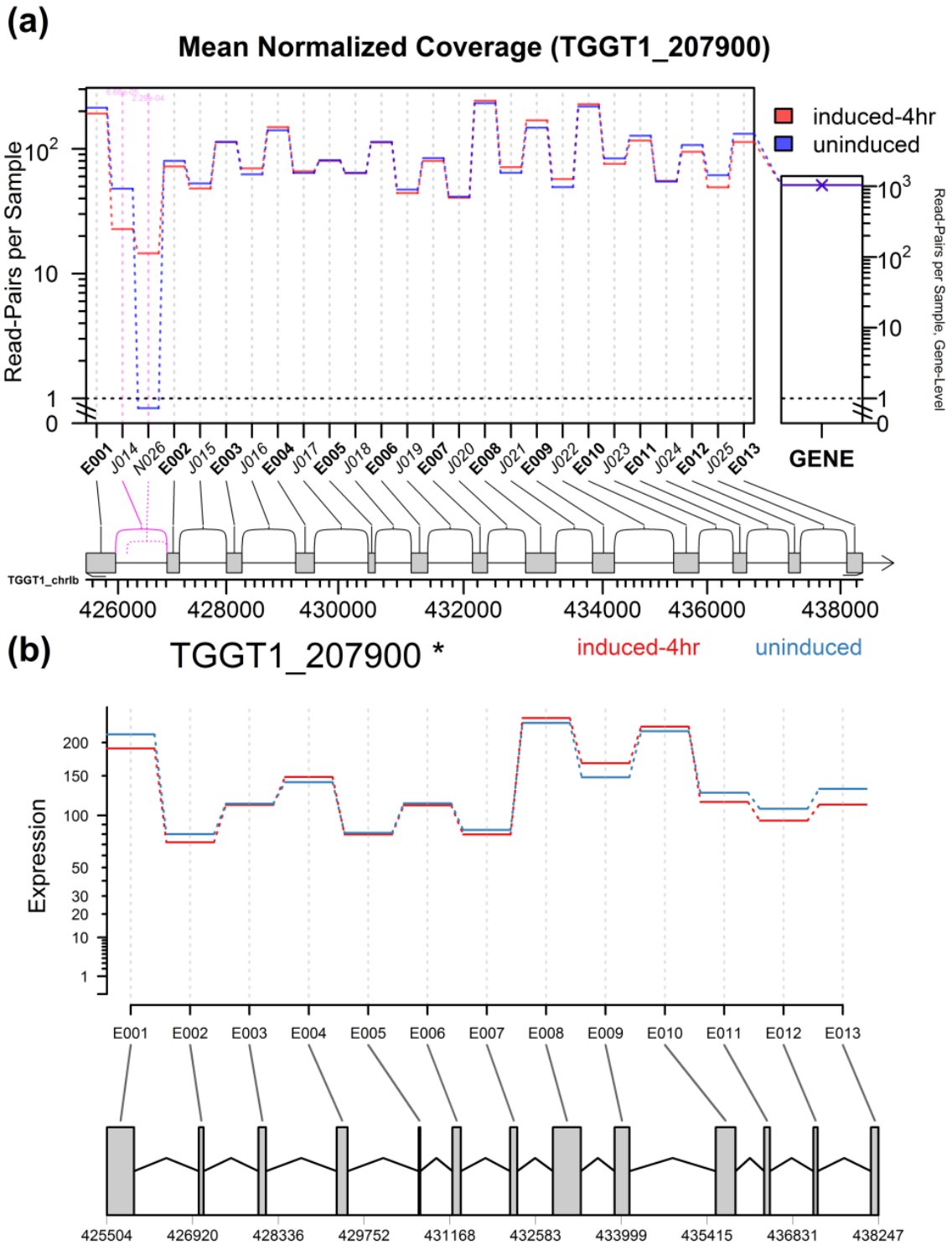
Detection and Visualization of Differential Exon and Splice Junction Usage in RNA-Seq Data with JunctionSeq (Supplement)

Table of Contents

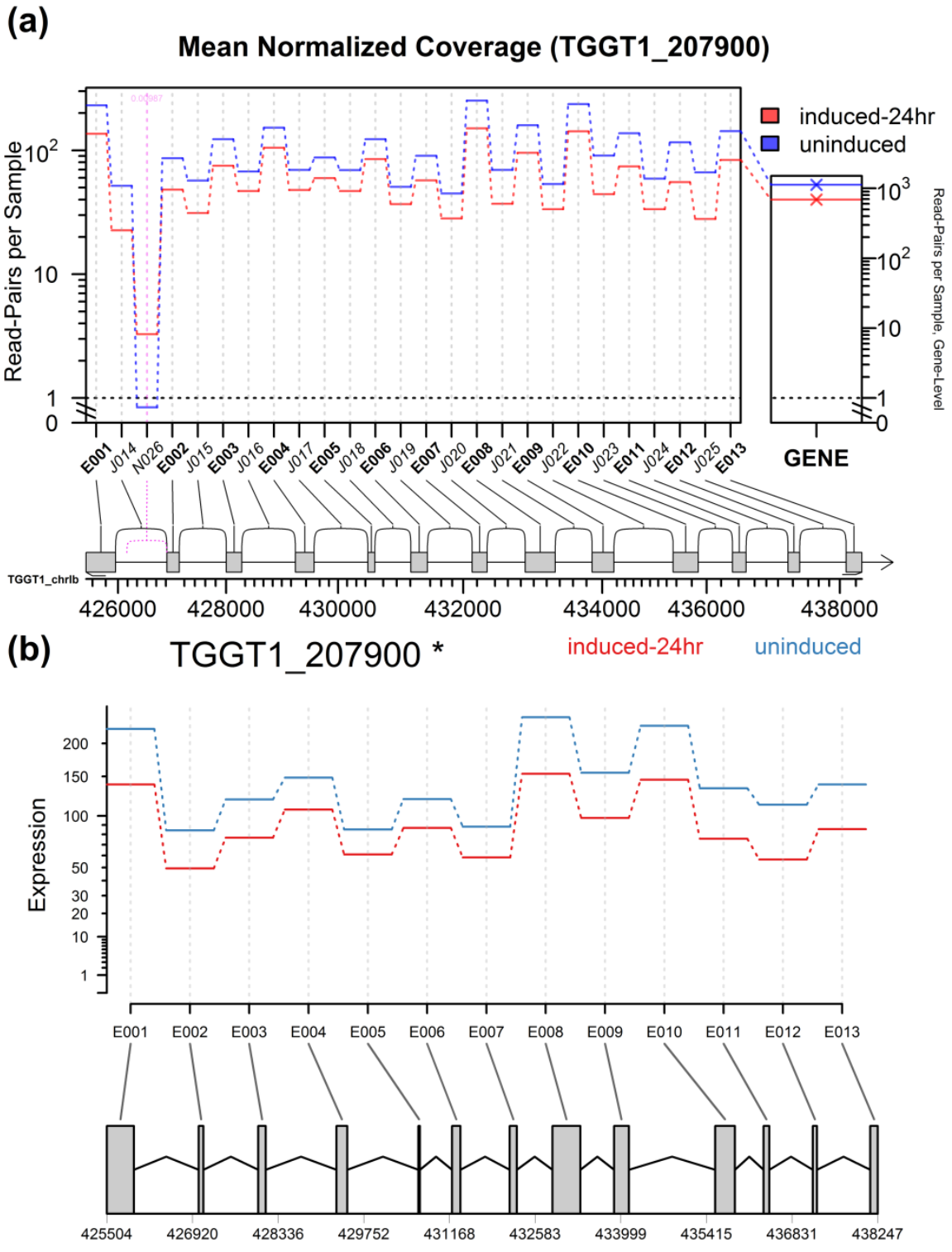
1 Supplementary Results

1.1 Supplemental fig. 1: Gene profile plots, 4-hr vs control.



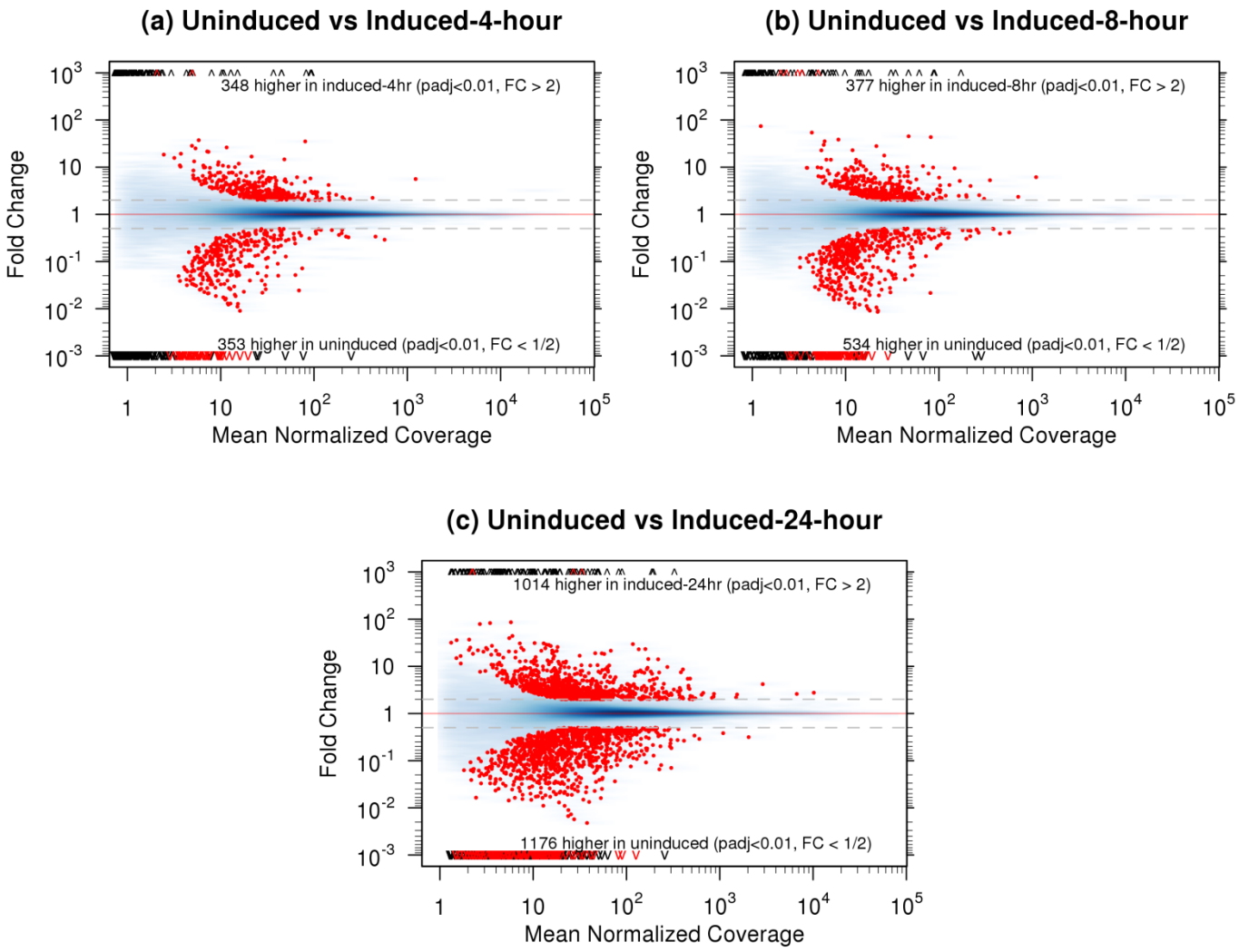
Displays the same information as figure 3 from the main text, except for the uninduced vs 4-hour-induced *Toxoplasma gondii* experiment.

1.2 Supplemental fig. 2: Gene profile plots, 24-hr vs control.



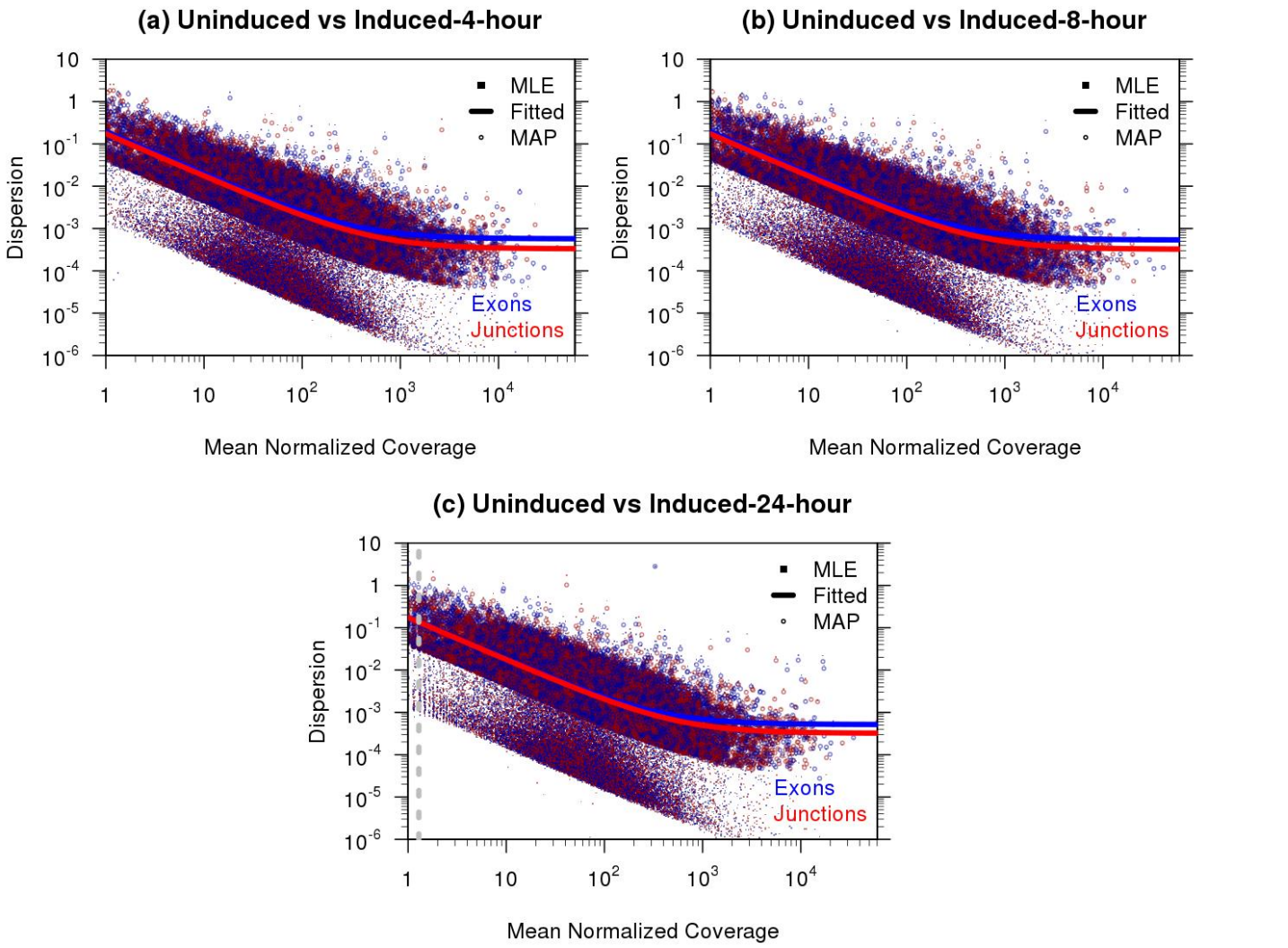
Displays the same information as figure 3 from the main text, except for the uninduced vs 24-hour-induced *Toxoplasma gondii* experiment.

