Supplementary material for: Intron Length and Recursive Sites are Major Determinants of Splicing Efficiency in Flies

Athma A. Pai, Telmo Henriques, Joseph Paggi, Adam Burkholder, Karen Adelman, and Christopher B. Burge

February 10, 2017

Contents

1 Introduction

In the following, we repeat some of the text from the Methods section of the main manuscript. We have chosen to provide those sections here in order to have a complete, uninterrupted description of the study design, bioinformatics pipeline, and statistical analyses in one file. All analyses were done with a combination of custom python and R scripts, using appropriate bioinformatics packages as noted.

2 Calculating splicing rates

2.1 Estimating splicing using junction reads exclusively

As an alternative method for calculating Ψ values, we used only reads crossing exon-exon or intron-exon junction regions for each intron. Each read was considered to be a junction read if it had a 10 bp overlap on either side of the junction point. We then calculated Ψ values as:

$$\Psi_{intron} = \frac{\frac{1}{2}(a+b)}{\frac{1}{2}(a+b)+c}$$

where: (1) a is the read density for the exon-intron boundary at the $5'$ splice site, (2) b is the read density for the exon-intron boundary at the $3'$ splice site, and (3) c is the read density for the exon-exon boundary. If there were a substantial proportion of reads deriving from spliced intronic lariats in the libraries, we would expect an inflation of MISO-derived Ψ values (which incorporate intronic read density) relative to Ψ values based on junction reads along. In fact, we observed the opposite effect, where MISO-derived Ψ values were generally lower than junction read derived Ψ values, suggesting minimal contribution from intron lariats, as expected given that lariats are expected to decay in seconds after splicing.

2.2 Simulations to assess our accuracy in estimating splicing half-lives

To investigate biases in our measurements of splicing half-lives, we simulated Ψ values from a range of half-lives and assessed our ability to accurately estimate the simulated half-lives. Specifically, we simulated Ψ values from 5, 10, and 20 minute timepoints, from 10,000 introns with half-lives between 0.1-1,000 minutes and 10,000 introns with half-lives between 0.5-20 minutes. Our simulations closely matched the 4sU-seq dataset, with 3 replicates simulated per time-point. For each time-point, we added jitter to the Ψ values across the three replicates to match the mean variance across replicates in the true data. If the simulated Ψ value from any time-point was below 0, the value was adjusted to 0; similarly, if the simulated Ψ value was above 1, the value was adjusted to 1. The simulated Ψ values were used in the exponential model described in the Methods to estimate half-lives and compare to the true half-lives from which the simulated Ψ values were derived.

2.3 Correcting for intron length.

In order to estimate splicing half-lives that are corrected for intron lengths in Supplementary Figure 8B, we estimated a length-specific median half-life in each of 50 bins across the distribution of intron lengths. This length-specific median was then subtracted from the estimated half-life to obtain a corrected splicing half-life. For visualization and comparisons purposes, a constant representing the median non-corrected half-life was added to all corrected splicing half-lives.

3 Statistical models to account for variance in splicing rates

3.1 Estimating contribution of intron lengths to variance in splicing half-time

To estimate the extent to which intron length accounted for variance in splicing rates, we fit a linear model of the following form to all introns i:

$$y_i = \beta_0 + \beta_i x_{i1} + \beta_i x_{i2} + \epsilon_i$$

where:

β_0 = a constant for all introns

y_i = intron half-life (log_{10}), in minutes

x_{i1} = intron length (log_{10}), in nt

x_{i2} = indicator variable (0 for intron length < 60 nt, 1 for intron length 60 nt)

ϵ_i = intron-specific error term

We included an indicator variable categorizing introns as either very short (< 60 nt) or longer given the non-linear relationship that we observed between intron length and half-life, in which there is a negative relationship between half-life and length for introns ¡ 60 nt, and a positive relationship for introns 60.

