation. *Nature,* **477**(7364), 289–294.

[9] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., et al. (2009) The sequence alignment/map format and samtools. *Bioinformatics,* **25**(16), 2078–2079.

[10] Zhou, H., Lange, K., and Suchard, M. (2010) Graphical processing units and high-dimensional optimization. *Statistical Science,* **25**, 311–324.

[11] Zhou, H. and Lange, K. (2010) MM algorithms for some discrete multivariate distributions. *Journal of Computational and Graphical Statistics,* **19**, 645–665.

[12] Crowley, J., Adkins, D., Pratt, A., Quackenbush, C., van denOord, E., Moy, S., Wilhelmsen, K., Cooper, T., Bogue, M., McLeod, H., et al. (2012) Antipsychotic-induced vacuous chewing movements and extrapyramidal side effects are highly heritable in mice.. *The pharmacogenomics journal,* **12**(2), 147.

[13] Montgomery, S., Sammeth, M., Gutierrez-Arcelus, M., Lach, R., Ingle, C., Nisbett, J., Guigo, R., and Dermitzakis, E. (2010) Transcriptome genetics using second generation sequencing in a Caucasian population. *Nature,* **464**(7289), 773–777.