1.3 Supplemental fig. 3: MA-Plots for the Toxoplasma dataset



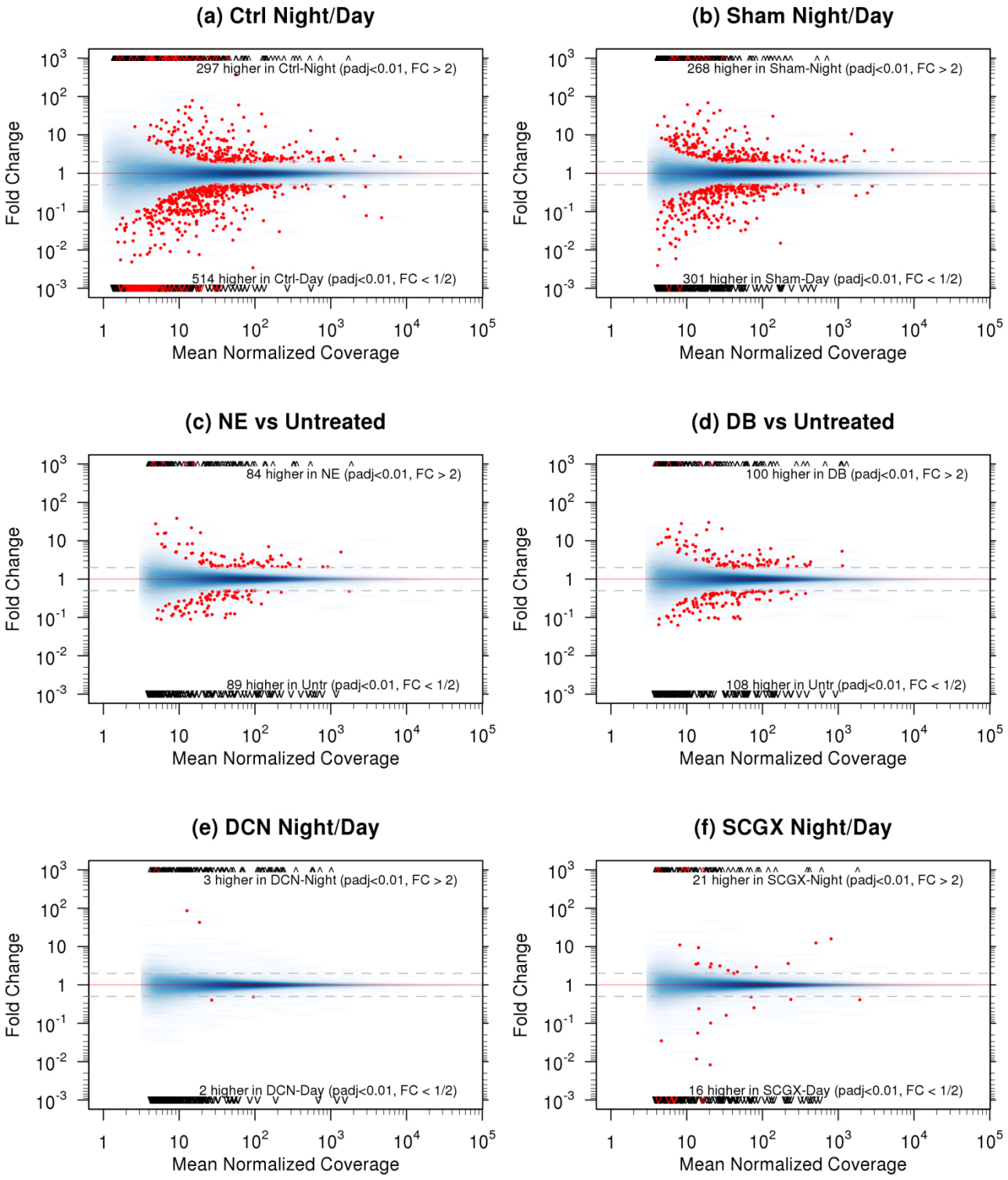
This standard diagnostic plot shows the mean normalized coverage against the fold change for all features in each experiment. The density of the large number of low-fold-change or non-significant features is shaded in blue. Features that pass the significance and fold-change thresholds (p-adjust < 0.01, fold change >2 in either direction) are marked with red dots.

1.4 Supplemental fig. 4: Dispersion estimation plot, Toxoplasma dataset.



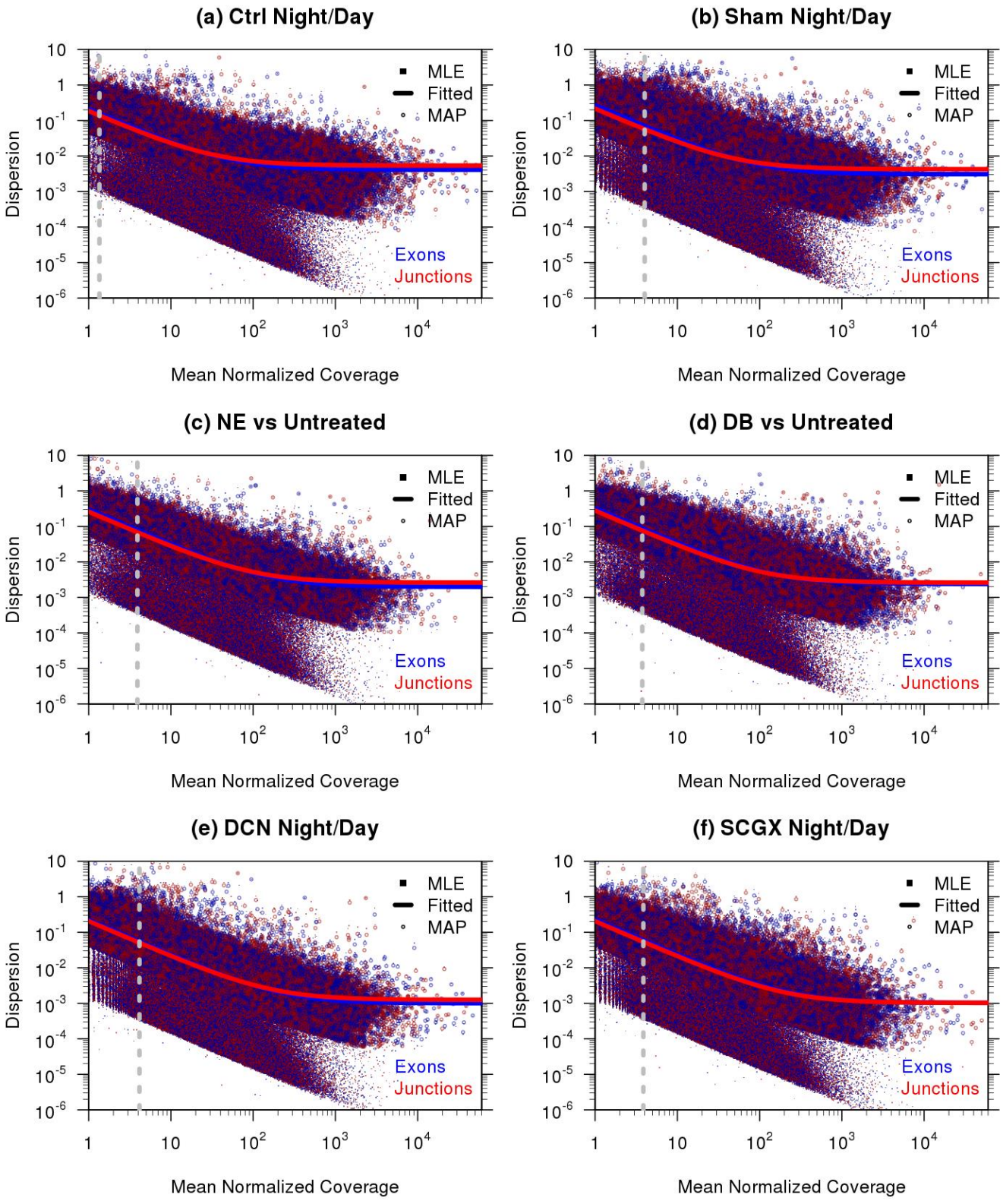
This standard diagnostic plot shows the various dispersion estimates for the toxoplasma dataset. The "unshared" or "feature-wise" dispersion estimates are drawn as blue or red points representing exons or junctions, respectively. The fitted dispersion estimate trend is drawn as a blue and red line for the exon and junction fitted dispersions, respectively. The maximum a posteriori dispersion, which is the default dispersion used in hypothesis testing, are drawn as blue or red circles.

1.5 Supplemental fig. 5: MA Plots for the 6 rat pineal experiments.



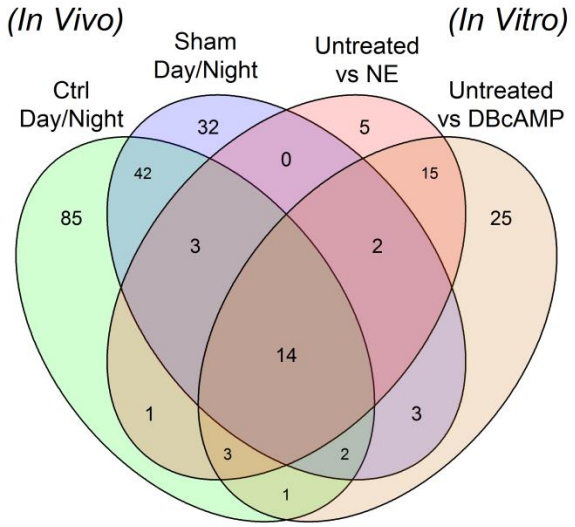
This standard diagnostic plot shows the mean normalized coverage against the fold change for all features in each experiment. The density of the large number of low-fold-change or non-significant features is shaded in blue. Features that pass the significance and fold-change thresholds (p-adjust < 0.01, fold change >2 in either direction) are marked with red dots.

1.6 Supplemental fig. 6: Dispersion estimation plot, rat pineal dataset.



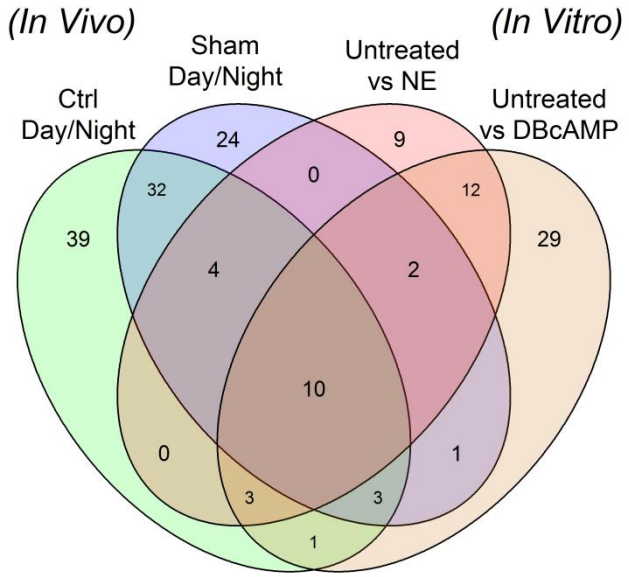
This standard diagnostic plot shows the various dispersion estimates for the toxoplasma dataset. The “unshared” or “feature-wise” dispersion estimates are drawn as blue or red points representing exons or junctions, respectively. The fitted dispersion estimate trend is drawn as a blue and red line for the exon and junction fitted dispersions, respectively. The maximum a posteriori dispersion, which is the default dispersion used in hypothesis testing, are drawn as blue or red circles.

1.7 Supplemental fig. 7: JunctionSeq Results Venn diagrams.



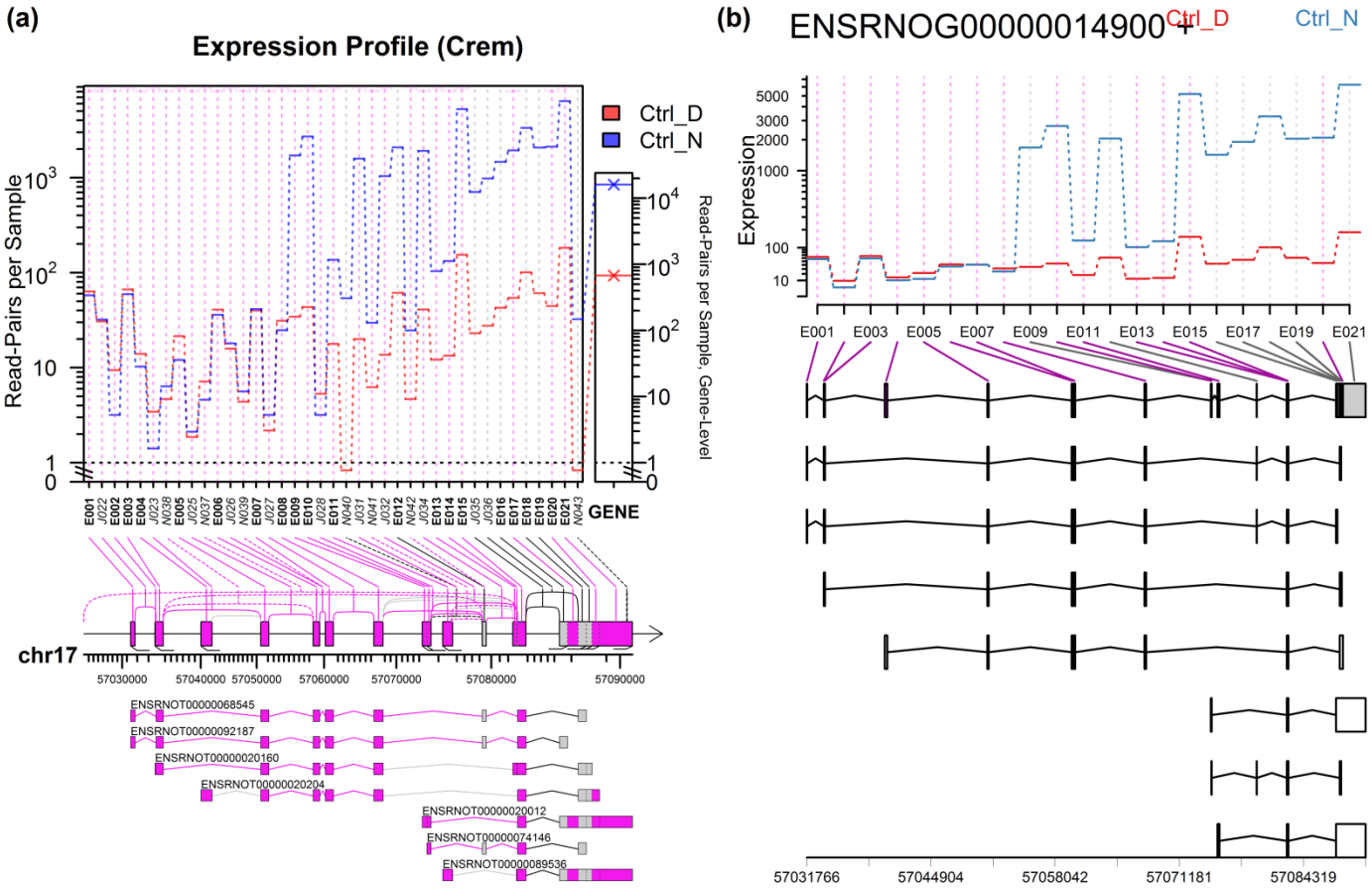
Venn diagrams describing the overlap in the genes found to possess splice junctions that display statistically significant (adjusted-p < 0.000001) differential usage in the four rat pineal analyses.

1.8 Supplemental fig. 8: DEXSeq results Venn diagrams.



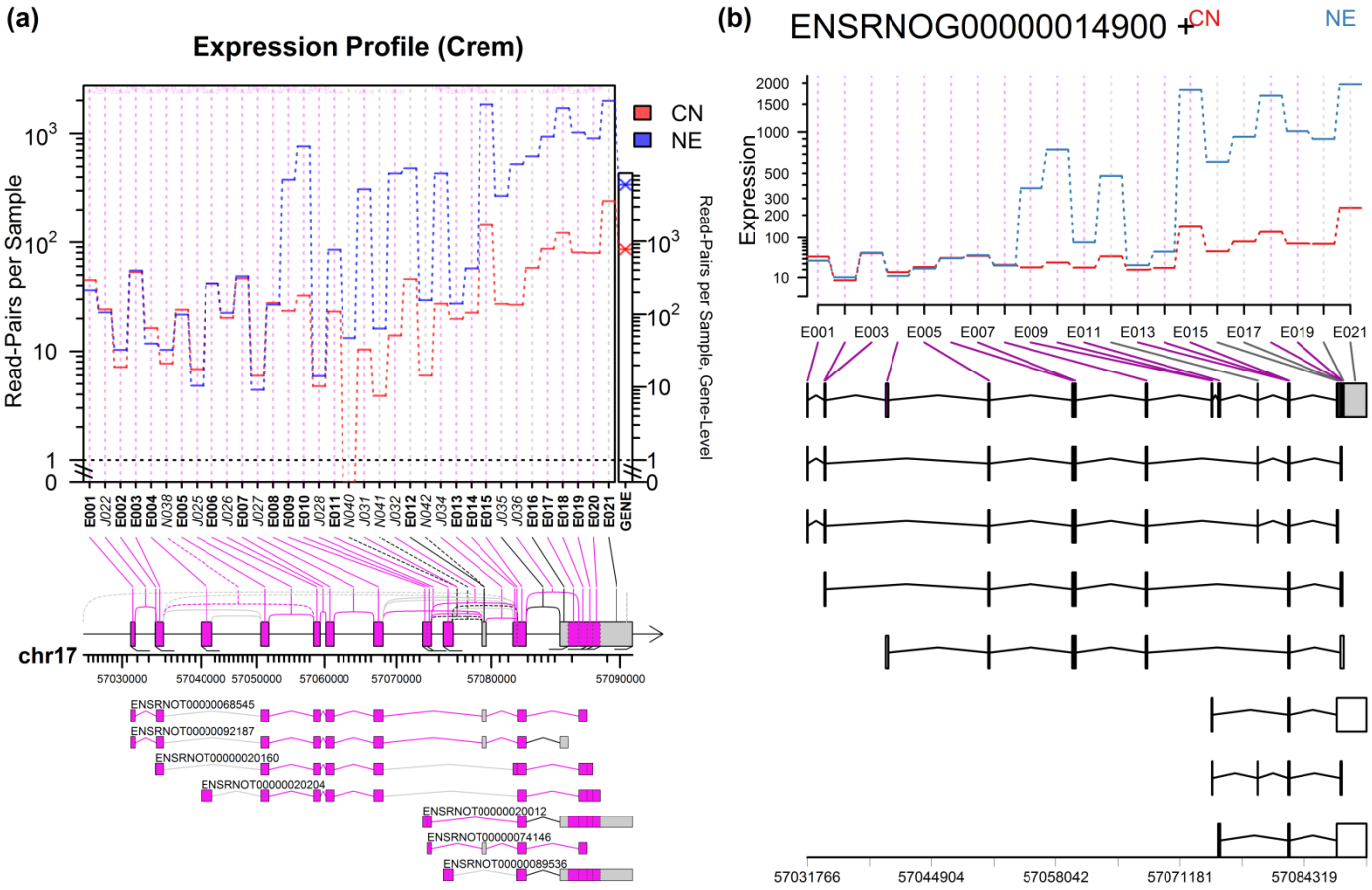
This Venn diagram, similar to the previous figure, displays the overlap between the results from the various in vivo and in vitro analyses performed by DEXSeq. As before, each cell displays the number of genes shared between the various analyses at adjusted-p < 0.000001.