3.2 Estimating contribution of other factors to variance in splicing rates

To estimate the extent to which factors other than intron length accounted for variance in splicing rates, we fit a linear model of the following form to all non-first introns i between 60-70nt:

$$y_i = \beta_0 + \beta_{i1} + \beta_{i2} + \beta_{i3} + \beta_{i4} + \beta_{i5} + \beta_{i6} + \beta_{i7} + \beta_{i8} + \beta_{i9} + \beta_{i10} + \beta_{i11} + \beta_{i12} + \beta_{i13} + \beta_{i14} + \epsilon_i$$

where:

y_i are half-lives (log_{10}) in min

x_{i1} = intron length (log_{10}) in nt

x_{i2} = position of intron in transcript

x_{i3} = gene expression (log_{10}(TPM))

x_{i4} = indicator of presence of enhancer in intron i (based on STARR-seq)

x_{i5} = $5'$ splice site score

x_{i6} = $3'$ splice site score

x_{i7} = length of first intron (log_{10}) in nt

x_{i8} = half-life of first intron (log_{10}) in min

x_{i9} = indicator of presence of enhancer in intron i (based on STARR-seq)

x_{i10} = length of upstream exon (log_{10}) in nt

x_{i11} = length of downstream exon (log_{10}) in nt

x_{i12} = % of A + U nt in intron (excluding splice site regions)

x_{i13} = % of A + U nt in $3'$ region of intron ($-40 : -21$ from $3'$ splice site)

x_{i14} = % of A + U nt in $5'$ region of intron ($+7$ from $5'$ splice site to -41 from $3'$ splice site)

We used the values from this multiple linear regression model to estimate the relative importance of each parameter contributing to variance in splicing rates. To do so, we used the $relaimpo$ package in the R environment [1], which arrives at a relative importance percentage by averaging the sequential sum-of-squares obtained from all possible orderings of the predictors in the model.

4 Identifying sites of recursive splicing

4.1 Motif Scoring

We calculated position weight matrices (PWM) for the intronic portions of *Drosophila* $5'$ and $3'$ splice sites using all annotated splice sites. These weight matrices were then juxtaposed with $3'$ss PWM followed by $5'$ss PWM to create a recursive splice site motif PWM. Individual motif occurrences were scored using a normalized bit score [2]. The bit score for each motif occurrence is defined as the sum across the log probabilities for each nt being drawn from the motif. We calculated normalized scores by subtracting the minimum possible score and dividing by the range of possible bit scores.

4.2 Using splice junction reads and junction spanning read pairs

4.2.1 Extracting recursive splice junction reads

Splice junction reads that span putative recursive junctions provide direct evidence for recursive splicing (Supplementary Figure 3A, top panel). In order to identify such reads, we extracted the coordinates of annotated introns and exon-exon junctions from FlyBase *D. melanogaster* Release 5.57 [**?**]. We aligned the 4sU-RNA-seq reads to the corresponding genome release by using hisat2 [3]. We then extracted reads with an upstream junction matching an annotated $5'$ss and a downstream end mapping to an AGGT that is upstream of the downstream most corresponding annotated $3'$ss with the help of Pysam [4].

4.2.2 Extracting recursive splice junction read pairs

In addition to splice junction reads, read pairs with one end on either side of a recursive splice junction, which we will henceforth refer to as recursive junction spanning read pairs, identify recursive sites. We defined putative recursive junction spanning read pairs as read pairs with a first read aligning close upstream of an annotated $5'$ss and a second read aligning to an intronic region more than 1000 bp downstream of the first read. Additionally, we filtered out read pairs that have an insert length of less than 1000 bp conditioned on completion of an annotated splicing event (excluding cassette exons with an AGGT at their $5'$ end).

Unlike splice junction reads, recursive junction spanning read pairs do not immediately implicate a specific recursive site. Instead, a recursive site must be inferred based on the empirical insert length distribution and genomic sequence information. To do this, we adapted the GEM algorithm, which was originally used to infer protein binding sites from ChIP-seq data [5].

Our modifications to the algorithm and choices for parameters described in the GEM paper are as follows:

1. The probability of a read, r_n, given that there is a recursive site at position m, $P(r_n|m)$, was defined as the probability of observing the implied insert length in the empirical insert length distribution.

2. The prior probabilities of each position being a recursive site, Π_{1-N}, were set such that $\Pi_i \propto \max(0, M(i) - .8)$, where $M(i)$ is the motif score for position i as described above. This function was used to determine prior probabilities that reflect the preference for strong motifs observed in the Duff et al. set [].

3. Recursive splice junction reads were counted within the number of effectively assigned reads in the M-step. This served to impose that sites with support from recursive junction reads are more likely to be recursive sites.

4. The sparsity parameter, α_s, was defined as the number of assigned reads divided by 40.

5. The algorithm converged when prior probability did not changes by more than 10^{-5} between iterations. Upon convergence, read pairs were assigned to a putative recursive site using the MAP estimate.

The modified GEM algorithm was run with all read pairs and splice junction reads pooled together.

4.3 Using sawtooth pattern in reads

Recursively spliced introns contain a distinct "sawtooth" pattern due to the co-transcriptional nature of splicing [6]. The shape of this pattern and a graphical explanation for its origin is shown below in Supplementary Figure S3, bottom panel. In the upper panel, the horizontal lines represent elongating RNAs. Here, we assume that over a population of cells there will be a uniform distribution of RNA extensions. The blacked out sections of these RNAs represent segments that have already been spliced out of the growing RNA and degraded. Assuming efficient, co-transcriptional splicing, the RNA will be spliced shortly after elongating past the point of a $3'$ss or recursive splice site. When RNA-seq is performed, reads are only observed from section of RNA that have not previously been degraded. In the lower half of the same panel, we diagram just these intact (orange) RNAs and see that their density exhibits a linear decay across each recursive segment.

We developed an algorithm to predict recursive splice sites from the presence of a sawtooth pattern in introns. Our algorithm can be broken into three distinct phases: pre-processing of the RNA-seq data, Monte Carlo Markov Chain based inference of the presence of a sawtooth pattern, and the prediction of recursive sites based on the output of our inference and sequence information.

4.3.1 RNA-seq data pre-processing

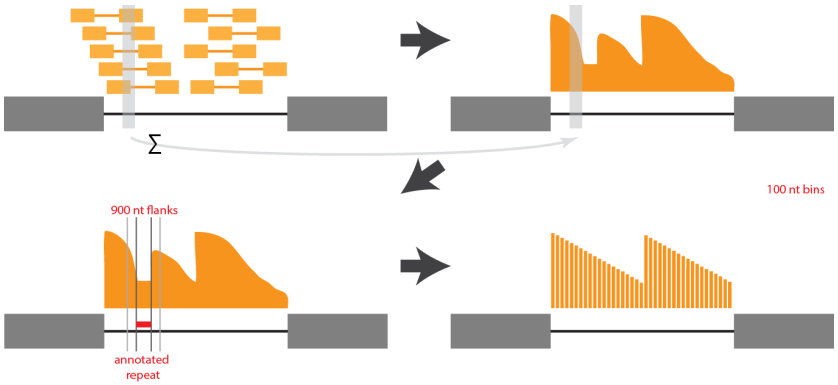


Figure 1: RNA-seq preprocessing converts reads into an array of read densities.

We searched for the presence of a sawtooth pattern in the read distributions of all introns over 8 kb that received at least one splice junction read in any sample. Empirical testing suggested that in introns under 8 kb our method displayed a high rate of false positives, likely due to regression over short segments being more sensitive to noise in read density. Regions annotated as exons were removed using the bedtools subtract command. We summed the number of read pairs aligning to each position to obtain per base coverage counts. When computing this sum, read pairs straddling a given position were counted as aligning there.