1.9 Supplemental fig. 9: Gene profile plots for Crem, control night/day.



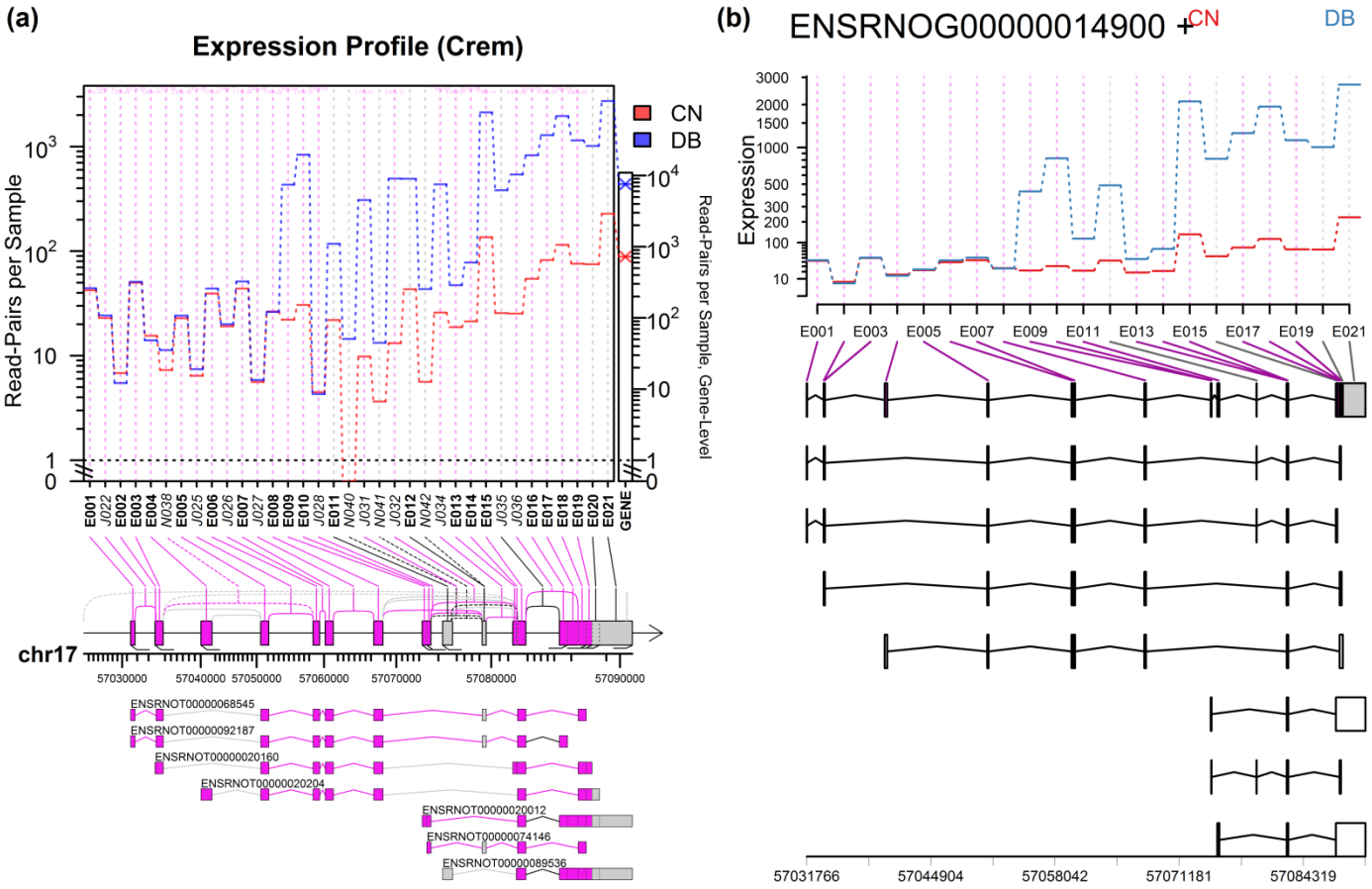
Comparison of gene profile plots generated by DEXSeq (a) and JunctionSeq (b) for the Crem gene in the sham day/night rat pineal gland experiment.

1.10 Supplemental fig. 10: Gene profile plots for Crem, untreated vs norepinephrine.



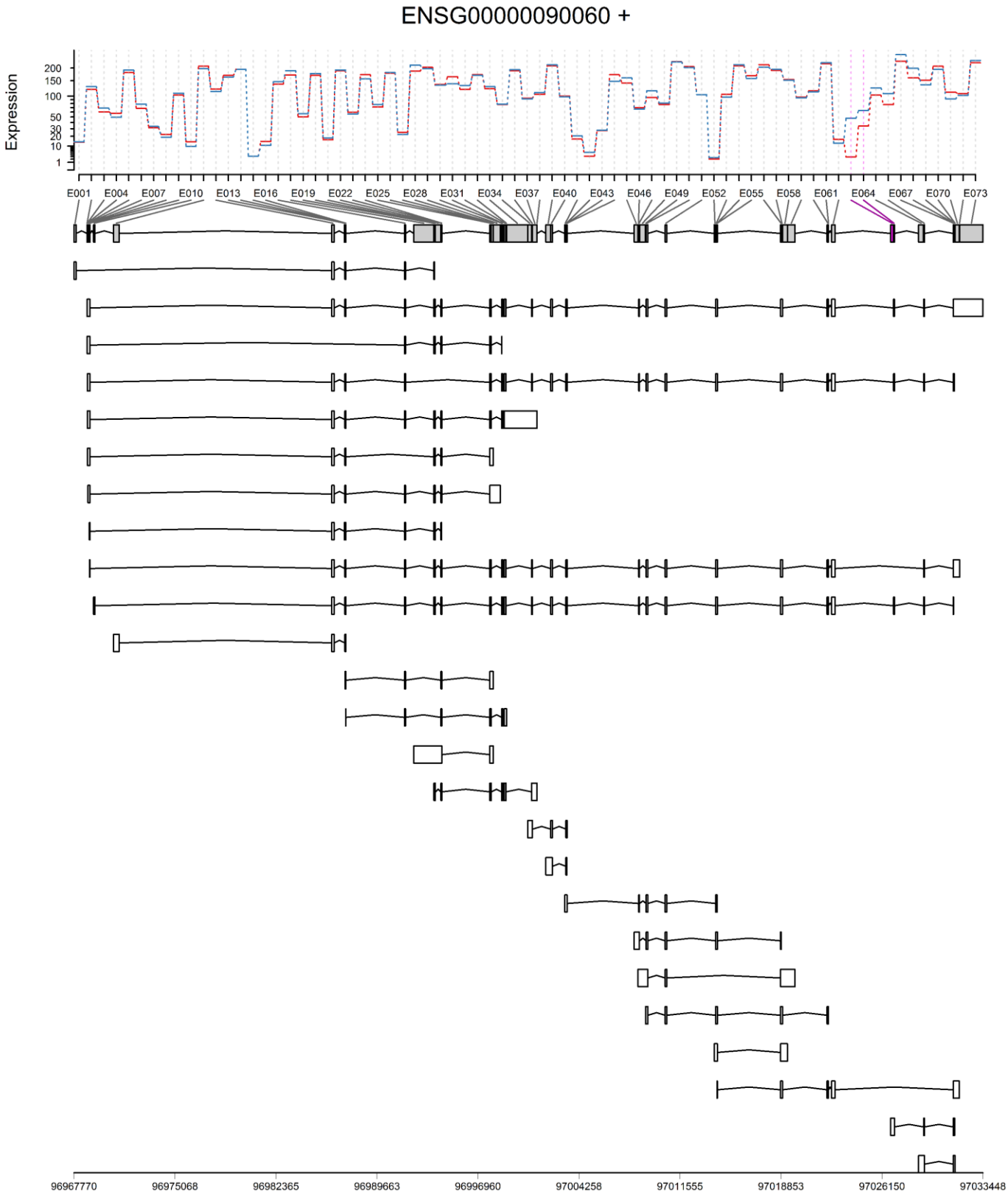
Comparison of gene profile plots generated by *JunctionSeq* (a) and *DEXSeq* (b) for the Crem gene in the untreated control vs DBcAMP-treated rat pineal gland experiment.

1.11 Supplemental fig. 11: Gene profile plots for Crem, untreated vs DBcAMP.



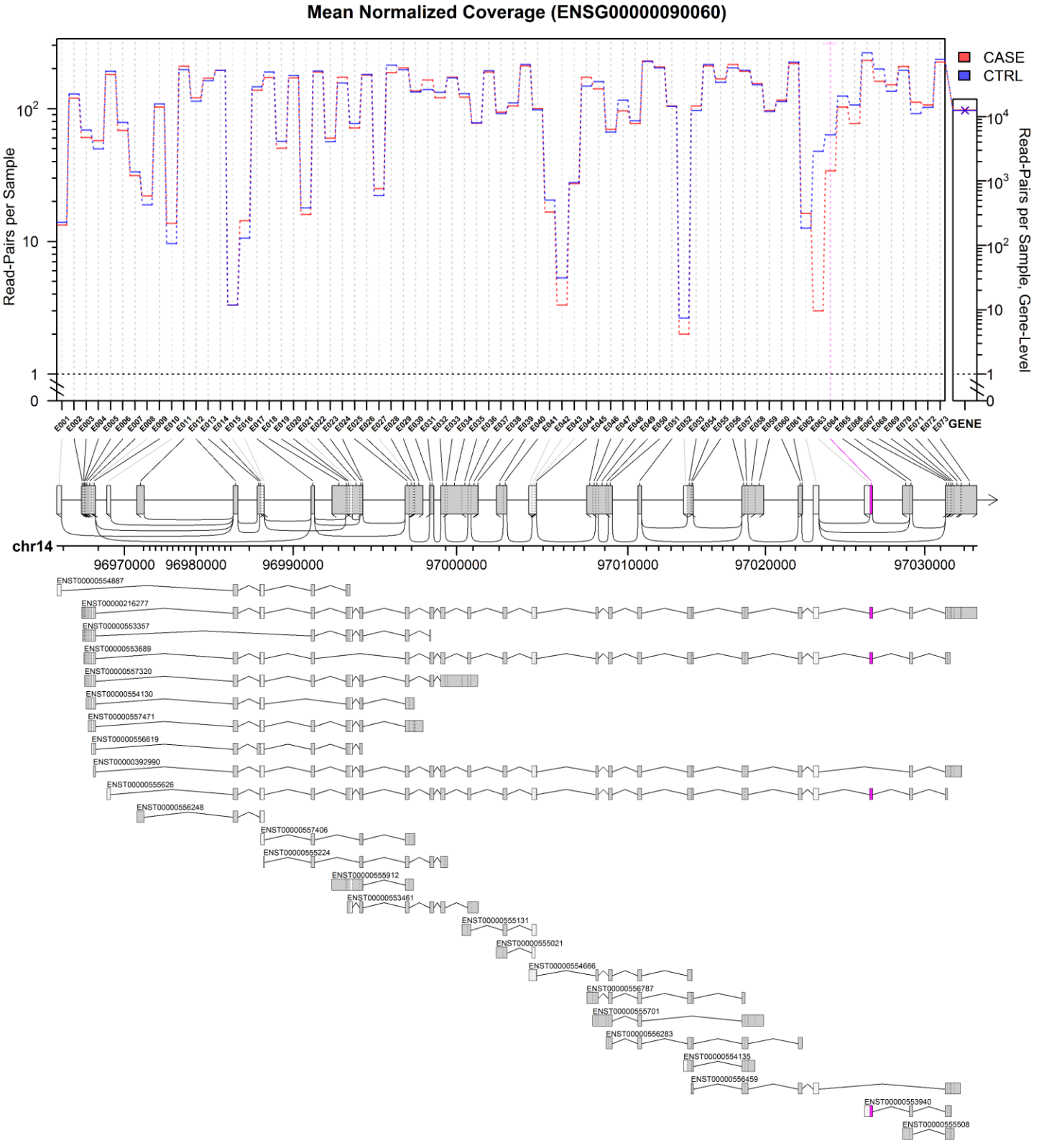
Comparison of gene profile plots generated by *JunctionSeq* (a) and *DEXSeq* (b) for the Crem gene in the untreated control vs NE-treated rat pineal gland experiment.

1.12 Supplemental fig. 12: DEXSeq Gene profile plot, example human gene (simulated data).



The above plot shows a DEXSeq plot for a moderately-complex human gene and a simulated dataset. Note that it is completely impossible to distinguish many exonic and splicing variants in the gene diagram, because many of the features are less than a pixel wide even at high resolution.

1.13 Supplemental fig. 13: JunctionSeq Gene profile plot, example human gene (simulated data).



Using a specific set of optional parameters, JunctionSeq will reproduce a standard DEXSeq analysis, and then output plots using its own improved visualization engine. The above plot shows a JunctionSeq plot replicating the analysis that produced Supplemental fig. 12for a moderately-complex human gene and a simulated dataset. Note the various improvements in the lower plot produced by JunctionSeq. For example: exonic labels are shrunk and drawn at an angle, exonic and intronic regions are rescaled to improve readability, the overall gene-level expression is shown on the right, nested splice junctions are included, exon fragments are separated by dotted lines instead of solid ones, and exonic regions that do not satisfy the hypothesis test inclusion thresholds are drawn with a lighter gray.

1.14 Supplemental fig. 14: Differential isoform usage does not necessarily perturb exon counts.

(a) Isoforms & Isoform Expression Levels

Isoforms of Hypothetical Gene X	TX	Isoform Fraction	
		CASE	CTRL
	1	50%	0%
	2	0%	50%
	3	0%	50%
	4	50%	0%

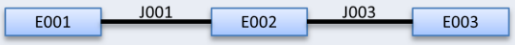
(b) Exon/Junction Counts

Exon/SJ	Mean Count (CASE)	Mean Count (CTRL)
E001	100	100
E002	50	50
E003	50	50
E004	100	100
J001	50	50
J002	0	50
J003	50	0
J004	50	0
J005	0	50
J006	50	50

A hypothetical scenario in which differential usage of multiple transcripts does not necessarily result in changes to the exon counts. In this (extreme) example, isoforms 1 and 4 are present in equal quantity the CASE samples and isoforms 2 and 3 are completely absent, whereas in the CTRL samples the opposite is true: isoforms 2 and 3 are present in equal quantities and isoforms 1 and 4 are completely absent. Even with such extreme change in the isoform fractions, the exon counts remain identical between cases and controls. As such, DEXSeq could not possibly detect such differences. Note, however, that splice junctions J002 through J005 obviously differ strongly between the two conditions.