In order to avoid erratic read coverage in repeat regions inhibiting our ability to perform meaningful regressions in later steps of this analysis, we masked the read densities in repeat regions. We replaced the read counts in RepeatMasker annotated repeat regions [7] and the 100 flanking nucleotides with the median read density in the 900 nt flanking on either side. This length was chosen because it was short enough that read densities in this range should be comparable to those in the masked region, but long enough to avoid unneeded sensitivity to noise in read density.

In order to attain additional smoothing and reduce the time required to perform the regres-

sions in the next step of our analysis, we separated introns into 100 nt bins and calculated the average of each bin. Throughout the rest of our analysis, we represented the read density of each intron using arrays of these average values.

4.3.2 Regression

We performed linear regression on all sub-regions of each intron. It was assumed that variance in read density at each position is proportional to the coverage level there. This is justified by the fact that RNA-seq data coverage is intrinsically the sum of Bernoulli random variables. We developed a function to calculate these regressions that made use of the Scipy stats weighted linear regression function [8] as a sub process, which we present in psuedocode.

Note that by $|curW - nextW| \leq 10^{-3}$ we mean to check whether all weights changed by at most 10^{-3}.

4.3.3 MCMC

We developed a Monte Carlo Markov Chain (MCMC) algorithm to detect the presence of a sawtooth pattern in each intron. An MCMC algorithm was the ideal algorithm to efficiently explore the complex sample space encountered when considering a non-fixed number of recursive splice sites. By using this method, as opposed to deterministic methods, we were able to efficiently consider all nucleotides as potential recursive splice sites, not just ones at the center of strong motifs. This allowed us to independently use sequence information to assess the false-positive rate of our method.

In this paragraph, we summarize the general flow of our algorithm, while leaving the details of each step to the subsequent paragraphs. Our algorithm is round based. Entering each round, we have an accepted state, consisting of a set of proposed recursive sites in the intron. In each

Algorithm 1: Heteroscedastic Regression

Data: $A \leftarrow$ Array of RNA-seq density

Result: slope, yInt, and weights for regression

$nextW \leftarrow [1..1]$;

$curW \leftarrow [0..0]$;

while $|curW - nextW| \leq 10^{-3}$ **do**

  $curW \leftarrow nextW$;

  $slope, yInt \leftarrow regression(curW, A)$;

  for *position* $\in intron$ **do**

    $nextW[position] \leftarrow \frac{1}{yInt + position * slope}$;

  end

end

return *slope, yInt, nextW*;

round, a new state is proposed by perturbing the current state. We use a scoring function and transition rules we define below to decide if we wish to accept this proposed state or stick with our current state. This procedure is iterated many times and every so often, we record a sample of the current state. The number of samples recorded in each state is proportional to the probability that the intron is best fit by the model corresponding to that state. Therefore, if we normalize the number of samples recorded of each state by the total number of samples, we attain probabilities that each state is the most accurate model.

There are three classes of perturbations used to propose new states. We describe them below and for each refer you to a visual example in the MCMC figure.

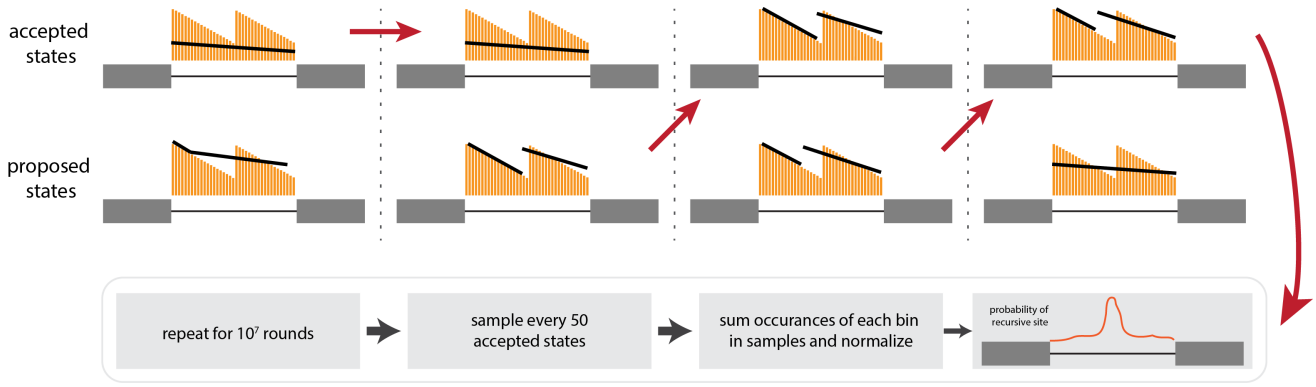


Figure 2: MCMC algorithm infers probability that each position in intron in a recursive splice site.

1. A new recursive site was added with probability .4. Transitions A and B in figure.
2. A recursive site was removed with probability .4. Transition D in figure.
3. A recursive site was slightly perturbed with probability .2. Transition C in figure.

States are scored using a function taking into account how well the corresponding regression fits the observed RNA-seq read density as well as the number of free parameters in the model. At the heart of our scoring function is the Bayesian Information Criteria (BIC).

$$BIC(M) = L * RSS(M) + 2 * (2N) * \log(L).$$

Where $RSS(M)$ is the weighted sum of squared deviations for all recursive segments, L is the intron length, and N is the number of recursive sites. Note that $2N$ is the number of free parameters in the model, as each recursive segment is fit for its own slope and y-intercept.

The score is then given by:

$$Score(M) = \exp(BIC(M)/T)$$

Where T is a constant used to scale the magnitude of the scores. Large values of T result in the algorithm T = 5 was used for all data presented.

In order to constrain our algorithm to fit sawtooth patterns and not more general patterns in the read density, new states are only considered if, at each recursive site, the RNA-seq density predicted by regressions increased by a factor of at least 1.5.

We used the standard transition rules for MCMC inference, which we outline here for convenience. If the score for the new state is lower than the score for the old state, the new state is deterministically adopted. Otherwise the new state is adopted with probability Score(New) / Score(Old). When the old state had zero recursive sites this probability was divided by two to account for the imbalance in transition probabilities.

We chose parameters for burn-in-time, number of iterations and sampling frequency that empirically resulted in consistent convergence across multiple runs of the algorithm. These values were: a burn in of 10^5 iterations, sampling frequency of 50 iterations, and a total of 10^7 iterations.

After all samples were collected, we calculated the probability that each position in the is a recursive site. For each position, we summed the occurrences of that position as a recursive site across all samples. Probability scores were then calculated for each position by dividing this sum by the total number of samples.

4.3.4 Peak Calling

We predicted recursive sites from the MCMC probability scores in a two step process. First, regions with probability above a given threshold (0.08) were recorded. Any of these regions within 500 nucleotides of each other were merged. For each of these regions, a position potential function, P, was defined as 1 inside the peak and flanked by a logistically decaying curve on either side. The logistic function is given by $f(x) = 1/(1 + \exp(-k(x - x_0)))$. The parameters

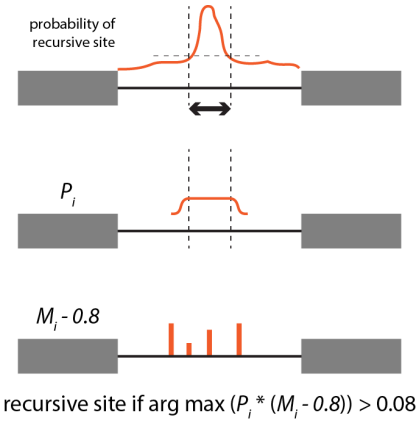


Figure 3: Sequence information is used in conjunction with MCMC-inferred probabilities to predict recursive sites.

were set as $x_0 = 500$ nucleotides from either end and k was set to be 6 / 500 for the left flank and -6 / 500 for the right flank. The resulting distribution has value very close to zero 1000 BP away from peak and a value of 0.5 at a distance of 500 BP. This distribution was chosen based on the empirical performance of the MCMC-based inference when compared to random (SF5C?).