1.15 Supplemental fig. 15: Exon counts may understate isoform differences

(a) Isoforms & Isoform Expression Levels

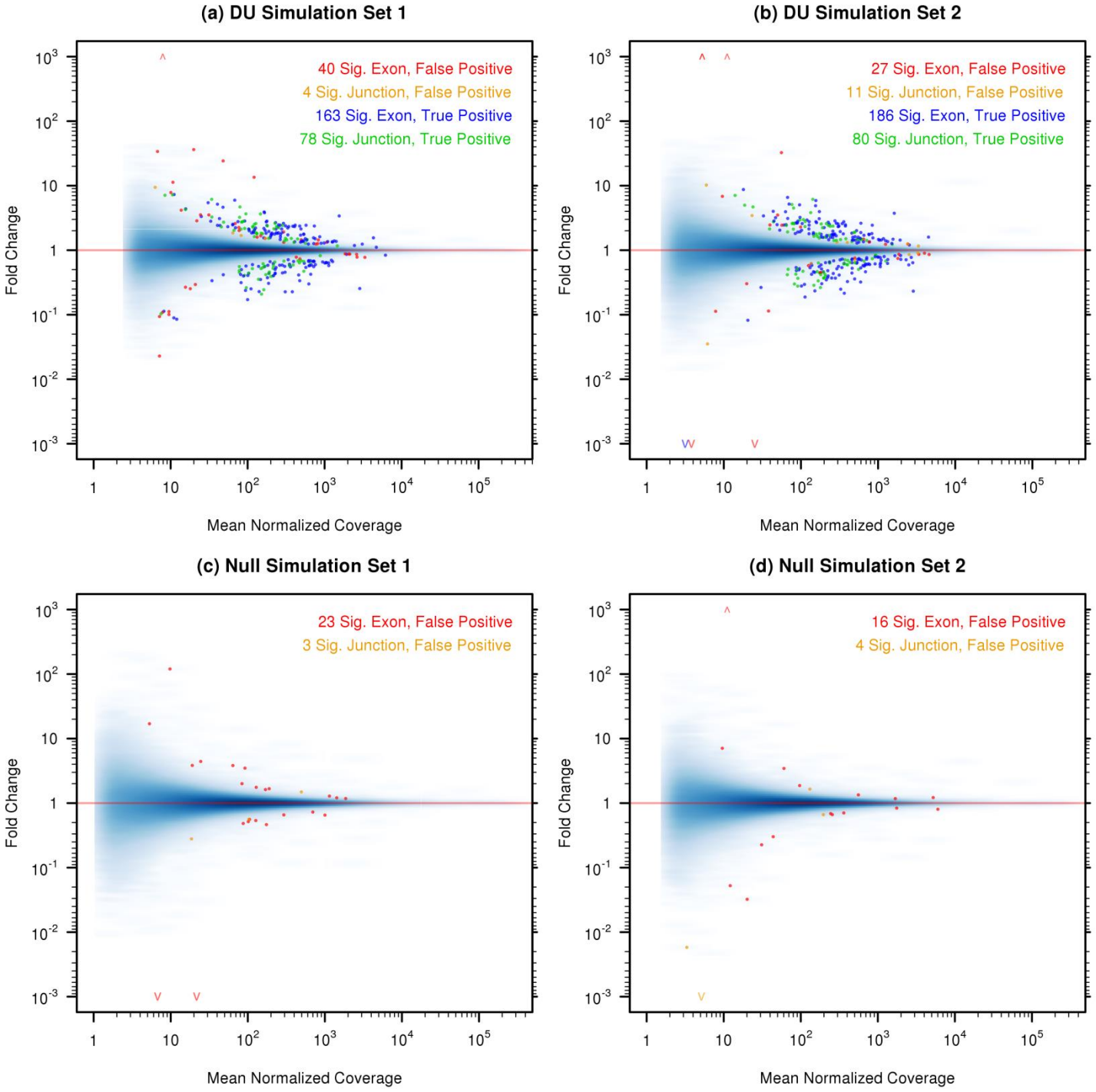
Isoforms of Hypothetical Gene X	TX	RPKM	
		CASE	CTRL
	1	1000	1000
	2	200	50

(b) Exon/Junction Counts

Exon/ SJ	RPKM (CASE)	RPKM (CTRL)	Fold Difference (CASE / CTRL)
E001	1200	1050	1.142
E002	1000	1000	1.000
E003	1200	1050	1.142
J001	1000	1000	1.000
J002	200	50	4.000
J003	1000	1000	1.000
Gene	2200	2100	1.048

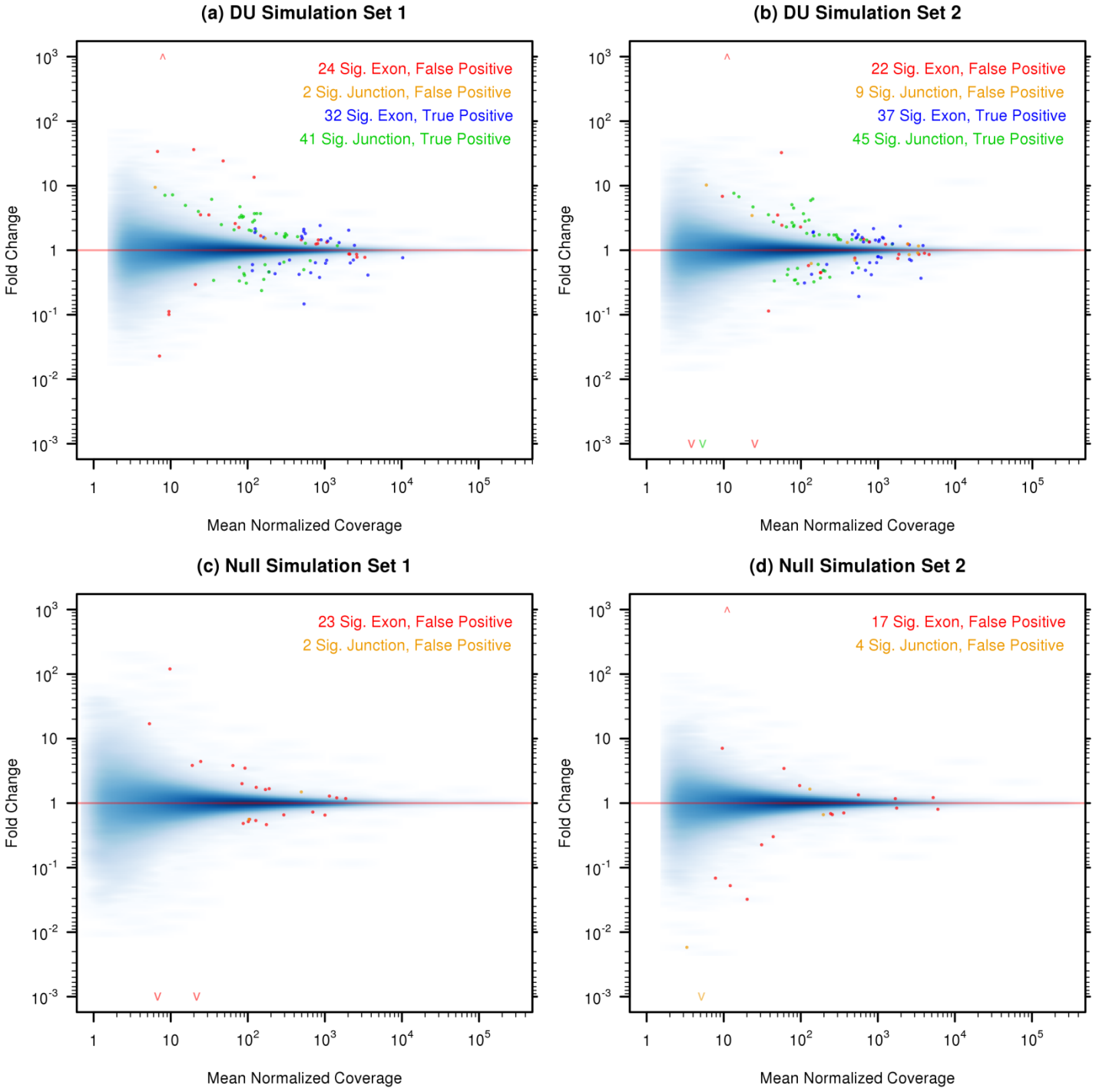
Another hypothetical scenario in which exon counts alone do not adequately characterize an instance of alternative isoform regulation (AIR). In this scenario the minor isoform (transcript 2) is upregulated by a factor of 4 in the CASE samples. However, even in the CASE samples the expression of transcript 2 is still much less than the expression of the major isoform (transcript 1). Although in this scenario the difference does change exon counts to some extent, estimating differential isoform usage using only the exon counts would underestimate the strength of the expression differences between cases and controls. Visual examination of a gene profile plot that only showed exon counts would similarly underestimate the strength of the effect. (Note: RPKM's are used in this example for simplicity. *JunctionSeq* does **not** use RPKMs.)

1.16 Supplemental fig. 16: MA Plots, Simulated Data, Full Annotation, p-adjust < 0.0001



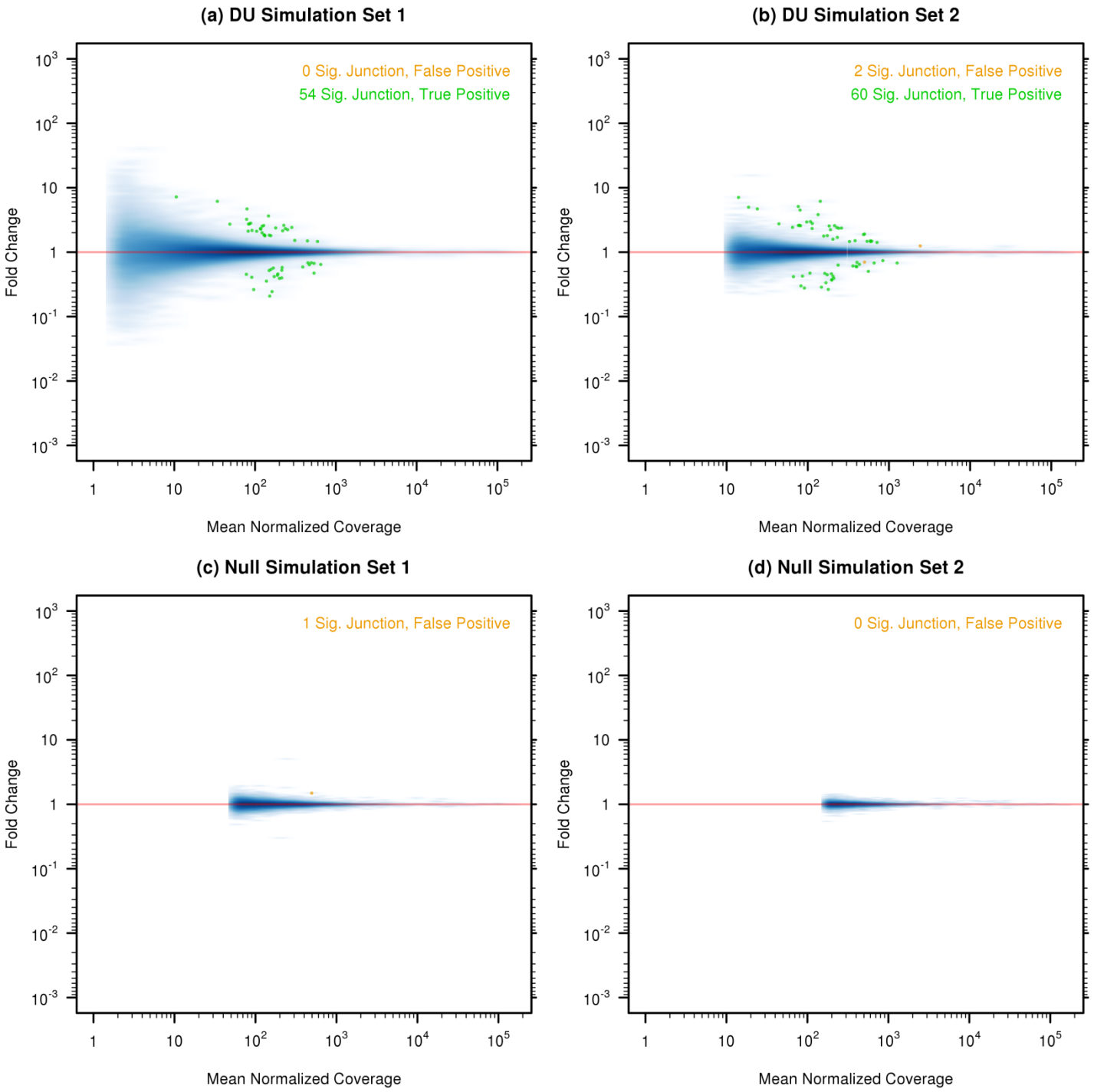
These plots display the fold change vs the mean normalized counts for features in the four simulated analyses, generated via JunctionSeq using the incomplete annotation set. Features are marked as significant if the adjusted-p-value was less than 0.0001. (a) and (b) display results for the simulated analyses in which 250 genes had simulated differential usage (DU) of transcripts. (c) and (d) display the results for the simulated analyses in which there was no true differential usage (and thus all significant features are false positives). Significant exons or junctions that belong to true AIR genes are marked with blue or green dots, respectively. Significant exons or junctions that do NOT belong to true AIR genes are marked with red or orange dots, respectively. Note that there are much fewer junction false discoveries than exon false discoveries.

1.17 Supplemental fig. 17: MA Plots, Simulated Data, Incomplete Annotation, p-adjust < 0.0001



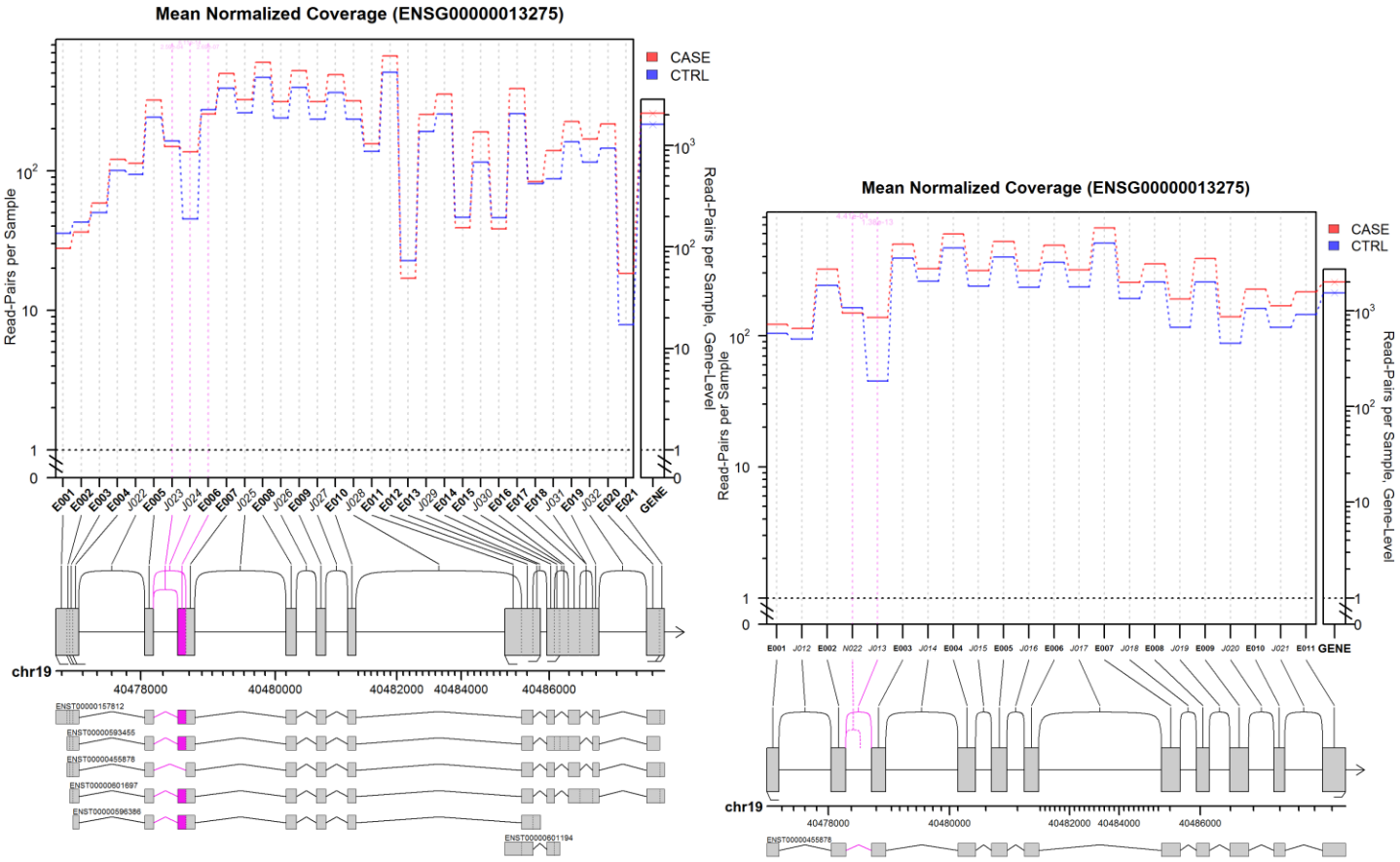
These plots display the fold change vs the mean normalized counts for features in the four simulated analyses, generated via JunctionSeq using the incomplete annotation set. Features are marked as significant if the adjusted-p-value was less than 0.0001. (a) and (b) display results for the simulated analyses in which 250 genes had simulated differential usage (DU) of transcripts. (c) and (d) display the results for the simulated analyses in which there was no true differential usage. Significant exons or junctions that belong to true AIR genes are marked with blue or green dots, respectively. Significant exons or junctions that do NOT belong to true AIR genes are marked with red or orange dots, respectively. Note that there are much fewer junction false discoveries than exon false discoveries.

1.18 Supplemental fig. 18: MA Plots, Simulated Data, Incomplete Annotation, p-adjust < 0.0001, using the splice-junctions-only and no-MAP options.



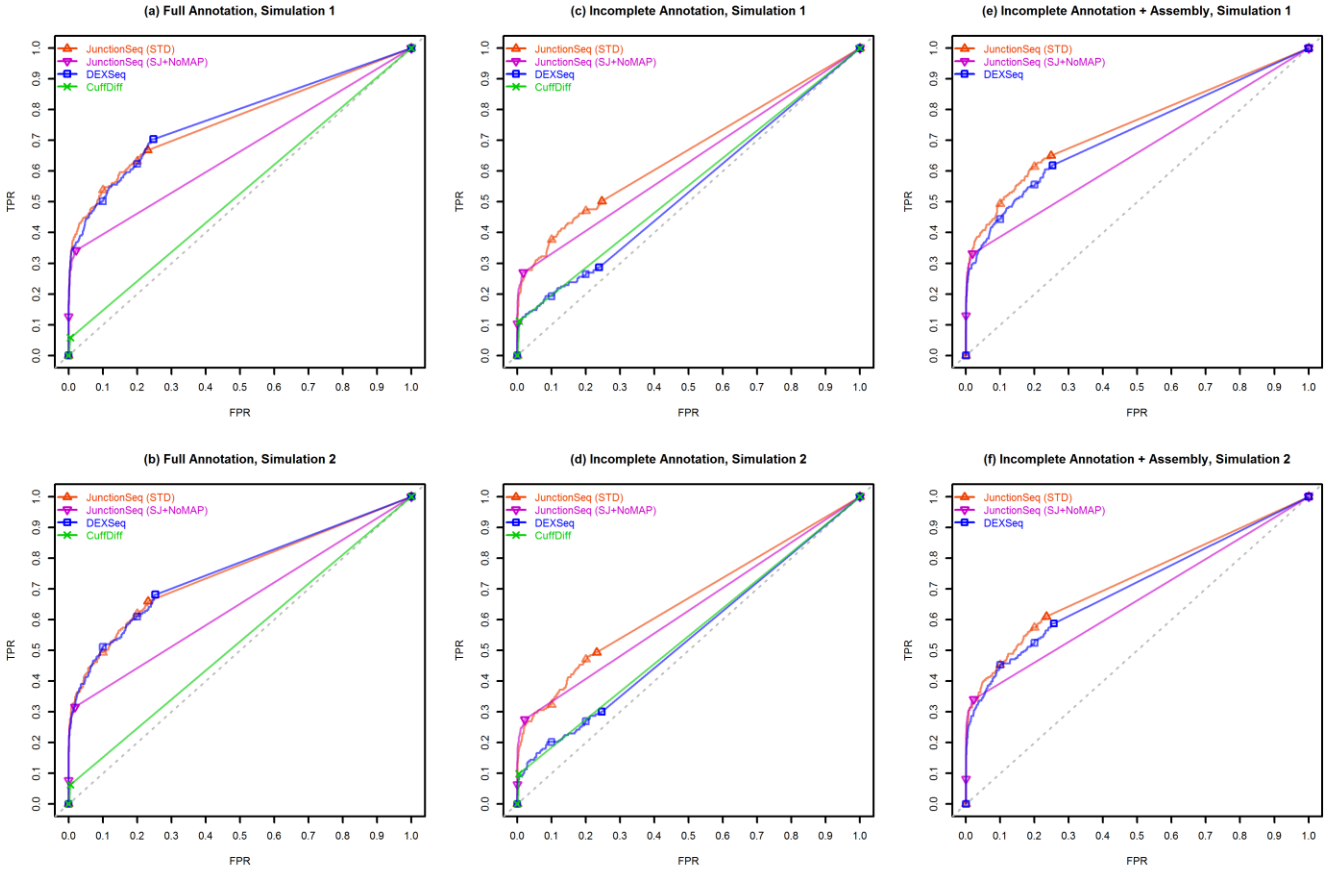
This table is analogous to Supplemental fig. 16, except that JunctionSeq was run with the options use.exons = FALSE and method.dispFinal = "max". These options cause JunctionSeq to only test splice junctions (not exonic regions), and to use the simple maximum of the unshared and fitted dispersions (as opposed to the maximum *a posteriori* dispersion) for hypothesis testing. This set of parameters is much more conservative, with a much lower false discovery rate analyses but also substantially reduced statistical power. Features are marked as significant if the adjusted-p-value was less than 0.0001.

1.19 Supplemental fig. 19: Example output from simulations analysis



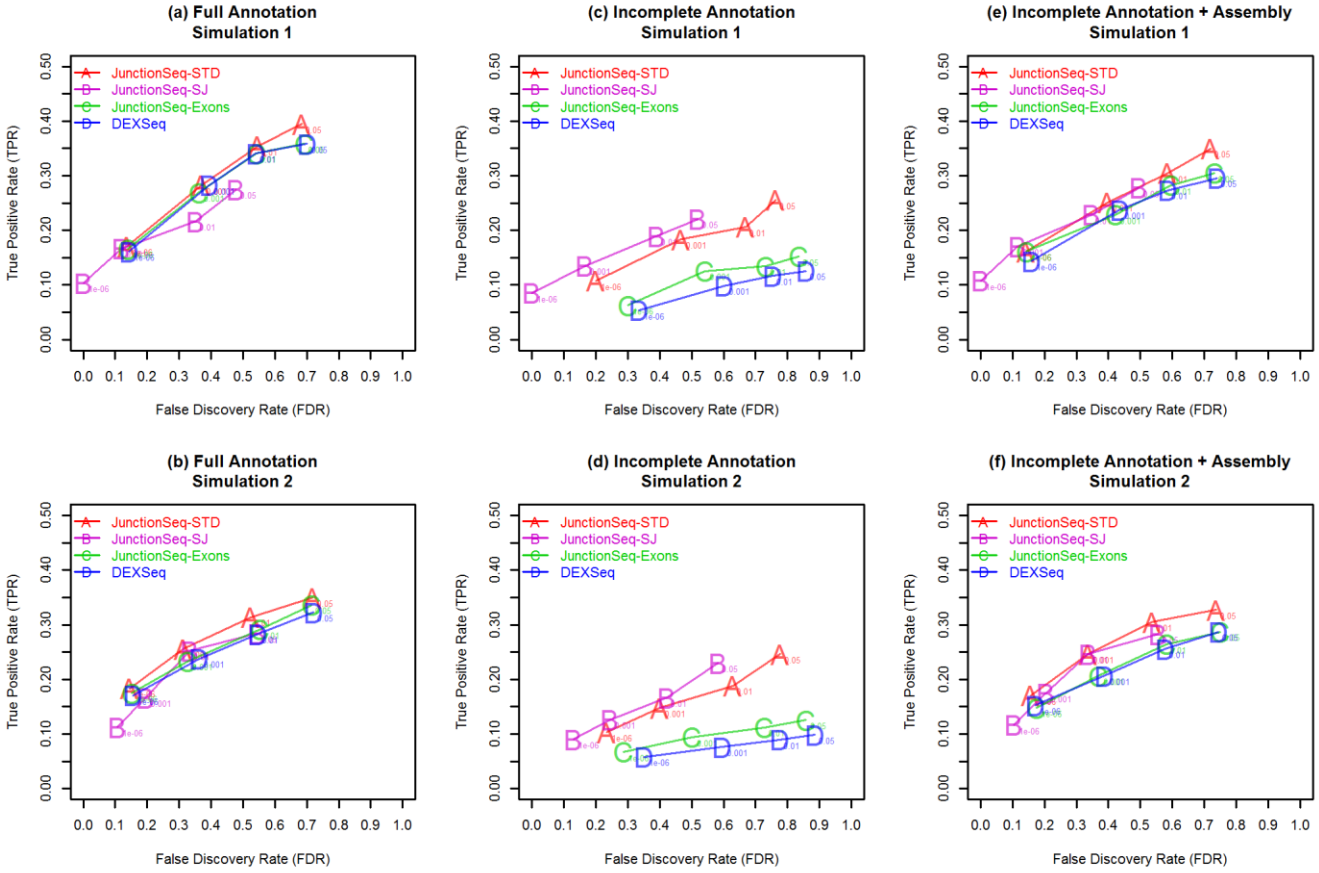
These plots display one example gene that was found to be highly significant by *JunctionSeq* both with and without the complete annotation (left and right figures, respectively).

1.20 Supplemental fig. 20: Full-range ROC Curves for *JunctionSeq*, *DEXSeq*, and *CuffDiff*



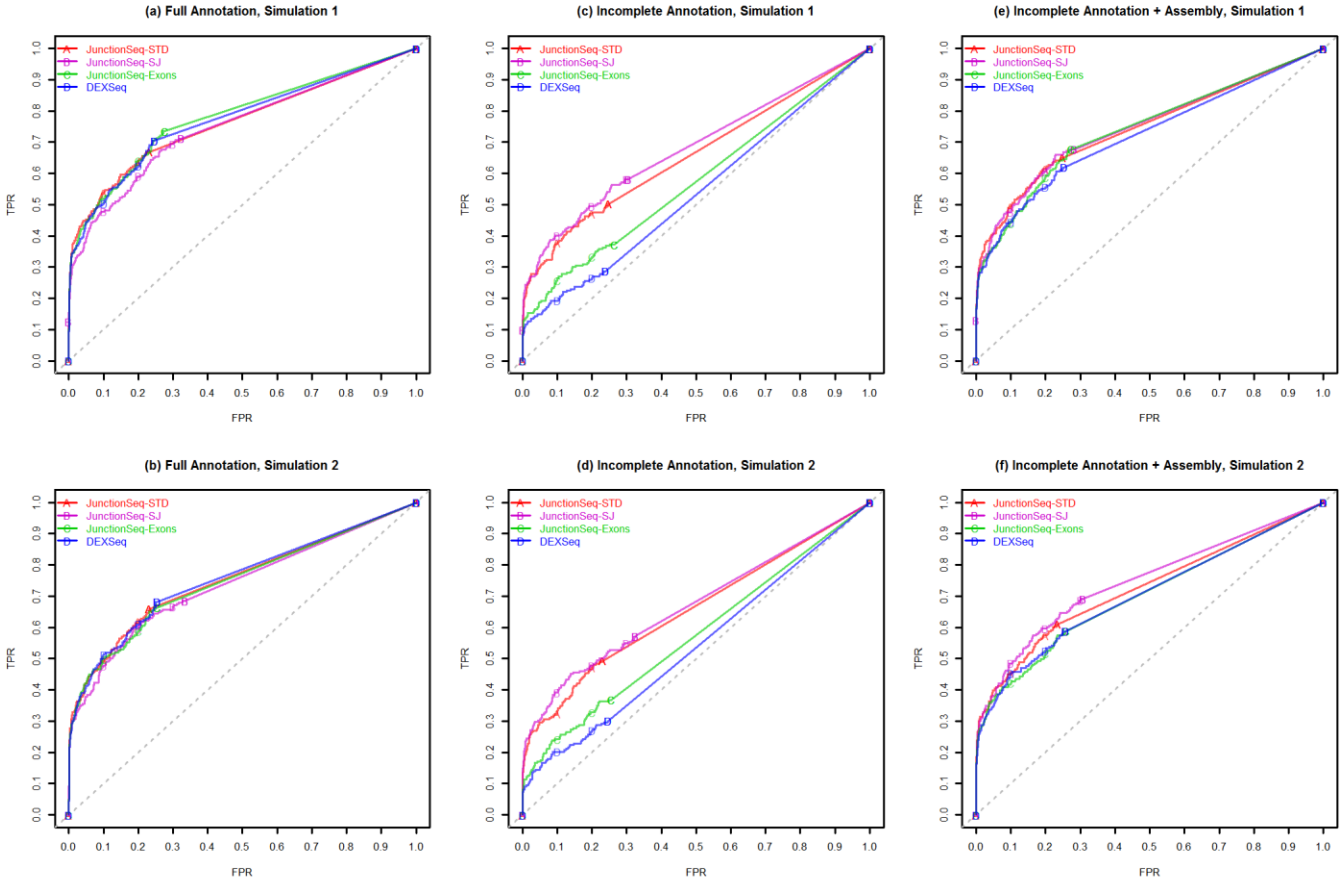
This figure displays the same curves from figure 1 in the main text, but over the full data range.

1.21 Supplemental fig. 21: JunctionSeq/DEXSeq TPR/FDR at various p-values, for various count-set parameters



This plot compares the TPR and FDR at various adjusted p-value thresholds for JunctionSeq runs that use both exons and junctions (magenta), junctions only (red), and exons only (green), compared with DEXSeq (blue, which always runs only using exons). The upper figures (a,c,e) show the results for the first simulated dataset; the lower figures (b,d,f) show the results for the second simulated dataset. Figures a-b show the results when using the full annotation, figures c-d show the results given the incomplete annotation, and figures e-f show the results for the incomplete annotation with a cufflinks assembly. See Supplemental Table 9 for a description of the options used in each JunctionSeq analysis.

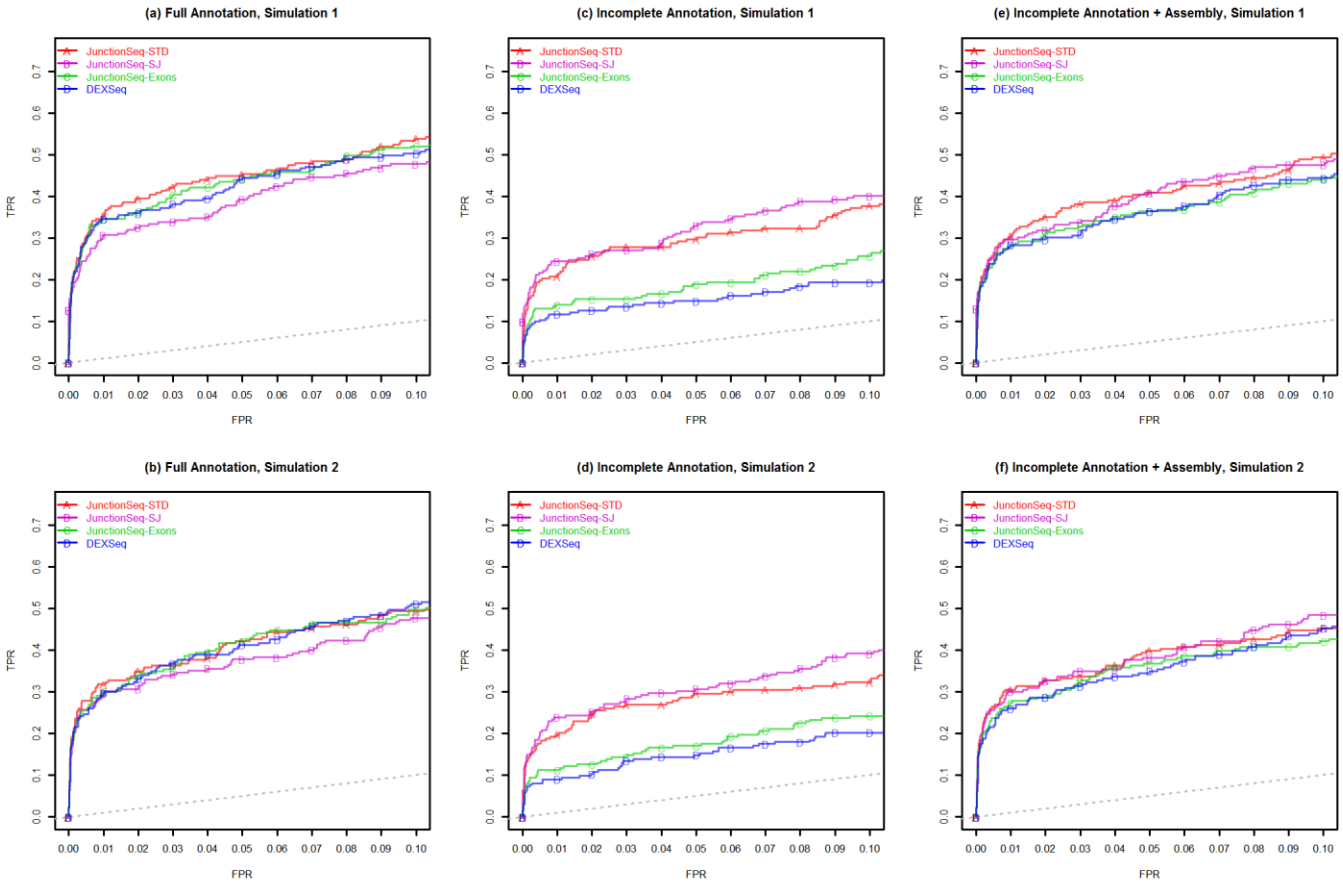
1.22 Supplemental fig. 22: JunctionSeq/DEXSeq ROC curves, various count-set parameters



ROC curves for the same analyses as the previous figure. Note that the JunctionSeq-SJ mode does not show clear superiority over the standard JunctionSeq analysis in all cases.