Each AGGT in the intron was then scored by

$$S(i) = P(i) * \max(M(i) - .8, 0)$$

. The maximum scoring AGGT was then reported as a putative recursive site.

4.3.5 FDR Quantification

Shuffled peaks were produced to evaluate the false discovery rate of the sawtooth pattern identification pipeline. For each intron, the initially recorded regions of probability exceeding a set threshold were redistributed with uniform probability across the intron. The length and number of regions was maintained. The remainder of the peak calling procedure was then applied.

4.4 Determining a final set of recursive sites

Out of the final set of recursive sites that we identified, we filtered down to a set of sites based on the following criteria:

1. Sites in genes with TPM ≥ 1 in the total RNA libraries.
2. Sites in introns with at least 3 reads spanning the 5' to 3' splice sites (using the largest annotated intron)

This resulted in a total of 706 recursive sites identified by any method, and 243 high-confidence sites.

When determining high-confidence sites, we followed the protocol used by Duff et al. We wrote a script to iterate through introns and plot the read density and putative recursive sites. We then manually filter each site based on the presence of a recognizable sawtooth pattern.

4.5 Estimating true number of recursive sites

In order to assess the sensitivity of our recursive site detection pipeline, we subsampled our reads to various proportions of the total read coverage and re-assess the number of recursive sites detected. To do so, we used the `samtools view -s` command to subsample each fastq file from all samples to the following fractions: 0.1%, 0.5%, 1%, 2.5%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90%. For each of these subsampled read sets, we re-ran the entire recursive site detection pipeline as described above to assess the number of recursive sites detected.

To assess the impact of gene expression levels on our power to detect recursive sites, we separated long introns into those from lowly expressed genes ($TPM \leq 20$) and highly expressed genes ($TPM > 20$). Using the subset of reads mapping to these genes, we repeated

the subsampling procedure and the entire recursive site detection pipeline described above to characterize the percentage of lowly or highly expressed long introns that have recursive sites.

Finally, to understand whether the lower proportion of lowly expressed introns that have recursive sites is due to technical or biological reasons, we subsampled reads from long intron within highly expressed genes to match the read distribution of a comparable number of long introns from lowly expressed genes. Specifically, we isolated all reads from long introns in highly expressed genes and used a custom python script with the pysam package to randomly subsample these reads to match the distribution of reads from lowly expressed introns. Using this full subset of reads from highly expressed genes, we again repeated the subsampling procedure and the entire recursive site detection pipeline described above to characterize the percentage of highly expressed introns that have recursive sites when reads from these introns are subsampled to a lower read coverage.

4.6 Determining the order of recursive splicing

Previous studies have searched exclusively for recursive junction reads consistent with the 5' to 3' removal of recursive segments. In order to determine if recursive splicing does in fact follow a 5' to 3' order, we implemented a computational search for junction reads consistent with alternative orders of recursive splicing. These reads fall into two categories: junction reads between two intronic AGGTs and junction reads from an intronic AGGT to an annotated 3'ss.

We constrained our search to combinations of recursive sites producing recursive segments of at least 1 KB. Nearly all recursive segments detected in our study were greater than 1 KB in length and adding this constraint filtered out spurious hits likely caused by alignment errors and unannotated splicing events.

We considered all events with support from at least 3 uniquely aligning reads with recursive

splice sites scoring above 0.85 based on the previously described scoring metric. Requiring three uniquely aligning reads matches the cutoff used for our previous analysis. In our previous analysis, we found that recursive splice sites generally have strong motifs that score greater than 0.85.