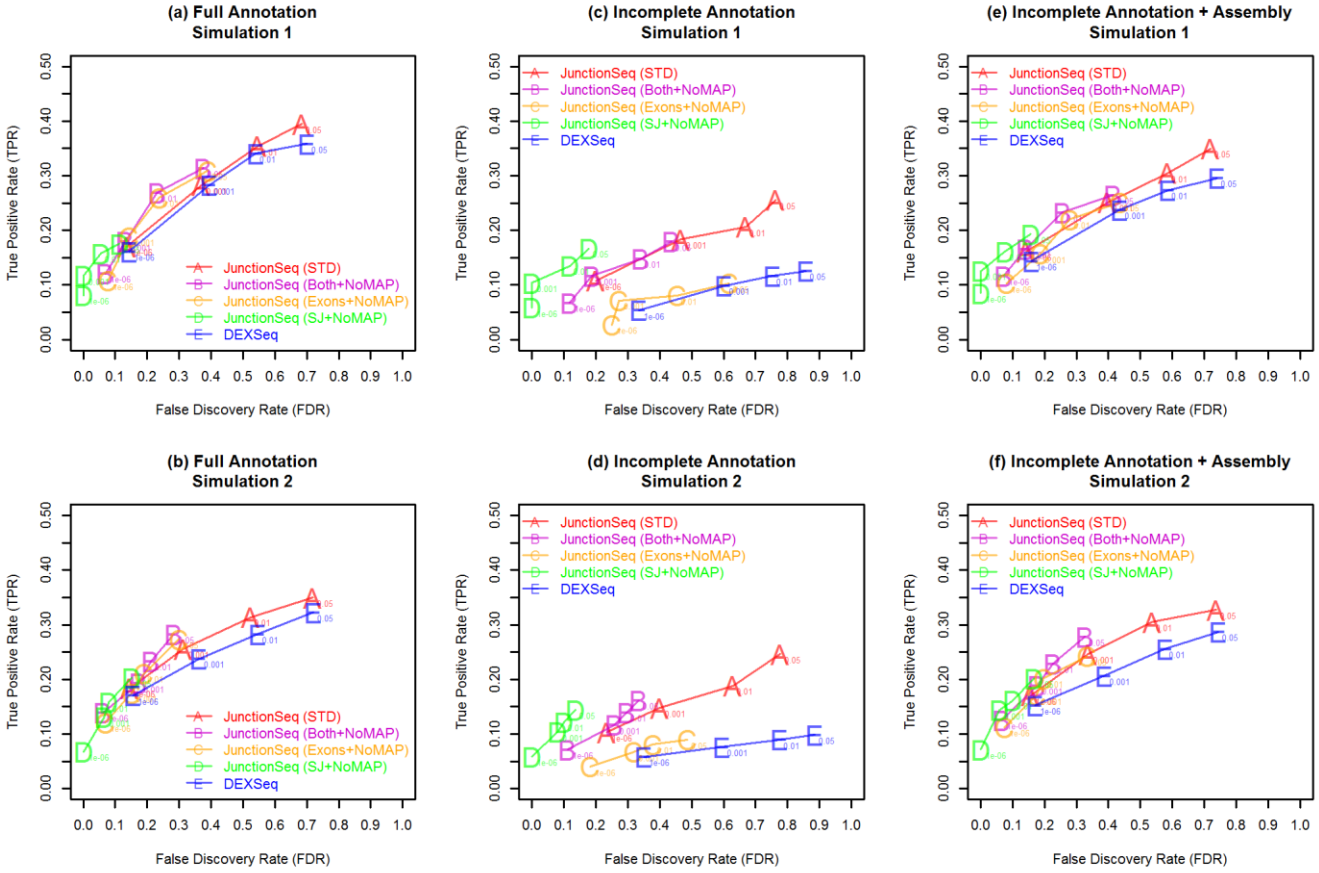
See Supplemental Table 9 for a description of the options used in each JunctionSeq analysis.

1.23 Supplemental fig. 23: JunctionSeq/DEXSeq ROC curves, various count-set parameters



Detail from the high end of the ROC curves from the previous figure. See Supplemental Table 9 for a description of the options used in each JunctionSeq analysis.

1.24 Supplemental fig. 24: JunctionSeq/DEXSeq TPR/FDR at various p-values, for various count set parameters, using the "max" final dispersion rule

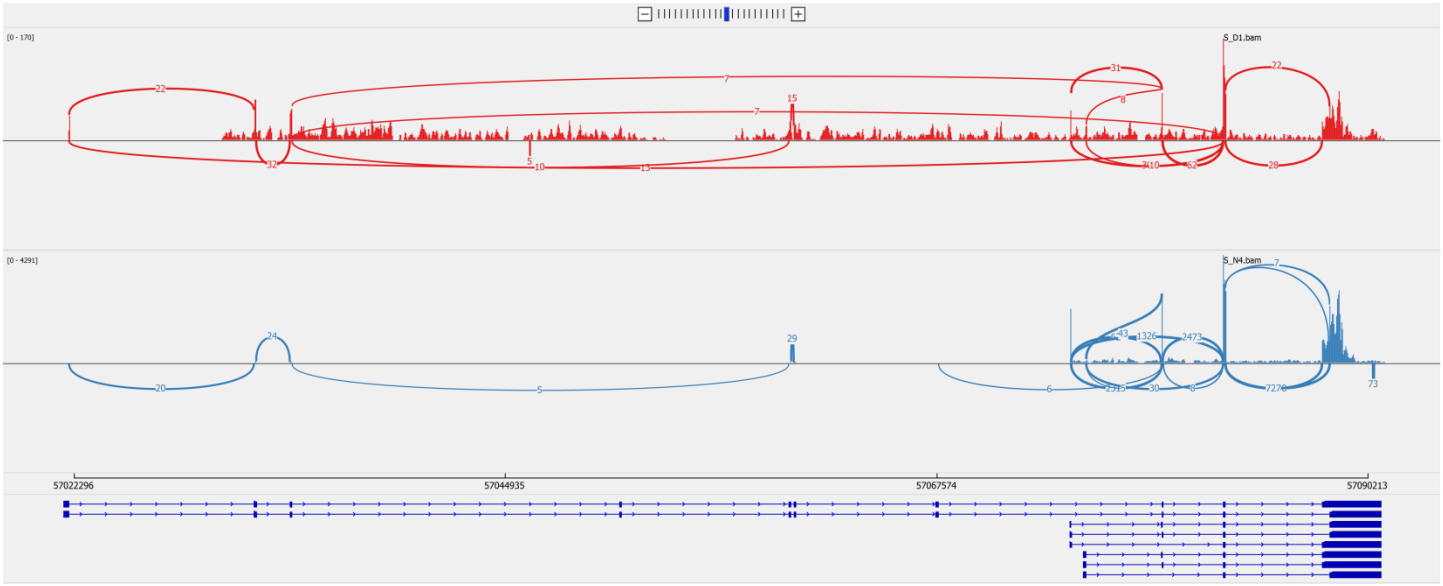


This plot is similar to Supplemental fig. 21, except that it shows the standard JunctionSeq and DEXSeq analyses compared with various JunctionSeq analyses that do not use the maximum *a posteriori* estimates for the final dispersion, instead using the maximum of the unshared and fitted dispersion estimates. See Supplemental Table 9 for a description of the options used in each JunctionSeq analysis.

1.25 Supplemental fig. 25: Snapshot of IGV browser view of the CREM gene for two rat pineal samples.

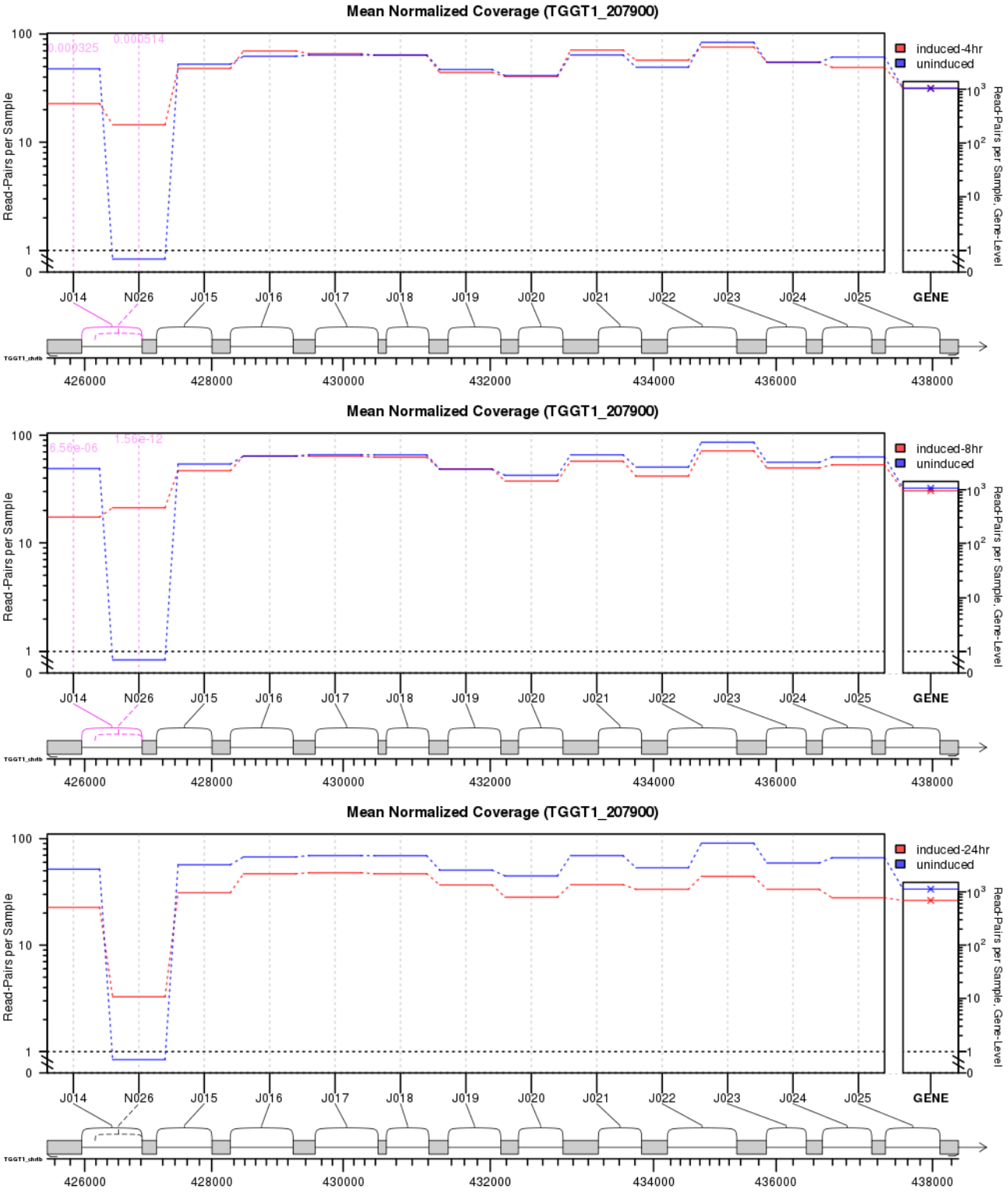
Snapshot of IGV panels for aligned reads across the CREM gene for two samples from the rat pineal data (one day and one night from the Sham-surgery group).

1.26 Supplemental fig. 26: IGV Sashimi plot for two samples in the rat pineal dataset.



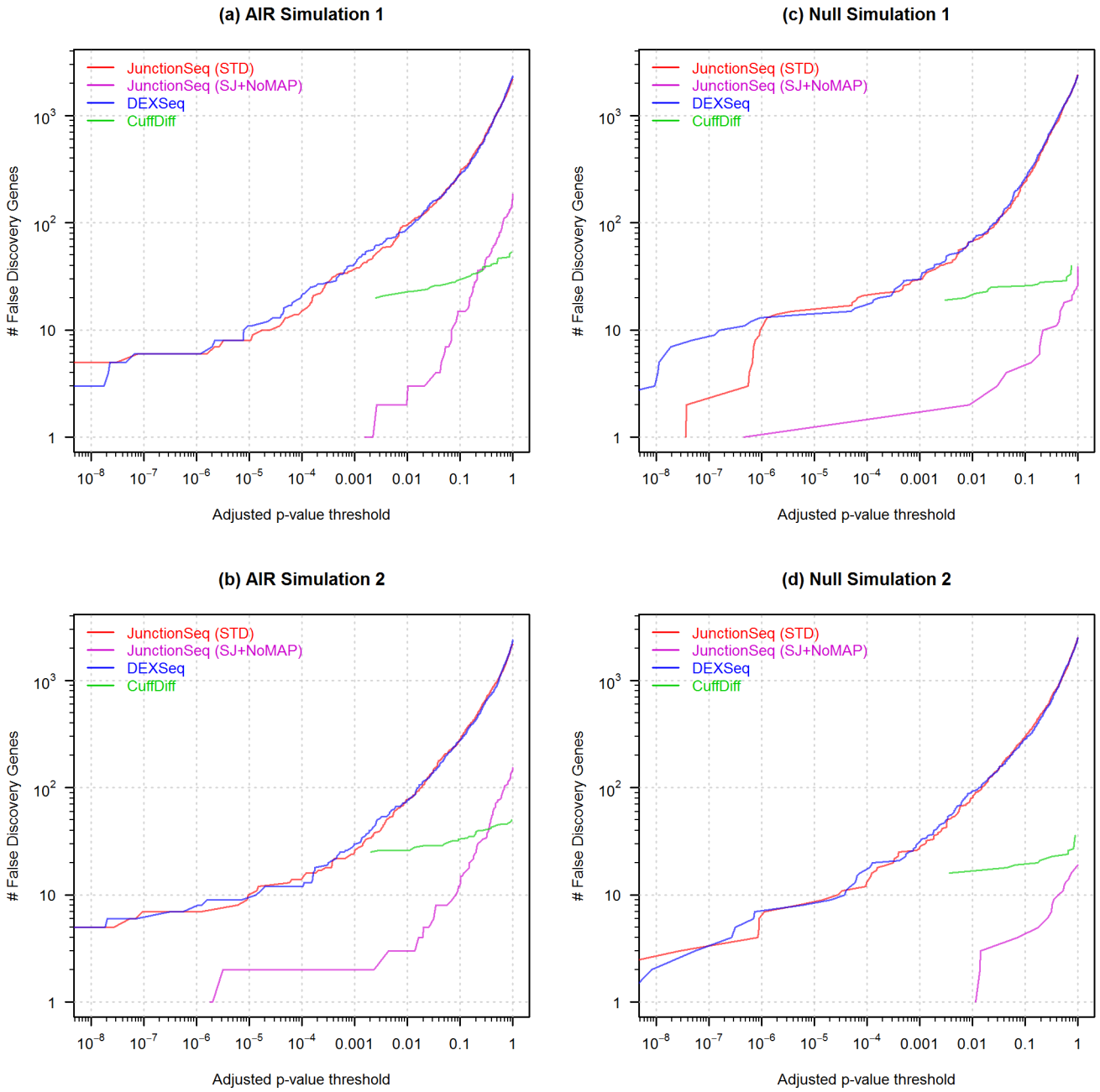
A “Sashimi” plot generated by IGV for the same two samples and the same interval as Supplemental fig. 25. Note that many of the splice junctions are plotted over by other splice junctions, rendering them illegible.

1.27 Supplemental fig. 27: Results for the confirmed AIR gene in the toxoplasma dataset, using the splice-junction-only and no-MAP options



These plots are analogous to Figure 3a in the main text and Supplemental fig. 1a and Supplemental fig. 2a, except that JunctionSeq was run with the options use.exons = FALSE and method.dispFinal = "max". These options cause JunctionSeq to only test splice junctions (not exonic regions), and to use the simple maximum of the unshared and fitted dispersions (as opposed to the maximum *a posteriori* dispersion) for hypothesis testing. This set of parameters is much more conservative, with a much lower false discovery rate in simulated analyses but also substantially reduced statistical power. As anticipated, in most of the comparisons the results are approximately the same, but in the comparison with the weakest effects (uninduced vs induced-24hr) the differential usage is no longer statistically significant.

1.28 Supplemental fig. 28: False discoveries in AIR and null simulations analyses, by p-value threshold



The above plot displays the number of false positive genes discovered by each method in the four simulated comparisons: 2 with AIR genes ("AIR simulations"), 2 with no AIR genes ("Null Simulations"). For each adjusted p-value threshold (x-axis), the number of non-AIR genes that are falsely detected is shown (p-adjust < threshold).

Note that the JunctionSeq (SJ+NoMAP) analysis method generally has the fewest false discoveries. Also note that even under idealized circumstances the presence of true-AIR genes increases the raw number of false discoveries. This is because the p-value adjustment method used is designed to control the false discovery rate and thus will be less stringent in the presence of a large number of true positives.

1.29 Supplemental Table 1: *DEXSeq* results summary, rat pineal dataset

Adjusted p-value threshold	In vivo			In vitro			Overlap, All Four
	Ctrl Day/Night	Sham Day/Night	Overlap, in vivo	CN vs NE	CN vs DBcAMP	Overlap, in vitro	
0.01	289	240	110	121	192	86	28
0.001	189	152	85	90	129	67	20
0.0001	147	106	69	61	87	42	17
0.00001	112	82	56	44	67	32	10
0.000001	92	76	49	40	61	27	10

Similar to Table 1 from the main text. Lists the numbers of genes found by *DEXSeq* to exhibit significant differential exon usage for the five rat pineal gland analyses**.**

1.30 Supplemental Table 2: Results for the four known-AIR genes using JunctionSeq using the splice-junction-only and no-MAP options

Gene Symbol	Full Annotation				Incomplete Annotation (1 "known" isoform)			
	Ctrl Day/Night	Sham Day/Night	CN vs NE	CN vs DBcAMP	Ctrl Day/Night	Sham Day/Night	CN vs NE	CN vs DBcAMP
Atp7b	4.6e-6	0.045765	0.000138	0.573551	4.6e-6	0.045	0.00011	0.55
Crem	<1e-8	<1e-8	<1e-8	<1e-8	<1e-8	<1e-8	3.7e-8	1.2e-7
Pde4b	<1e-8	<1e-8	3.0e-8	3.7e-06	<1e-8	<1e-8	2.7e-6	7.7e-5
Slc15a1	7.4e-8	2.79E-05	0.005829	1.0	7.4e-8	2.8e-5	1.0	1.0

This table is analogous to table 2 in the main text, except that JunctionSeq was run with the options use.exons = FALSE and method.dispFinal = "max". These options cause JunctionSeq to only test splice junctions (not exonic regions), and to use the simple maximum of the unshared and fitted dispersions (as opposed to the maximum *a posteriori* dispersion) for hypothesis testing. This set of parameters is much more conservative, with a much lower false discovery rate in simulated analyses but also substantially reduced statistical power. As expected, using this parameter set JunctionSeq reports less-significant p-values for most of the tests (compared with table 2 from the main text). However, JunctionSeq is still able to detect the genes in the majority of cases.

1.31 Supplemental Table 3: Summary of JunctionSeq results in the rat pineal dataset using the splice-junction-only and no-MAP options.

Adjusted p-value threshold	In vivo			In vitro			Overlap, All Four	No Stimulus (*in vivo*)		
	Ctrl Day/Night	Sham Day/Night	Overlap, in vivo	CN vs NE	CN vs DBcAMP	Overlap, in vitro		SCGX Day/Night	DCN Day/Night	Overlap, No Stimulus
0.01	225	122	80	41	45	28	16	21	4	0
0.001	159	85	58	30	34	20	10	17	3	0
0.0001	119	66	49	22	25	16	8	15	2	0
0.00001	98	55	37	16	19	12	5	11	2	0
0.000001	75	41	25	13	15	8	3	10	2	0

This table is analogous to table 1 in the main text, except that JunctionSeq was run with the options: use.exons = FALSE and method.dispFinal = "max". These options cause JunctionSeq to only test splice junctions (not exonic regions), and to use the simple maximum of the unshared and fitted dispersions (as opposed to the maximum *a posteriori* dispersion) for hypothesis testing. This set of parameters is much more conservative, with a much lower false discovery rate in simulated analyses but also substantially reduced statistical power.

2 Statistical Methodology

2.1 Model framework

Each exonic region or splice junction locus (or "feature") is fitted to a separate model. All terms from here forward are relative to a specific feature j, which is an exonic region or splice junction locus on gene g. Let there be n biological replicates (or "samples").

For this feature j, we define two "counting bins": $\vec{\boldsymbol{y}}_1 = (y_{11}, y_{12}, \ldots, y_{1n})$ and $\vec{\boldsymbol{y}}_0 = (y_{01}, y_{02}, \ldots, y_{0n})$.

The counts y_{1i} are defined as the number of reads (or read-pairs) in sample i that cover the feature j. That is, the number of reads or read-pairs that intersect with the exonic region (if j is an exon) or align over the splice junction (if j is a splice junction). The counts y_{0i} are defined as the number of reads (or read-pairs) that intersect with the gene g, but that do **not** cover the specific feature j. The gene-level counts are calculated using the standard HTSeq-based method used for DESeq2/edgeR differential gene expression.

Note: for the exon/junction counts, only read-pairs that actually align to the junction or exon itself are counted, read-pairs that only *flank* a feature are NOT counted towards that feature.

In the framework used by DEXSeq 1.12.2 (which, it should be noted, differs slightly from the original framework used by some earlier versions and presented in the DEXSeq methods paper) the feature counts (y_{1i}) are compared with the sum of all other feature counts belonging to the same gene. This means that some reads may be counted more than once if they span multiple features. When reads are relatively short (as was typical when DEXSeq was first introduced) this effect is minimal, but it becomes progressively less precise for longer reads (see Supplemental Table 6). While, in theory, the methods used by DEXSeq should be robust against this issue, under certain circumstances it can result in unexpected artifacts. For example, if a gene has a large number of features in a very small genomic area (for example, if an exon has numerous alternative donor/acceptor sites), then reads covering that region may be disproportionately over-weighted, altering the relative fold change estimates by warping the linear contrasts.

These counts can be generated via QoRTs, a freely-available and open-source software package that provides both QC and data processing for RNA-Seq datasets (1).

The gene and exon counts could theoretically be generated by HTSeq, but there is currently no HTSeq-based method of generating the JunctionSeq input files which also include counts for known and novel splice junctions.

2.1.1 Comparison with the DEXSeq model framework:

Under our framework, no read is ever counted more than once within a single model frame. Individual reads are still counted towards multiple exons and splice junctions, and may appear in multiple separate hypothesis tests.

To compare *DEXSeq* and *JunctionSeq* model frameworks, let us define the raw counts k as:

$$k_{gij} = \#\text{ reads or read-pairs that cover exonic region or junction } i \text{ of gene } g \text{ in sample } j$$

And:

$$k_{gj} = \#\text{ number of reads or read-pairs that cover } ANY \text{ exons of gene } g \text{ in sample } j$$

For *JunctionSeq* and *DEXSeq* both, the elements of counting bin $\vec{y}_1 = (y_{11}, y_{12}, \ldots, y_{1n})$ are the same, defined for each sample j as:

$$y_{1j} = k_{gij}$$

However, the counting bin $\vec{y}_0$ is different in *JunctionSeq* and *DEXSeq*. In JunctionSeq, the elements of $\vec{y}_0$ are defined:

$$y_{0j} = k_{g1} - k_{gij}$$

Whereas for *DEXSeq* the elements of $\vec{y}_0$ are defined:

$$y_{0j} = \left(\sum_{x\ in\ gene\ g} k_{gx1} \right) - k_{gij}$$

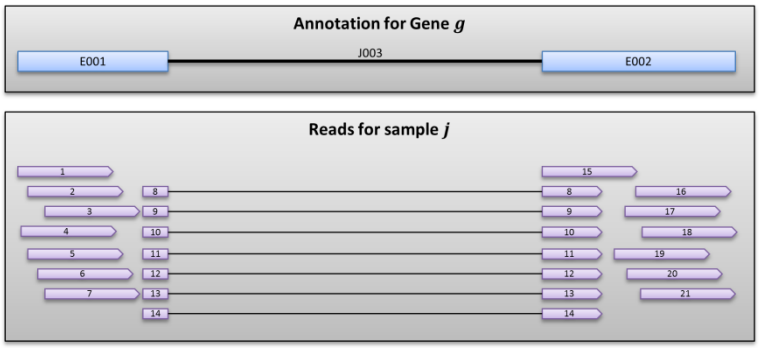
Where the left term sums across all features x that belong to gene g.

Note that, in general:

$$\sum_{x\ in\ gene\ g} k_{gxj} \geq k_{gj}$$

Because the left term multi-counts any reads that span multiple features belonging to g. See Supplemental fig. 29 for a visual representation of this difference.

Supplemental fig. 29: Diagram comparing the read-pair counting methods of JunctionSeq and DEXSeq (for hypothetical sample *j*, gene *g*)



Raw Count Table:

k	Counts
k_{g1j}	14
k_{g2j}	14
k_{g3j}	7
k_{gj}	21

Count Vectors:

Model Frame	Element	Counts: DEXSeq (Std)	Counts: DEXSeq (Exons + SJ)	Counts: JunctionSeq
E001	y_{1j} =	14	14	14
	y_{0j} =	14	21	7
E002	y_{1j} =	14	14	14
	y_{0j} =	14	21	7
J003	y_{1j} =	-	7	7
	y_{0j} =	-	28	14

Note that in the standard *DEXSeq* counting method, sample *j* contributes 28 read-pairs to each model matrix, even though there are only 21 read-pairs present. This is because reads 8-14 are counted twice, as they cover both exons E001 and E002. Further note that this issue is even more extreme when *DEXSeq* is run on exon and splice junction data, with a total count of 35 in each model frame.

Note however that under the *JunctionSeq* design, many reads are still counted towards more than one separate model frame. Reads 8 through 14 are counted in the y_{1j} value in the model frames for all three features E001, E002, and J003. It is only within a single given model frame that reads are not multi-counted.

The percentage of reads that are double- or triple-counted will vary considerably from experiment to experiment depending on the study condition, organism, and read length. Very short reads will produce less multi-counting and long reads will produce more. Similarly, organisms with very large exons and/or relatively few known splice variants (such as *drosophila*) will have far fewer splice junctions. In the rat pineal data, for example, the sum of exonic counts is 1.6 times higher than the number of exonic reads, whereas in the human data it is almost double (see Supplemental Table 7).

2.2 Novel splice junctions

In addition to the known splice junctions provided by a transcript annotation, novel splice junctions detected by the alignment software can be added to the analysis. These unannotated junctions can be selected for inclusion in the analysis if they had at least one endpoint within the span of a single gene and if they had a mean-normalized read-pair coverage (across all conditions) of greater than an assigned threshold (the default is 6 read-pairs per biological replicate).

2.3 Statistical Model

As in DEXSeq, we assume that the count y_{bi} is a realization of a negative-binomial random variable Y_{bi}:

$$Y_{bi} \sim NB\big(mean = s_i \mu_{bi}, dispersion = \alpha_j\big)$$

Where α_j is the dispersion parameter for feature j, s_i is the normalization size factor for sample i, and μ_{bi} is the normalized mean for sample i and counting-bin b. (Remember that $(b = 1)$ refers to the feature (ie. exon/splice-junction) counts and $(b = 0)$ refers to the gene counts minus the feature counts.)

The normalization size factors s_i are estimated using the "geometric" normalization method, which is the default method used by DESeq, DESeq2, DEXSeq, and CuffDiff (2-5). By default these normalizations are performed based on the gene-level counts.

2.4 Dispersion estimation

In many high-throughput sequencing experiments there are too few replicates to directly estimate the locus-specific dispersion term α_j for each feature j. This problem is well-characterized, and a number of different solutions have been proposed, the vast majority of which involve sharing information between loci across the genome (2,6). JunctionSeq uses the same method used by the DESeq2 package and by the more recent releases of the DEXSeq package. This method is described in detail elsewhere (7).

Briefly: individual-feature estimates of dispersion are generated via a Cox-Reid-based method. Then, a parametric model is fitted to these dispersions:

$$\alpha(\mu) = \alpha_0 + \alpha_1 \frac{1}{\mu}$$

Where μ is the “base mean”, or the sum of the normalized counts across the feature. The model fit coefficients, α_0 and α_1, are estimated using generalized linear models using the method described in (7). The final dispersions are based on the maximum *a posteriori* (MAP) estimator, which combines information from the feature-specific and fitted dispersion estimates. The JunctionSeq R package also implements a number of other optional methods for estimating the dispersion (see Additional File 1).

2.4.1 Separate fitting of exon and splice junction dispersion trends

It is necessary to fit separate dispersion trends to exonic regions and splice junctions. This necessity is clear even in simulated data, which suffers from fewer artifacts and biases compared with real data.

To demonstrate this, we ran a model fit based on the same model indicated above, but with two additional terms to allow exonic regions to differ from splice junctions in both the intercept and the slope of the trend:

$$\alpha(\mu) = \alpha_0 + \alpha_1 \frac{1}{\mu} + \alpha_{isExon}(X) + \alpha_{exonSlope}(\frac{1}{\mu} * X)$$

Where "X" equals 1 when the feature is an exon and 0 otherwise. The summary table for this generalized linear model is shown below. Note that the model fit parameters $\alpha_{isExon}$and $\alpha_{exonSlope}$ terms are highly significant, suggesting that exons and splice junctions do not follow the same dispersion distribution.

Note: JunctionSeq does NOT use the above model. The results below are simply shown to illustrate the trends that motivated the separate fitting of different feature types. Instead JunctionSeq fits entirely separate models for exonic regions and splice junctions. Separate model fitting makes even fewer assumptions about similarity and comparability between exons and splice junctions: for example, it does not assume that the dispersions of the dispersion estimates are the same.

Supplemental Table 4: Dispersion model fit table, including terms for exon status, from the first "Null" simulated dataset.

Parameter	Estimate	Std. Error	Test statistic	p-value
α_0	5.767e-04	2.200e-05	26.22	<2e-16
α_1	4.303e-01	2.412e-03	178.39	<2e-16
α_{isExon}	7.137e-04	3.188e-05	22.39	<2e-16
$\alpha_{exonSlope}$	1.644e-01	3.702e-03	44.42	<2e-16

2.5 Hypothesis tests for differential usage (DU)

The hypothesis test is performed using the same DESeq2-based methods used in DEXSeq v1.14.0, which are described in detail elsewhere (7). Note that the methods used in (7) include numerous improvements on the methods originally described in the DEXSeq methodology paper (3).

Briefly: two models are fitted to the mean μ_{bi}. First, the reduced (null hypothesis) model:

$$\log(\mu_{bi}) = \beta + \beta_b^{Bin} + \beta_i^{Samp}$$

And then the alternative model:

$$\log(\mu_{bi}) = \beta + \beta_b^{Bin} + \beta_i^{Samp} + \beta_{\rho_i b}^{Cond * Bin}$$

Where ρ_i is the experimental condition value for sample i.

Note that the experimental-condition/counting-bin interaction term ($\beta_{\rho_i b}^{Cond*Bin}$) is included, however, the experimental-condition main-effects term ($\beta_{\rho_i}^{Cond}$) is NOT included in this model, as this term is absorbed by β_i^{Samp}.

There are two components that can be treated as "noise": variation in junction-level expression and variation in gene-level expression. As proposed by Anders et. al. (3), in the model above we use a main-effects term for the sample ID (β_i^{Samp}), which subsumes the condition main-effect term. This subsumes both gene-level differences and random variation (noise) in the gene-level expression, improving the power for detecting differential interaction between the count-bin term and the experimental-condition term.

The reason that we design our model framework in this way is because it improves our statistical power, and because *JunctionSeq* is not designed to detect or assess gene-level differential expression (and thus we are not interested in testing $\beta_{\rho_i}^{Cond}$). See Supplemental Table 5 for an example model matrix, showing both the model matrix used by *DEXSeq* and *JunctionSeq*.

Supplemental Table 5: Example Model Matrix for an example 2x2 analysis. The model matrix shown here is only used to test for differential usage of the specific feature *i*.

	(Int)	Samples				Bo	C*B	*DEXSeq* Model Vector	*JunctionSeq* Model Vector
		S1	S2	S3	S4				
S1_B1	1	1	0	0	0	0	0	$y_{11} = k_{gi1}$	$y_{11} = k_{gi1}$
S2_B1	1	0	1	0	0	0	0	$y_{12} = k_{gi2}$	$y_{12} = k_{gi2}$
S3_B1	1	0	0	1	0	0	1	$y_{13} = k_{gi3}$	$y_{13} = k_{gi3}$
S4_B1	1	0	0		1	0	1	$y_{14} = k_{gi4}$	$y_{14} = k_{gi4}$
S1_B0	1	1	0	0	0	1	0	$y_{01} = \left(\sum_{x \text{ in gene } g} k_{gx1} \right) - k_{gi1}$	$y_{01} = k_{g1} - k_{gi1}$
S2_B0	1	0	1	0	0	1	0	$y_{02} = \left(\sum_{x \text{ in gene } g} k_{gx2} \right) - k_{gi2}$	$y_{02} = k_{g2} - k_{gi2}$
S3_B0	1	0	0	1	0	1	0	$y_{03} = \left(\sum_{x \text{ in gene } g} k_{gx3} \right) - k_{gi3}$	$y_{03} = k_{g3} - k_{gi3}$
S4_B0	1	0	0	0	1	1	0	$y_{04} = \left(\sum_{x \text{ in gene } g} k_{gx4} \right) - k_{gi4}$	$y_{04} = k_{g4} - k_{gi4}$

2.6 Parameter Estimation:

While the statistical model described above is robust, efficient, and powerful, it lacks main effect terms and thus cannot be effectively used to estimate the size of the differential effect.