Our analysis produced thirteen candidate intronic AGGT to annotated 3'ss recursive sites and zero candidate intronic AGGT to intronic AGGT recursive sites. These candidate recursive splice sites were evaluated visually in a genome browser. Two of these sites corresponded to a recursive splice site detected by both our study. One of these sites received 60 recursive junction reads supporting that it is spliced 5' to 3', while only 5 supporting a 3' to 5' ordering. The other 829 and 13, respectively. All other candidate sites did not appear to be true recursive sites, due to either a lack of a sawtooth pattern, low intron expression, or extensive repeats complicating alignments. These data suggest that recursive splicing overwhelmingly, but perhaps not always proceeds in a 5' to 3' order.

5 Splicing efficiency in recursively spliced introns

5.1 Estimating rate of splicing of recursive segments

We quantified splicing rates for each recursive segment independently by mapping the original alignments onto new gene models with the upstream exon, recursive segment and downstream segment contiguous and then running the pipeline described above on these transformed alignments. Specifically, the new gene models consist of a single intron with the same length as the recursive segment flanked by exons with the same lengths as the original up and downstream exons.

Reads were mapped onto these gene models such that whether any other recursive splicing

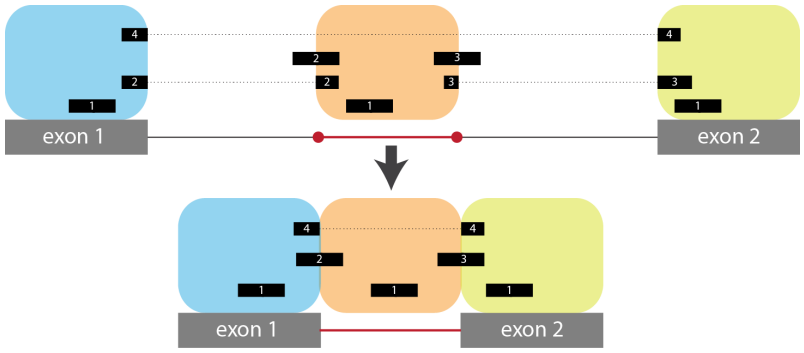


Figure 4: Read from recursively splice

event has yet been completed was irrelevant. These reads were split into four classes, which are numbered the same in the figure as they are below:

1. Reads entirely in the recursive segment or either exon.

   Reads entirely inside any region were directly mapped onto their corresponding region in the new gene model.

2. Reads overlapping the upstream end of the recursive segment.

   Reads can overlap the upstream end of the recursive segment either by being a junction read from the upstream exon to the beginning of the recursive segment or being an unspliced read overlapping the beginning of the recursive segment. In either case, a read was placed at the upstream exon-intron boundary in the new gene model.

3. Reads overlapping the downstream end of the recursive segment.

   Reads overlapping the downstream end of the recursive segment were placed on the intron-downstream exon boundary in the new gene model.

4. Exon-exon junction reads.

In order to be ambivalent to whether all other recursive splicing events have yet been completed all junction reads spanning from the upstream exon to a recursive splice site downstream of the recursive segment or the downstream exons were treated equivalently. All were added as splice junction reads between the upstream and downstream exons in the new gene model.

Any reads not fitting into one of these classes were not included in the new alignments. Notably, this includes reads overlapping either exon-intron junction in the original alignments, reads in recursive segments not being considered, and splice junction reads aligning with an upstream end not at the upstream exon.

5.2 Estimating rate of splicing of full recursive intron

To estimate the mean lifetime of a recursively spliced intron, we estimated the waiting time for all recursive segments to be spliced out by calculating the maximum of the set of individual exponentials from each segment. For one exponential, the mean lifetime $\tau = \frac{1}{\lambda}$ where λ is the coefficient from the exponential fit. There is an analytical solution for estimating the mean lifetime in situations when there are only two exponentials (corresponding to two recursive segments) to be combined. Thus, we conditioned our analyses on recursive introns with only one recursive site, corresponding to the presence of two recursive segments. For these introns, the mean lifetime $\tau_{recursive}$ can be calculated by:

$$\tau_{recursive} = \frac{1}{\lambda_1} + \frac{1}{\lambda_2} + \frac{1}{\lambda_1 + \lambda_2}$$

where λ_1 is the exponential coefficient for the first segment and λ_2 is the exponential coefficient for the second segment. To conservatively compare our recursive intron $\tau_{recursive}$ values with the mean lifetimes of non-recursive introns, we added the time necessary for the first segment

to be transcribed to our $\tau_{recursive}$, with the rationale that the first segment must be completely transcribed before the second can begin to be spliced. Assuming a 1.5kb/min transcription rate, $txn_{seg1} = \frac{l_1}{1500}$, where l_1 is the length of the first segment.