For the purposes of estimating the effect sizes and expression levels we create a separate set of generalized linear models for each feature. In this we diverge substantially from the current DEXSeq methods.

The mean μ_{bi} is modeled as:

$$\log(\mu_{bi}) = \beta + \beta_b^{Bin} + \beta_{\rho_i}^{Cond} + \beta_{\rho_i b}^{Cond * Bin}$$

This model is used to calculate the parameter estimates $\hat{\beta}$, $\hat{\beta}_b^{Bin}$, $\hat{\beta}_{\rho_i}^{Cond}$, and $\hat{\beta}_{\rho_i b}^{Cond * Bin}$, which are then used to calculate mean normalized coverage estimates $\hat{\mu}_\rho$ for each condition value ρ. Using linear contrasts, relative expression estimates can also be calculated for each junction locus and each condition value, producing an estimate of relative fold-change. This differs from the DEXSeq methodology, which fits one large model to the full set of features belonging to the given gene.

2.7 Multivariate Models:

If needed, additional covariates can be integrated into the generalized linear models. If we define τ_i as the covariate category for sample i, then we can define the hypothesis test reduced model as:

$$\log(\mu_{bi}) = \beta + \beta_b^{Bin} + \beta_i^{Samp} + \beta_{\tau_i b}^{T * Bin}$$

And the alternative hypothesis model:

$$\log(\mu_{bi}) = \beta + \beta_b^{Bin} + \beta_i^{Samp} + \beta_{\tau_i b}^{T * Bin} + \beta_{\rho_i b}^{Cond * Bin}$$

Like the main effects term for the condition variable, the main effects term for the confounding variable is absent from both of the models used in the hypothesis test. This main effects term is subsumed into the sample-ID term β_i^{Samp}.

The parameter-estimation model can be extended similarly:

$$\log(\mu_{bi}) = \beta + \beta_b^{Bin} + \beta_{\rho_i}^{Cond} + \beta_{\tau_i}^{T} + \beta_{\tau_i b}^{T * B} + \beta_{\rho_i b}^{Cond * Bin}$$

In general the expression estimates are generated by averaging over confounding variable status. Optionally the expression estimates could be generated for each confounder status separately.

3 Test Dataset Data Processing and Methods

Supplemental Table 6: Various summary statistics for the species and annotations used in the three test datasets. Note that there is considerable variation in the median exon length, number of known transcripts, number of known splice junctions, and the number of known exons. Also note that although the exon lengths of rat and human are approximately equal, the exonic region length is considerably smaller in the human annotation, due to the more comprehensive transcript annotations.

	T. gondii	Rat	Human
# Known genes	17,274	25,116	27,436
# Known Transcripts	25,734	57,135	156,848
# Distinct genes or aggregate genes	8,615	24,989	25,918
# Known exonic regions	47,712	230,986	508,682
# Known splice junctions	39,055	196,865	308,888
Median exon size (bp)	165	130	127
Median size of “exonic regions” (bp)	165	129	96

Supplemental Table 7: Various summary statistics for the three testing datasets. Note that the sum of all known exonic regions is substantially larger than the total number of reads covering known genes, as many reads will span multiple features. This effect would be made substantially worse with the inclusion of splice junctions in the model frameworks (*Novel splice junctions were only included if the mean normalized read count exceeded 3 and if they could be uniquely linked to a single specific gene.)

	T. gondii	Rat Pineal	Human, Simulated
Read length	100	100	126
# Novel splice junctions*	2,190	11,646	513
Avg. # mapped reads	15,019,042	41,959,794	26,956,206
Avg. # exonic reads	9,561,498	15,005,609	25,915,805
Avg. # intronic reads	542,354	14,084,198	10,602
Avg. sum of known exonic region read counts	13,009,980	24,015,288	50,948,167
Avg. sum of known splice junction read counts	3,118,001	8,234,189	14,536,246
Avg. sum of novel splice junction read counts	84,921	188,322	7,422

3.1 Toxoplasma gondii

Our first real test dataset originated from a previous study in which alternative splicing was detected and validated in *Toxoplasma gondii* between control samples and samples in which overexpression of the TgSR3 gene was induced (8). There were four sample groups of 3 biological replicates each: untreated; induced, 4 hours; induced, 8 hours; and induced, 24 hours. The dataset is available on the NCBI short read archive (SRA), accession number PRJNA252680.

Reads were realigned with *RNA-STAR* (9) to the ToxoDB v25 Toxoplasma gondii GT1 genome build and annotation (10). The dataset was processed via the *QoRTs* (1) data processing package and analyzed with *JunctionSeq*. Unlike with the previous study’s analysis, a full *CuffLinks* transcript assembly was unnecessary as *JunctionSeq* can test novel splice junctions of known genes. We ran three analyses comparing each of the three induced groups to the untreated group.

3.2 Circadian Rhythms in the Rat Pineal Gland

Our second test dataset consisted of 7 sample groups, 4 taken from live *Rattus norvegicus* pineal glands and 3 taken from rat pineal glands in organ culture, with 3 biological replicates each (21 biological replicate, total). The dataset is available online on the NCBI short read archive (SRA), accession number PRJNA267246. The surgical methods, sample collection, and sequencing is described in detail elsewhere (11). Briefly: the 12 *in vivo* samples were taken from no-surgery

(Ctrl) and sham-surgery (Sham) rats at two time points: night and day. The 9 *in vitro* samples consisted of pineal glands in organ culture treated with norepinephrine (NE) or dibutyryl cyclic AMP (DBcAMP, an analogue of the second messenger, cAMP), as well as untreated controls (CN).

The 21 samples were aligned to the rn6 rat genome build using the *RNA-STAR* aligner (9) with the ensembl transcript annotation (release 80). Gene, exon, and splice-junction read-pair counts were then generated by the *QoRTs* data processing utility (1), adding novel splice junctions if they could be matched to a single known gene and if their mean normalized coverage across all samples exceeded 3. The five analyses were then carried out using *JunctionSeq* and *DEXSeq* using the standard developer-recommended options. The adjusted-p-value threshold for all plots was set to p-adjust < 0.01, which is the default for *JunctionSeq*.

3.2.1 Incomplete Annotation

To demonstrate *JunctionSeq's* ability to detect differential usage in novel splice junctions even with an incomplete transcript assembly, we performed a second set of analyses with a reduced annotation. For each of the four known AIR genes (Crem, Pde4b, Slc15a1, and Atp7b), we manually removed all but one transcript from the ensembl annotation GTF and then re-ran the analyses. Since the gene Slc15a1 only has one known transcript, it was not altered in this analysis.

In general it was not possible to uniquely identify the dominant isoform from each gene. When selecting which isoform to leave in the annotation, we chose one of the most likely dominant isoforms based on the exon and junction coverages found by *JunctionSeq*. In truth it does not affect the validity of the experiment, since in practice the annotated isoform may be any one of the true isoforms (or in some cases even be a false isoform that does not actually exist). See Supplemental Table 8 for a full listing of the isoforms used in the partial annotation analysis.

All genes/transcripts other than the ones belonging to the four known AIR genes were included in the analysis as normal. They had to be included because *JunctionSeq* shares information across genes in order to estimate the size factors and biological dispersion.

Supplemental Table 8: Genes and transcripts selected for use in the "incomplete annotation" analysis.

Gene Symbol	# Known Transcripts	Gene Ensembl ID	Transcript used in the "incomplete" annotation
Atp7b	2	ENSRNOG00000012878	ENSRNOT00000089265
Crem	7	ENSRNOG00000014900	ENSRNOT00000074146
Pde4b	3	ENSRNOG00000005905	ENSRNOT00000007738
Slc15a1	1	ENSRNOG00000011598	ENSRNOT00000015890

It should be noted that the adjusted-p-values listed in the "full" and the "incomplete" analyses for the gene Slc15a1 were slightly different, despite the fact that the gene annotations were unchanged for this gene. This was due to (very slight) analysis-wide differences in the two sets of analyses. *JunctionSeq* and *DEXSeq* both share information between genes to calculate size factors and dispersion estimates. Furthermore, the FDR-based Benjamini-Hochberg multiplicity correction procedure causes the adjusted-p-values of each test to depend on the other p-values in each set (12). Thus, changes to one gene can (usually only slightly) change the results for the other genes.

3.3 Simulations Analysis

3.3.1 Data Simulation Methods:

Simulated data was generated using the *RSEM* (13) simulator based on the Ensembl human genome annotation, release 74 (hg19). Gene expression levels were generated randomly by drawing (with replacement) from the base mean expression levels found by DESeq2 in the rat pineal untreated vs norepinephrine-treated dataset, then re-adjusted to a mean total count of 30 million read pairs. Transcript fractions were randomly assigned by normalizing the output of a uniform distribution. In addition: unlike the simulations in (14), we did not use a Dirichlet distribution to randomize transcript fractions between samples. The negative-binomial distribution has a much stronger theoretical basis: assuming sequencer priming is a Poisson process, read coverage across a static interval will yield a Poisson distribution, and additional biological (replicate-to-replicate) variation will introduce "over dispersion" which can be effectively modelled using a gamma distributed mean parameter resulting in a negative binomial distribution (2,7,15-17). There is no real

credible reason to expect observed transcript fractions to follow a Dirichlet distribution, and most transcript quantification tools assume some form of negative binomial distribution.

3.3.2 Simulated Datasets

A total of twelve samples were generated with an average of 30 million read-pairs each and a read length of 126 base pairs. Six samples were "cases", and six were "controls". Differential expression was simulated in 500 genes by increasing or decreasing mean expression of all transcripts simultaneously, and differential transcript usage was simulated in 250 genes by increasing or decreasing mean expression of one transcript chosen at random. Differential effects of either form were restricted to genes with an expected read count above 25 and at least two different annotated isoforms. Natural-log-fold-change values were drawn from a uniform distribution between -2 and 2, generating fold changes between 1 and 7.38 in either direction. The expected counts in the "case" samples were determined via simple multiplication by this randomly-generated factor, or with the factor inverted if the resultant expectation would be above 10,000 or below 1.

An "incomplete" annotation was generated by removing all but the most highly expressed transcript from each of the 250 "differential transcript usage" genes. We elected to only remove transcripts from the DU genes so as to not introduce systematic analysis-wide differences between the two datasets that could alter dispersion estimation or p-value estimates. We used this "incomplete" annotation to test each method's ability to detect differential effects in genes that are poorly annotated.

3.3.3 Simulations analysis

The simulated samples were run through the same analysis pipeline as the real datasets. Reads were aligned to hg19 via the RNA-STAR aligner (9) and counts were generated via QoRTs (1).

JunctionSeq, *DEXSeq,* and *CuffDiff* were then run both with the full annotation and with the incomplete annotation. To improve comparability, the *DEXSeq* analysis excluded all aggregate genes, as this has been found to improve performance (14) and is the default behavior in *JunctionSeq*. For *CuffDiff*, one of the standard pipelines was applied: aligned reads were quantified and assembled via *CuffLinks*, the assemblies merged via *CuffMerge*, and the final differential usage analyzed with *CuffDiff*.

Gene-level adjusted p-values were used for defining "detection" in *JunctionSeq* and *DEXSeq*. For *CuffDiff*, a slightly different strategy was required, as CuffDiff uses different definitions of differential-splicing/DTU/AIR. The gene-level adjusted-p-value for each gene was defined as the minimum adjusted p-value found in any test connected to the given gene across all three isoform-level differential analysis results files: "cds.diff", "promoters.diff", and "splicing.diff". The other analysis files (ending with the suffix "exp.diff") only test for simple differences in expression levels, not cross-isoform differences and thus were ignored.

Note that the true positive rates and false discovery rates reported here are substantially worse than those reported in (14), even for the analysis tools shared by both simulation analyses. This is due to differences in the simulation procedure, and in particular in the number and effect strength of the AIR genes. The previous publication's dataset simulated 1000 AIR genes and restricted the selection of these genes to the set of genes with unusually high expression (expected_count > 500). Our simulations set had much fewer AIR genes (250), and had lower expression requirements (expected_count > 25). In addition, instead of simply switching the transcript fractions of the top two genes (which can produce enormous fold changes for highly-dominant transcripts), we randomly generated effect sizes from a spectrum. This produced rarer and weaker differential effects that were substantially more difficult to detect, resulting in substantially lower detection accuracy.

In particular, in the previous report *CuffDiff* was able to accurately control false discoveries, whereas in our datasets *CuffDiff* had very high false discovery rates. In both experiments *CuffDiff* returned very few statistically significant genes, but in our analysis much fewer of those genes were true positives.

Note: in the ROC/FDR plots the gene sets were restricted to only the genes that are distinct (non-overlapping) and highly expressed (expected_count > 25), and that were unambiguous in the *CuffLinks* assembly. This produced a set of 9465 non-AIR genes and 223 AIR genes. This ensured cross-comparability between methods and eliminated a number of artifacts that would produce misleading ROC curves. For example: the inclusion of all genes would artificially inflate the

right side of the ROC curves, since simply "detecting" all genes that have >25 reads would discriminate AIR genes better than random chance. The remaining genes were still included in the actual analysis, as they still contribute to the estimation of dispersions and size factors. Similarly: including non-distinct genes would make the different analyses non-comparable, since the different methods are based on conflicting definitions of "alternative isoform regulation" under these rare circumstances.

3.4 JunctionSeq Alternative Parameters

We tested several different optional parameters against one another. See the table below for the optional parameters passed to JunctionSeq in each analysis.

Supplemental Table 9: Table of alternative parameters tested with JunctionSeq.

Variant Title	Optional Parameter(s)	Description
JunctionSeq (STD)	-	Standard run with default options. Tests both exons and splice junctions (SJ). Uses maximum *a posteriori* dispersion.
JunctionSeq (Exons)	use.junctions=FALSE	Does not test splice junctions.
JunctionSeq (SJ)	use.exons=FALSE	Does not test exons.
JunctionSeq (Both + NoMAP)	method.dispFinal="max"	Uses simple maximum of unshared and fitted dispersions.
JunctionSeq (Exons + NoMAP)	method.dispFinal="max", use.junctions=FALSE	Uses simple maximum of unshared and fitted dispersions. Does not test splice junctions.
JunctionSeq (SJ + NoMAP)	method.dispFinal="max", use.exons=FALSE	Uses simple maximum of unshared and fitted dispersions. Does not test exons.

4 References:

1. Hartley, S.W. and Mullikin, J.C. (2015) QoRTs: a comprehensive toolset for quality control and data processing of RNA-Seq experiments. *BMC bioinformatics*, **16**, 224.

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

3. Anders, S., Reyes, A. and Huber, W. (2012) Detecting differential usage of exons from RNA-seq data. *Genome research*, **22**, 2008-2017.

4. Trapnell, C., Hendrickson, D.G., Sauvageau, M., Goff, L., Rinn, J.L. and Pachter, L. (2013) Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature biotechnology*, **31**, 46-53.

5. Dillies, M.A., Rau, A., Aubert, J., Hennequet-Antier, C., Jeanmougin, M., Servant, N., Keime, C., Marot, G., Castel, D., Estelle, J. *et al.* (2013) A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in bioinformatics*, **14**, 671-683.

6. Robinson, M.D., McCarthy, D.J. and Smyth, G.K. (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139-140.

7. Love, M.I., Huber, W. and Anders, S. (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol*, **15**, 550.

8. Yeoh, L.M., Goodman, C.D., Hall, N.E., van Dooren, G.G., McFadden, G.I. and Ralph, S.A. (2015) A serine–arginine-rich (SR) splicing factor modulates alternative splicing of over a thousand genes in Toxoplasma gondii. *Nucleic acids research*, **43**, 4661-4675.

9. Dobin, A., Davis, C.A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M. and Gingeras, T.R. (2013) STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, **29**, 15-21.

10. Gajria, B., Bahl, A., Brestelli, J., Dommer, J., Fischer, S., Gao, X., Heiges, M., Iodice, J., Kissinger, J.C. and Mackey, A.J. (2008) ToxoDB: an integrated Toxoplasma gondii database resource. *Nucleic acids research*, **36**, D553-D556.

11. Hartley, S.W., Coon, S.L., Savastano, L.E., Mullikin, J.C., Program, N.C.S., Fu, C. and Klein, D.C. (2015) Neurotranscriptomics: The Effects of Neonatal Stimulus Deprivation on the Rat Pineal Transcriptome. *PloS one*, **10**, e0137548.

12. Benjamini, Y. and Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B*, **57**, 289-300.

13. Li, B. and Dewey, C.N. (2011) RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC bioinformatics*, **12**, 323.

14. Soneson, C., Matthes, K.L., Nowicka, M., Law, C.W. and Robinson, M.D. (2016) Isoform prefiltering improves performance of count-based methods for analysis of differential transcript usage. *Genome biology*, **17**, 1.

15. Robinson, M.D. and Smyth, G.K. (2007) Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, **23**, 2881-2887.

16. Robinson, M.D. and Smyth, G.K. (2008) Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*, **9**, 321-332.

17. McCarthy, D.J., Chen, Y. and Smyth, G.K. (2012) Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic acids research*, **40**, 4288-4297.