5.3 Accuracy of splicing in recursive introns

We estimated the accuracy of splicing in *Drosophila* introns by identifying non-annotated junction reads with non-canonical splice site sequences within annotated introns. To do so, we first re-mapped the raw 4sU-seq reads with STAR v2.5 mapper [9], with the mapping parameter `--outSAMattribute NH HI AS nM jM` to mark the intron motif category for each junction read in the final mapped file.

The `jM` attribute adds a `jM:B:c` SAM attribute to all reads arising from exon-exon junctions. All junction reads were first isolated and separated based on the value assigned to the `jM:B:c` tag. Junction reads with splice sites in the following categories were considered to be annotated or canonical: (1) any annotated splice site based on FlyBase D. melanogaster Release 5.57 gene structures [`jM:B:c,[20-26]`], (2) intron terminal dinucleotides containing "GT-AG" (or the reverse complement) [`jM:B:c,1` or `jM:B:c,2`], (3) intron terminal dinucleotides containing "GC-AG" (or the reverse complement) [`jM:B:c,3` or `jM:B:c,4`], and (4) intron terminal dinucleotides containing "AT-AC" (or the reverse complement) [`jM:B:c,5` or `jM:B:c,6`]. Junction reads with `jM:B:c,0` were considered to arise from non-canonical non-annotated splice sites. We calculated the frequency of inaccurate splice junctions for each intron as a ratio of the density of reads arising non-canonical non-annotated splice sites to the density of all junction reads from the intron.

6 Legends for Supplementary Figures

Supplementary Figure 1. Calculating rates of splicing in *Drosophila*. **(A)** Proportion of intronic reads in a transcript (Ψ values, *y-axis*) for nascent RNA collected 5min, 10min, and 20min after 4sU labeling, with a time point labeled overnight representing steady-state or total RNA levels. The overall decreases in Ψ values over time indicate increased completed splicing over time. **(B)** The distribution of coefficients from an exponential fit to the Ψ values across time. **(C)** The distribution of R2 values obtained from fitting an exponential model to values across time, with Ψ values estimated using MISO (*blue*) or only junction reads (grey). The preponderance of positive coefficients and high R^2 values indicates that an exponential decay model is appropriate. **(D)** Standard error estimates on half-life propagated from Ψ confidence intervals (see Methods) across a range of splicing half-life times. **(E)** A high concordance between the distribution of Ψ values calculated using only exon-exon and exon-intron spanning junction reads (*x-axis*) vs. the Ψ values calculated using the MISO software, all at 5 minutes after 4sU labeling. **(F)** Splicing half-lives estimated from simulated data (*y-axis*) relative to the simulated half-life (*x-axis*), with splicing half-life range from 2 to 1000 minutes and 0.1 to 20 minutes (*inset*).

Supplementary Figure 2. Properties of splicing efficiency across varying intron lengths. **(A)** Distribution of intron lengths in the *Drosophila melanogaster* genome. **(B)** Distribution of splice site strengths (MaxEnt score, *y-axis*) across both 3′ splice sites (*orange*) and 5′ splice sites (*blue*) for introns between 40-100 nt (*x-axis*). On average, 40-50 nt introns have have weaker splice site scores. **(C)** The distribution of splicing efficiency (half-lives, *y-axis*) for very short 40-50 nt introns (*dark blue*), relative to the distributions for 60-70 nt introns matching for the distributions of 40-50 nt 3′ splice site strength (*light blue*, t-test P = 0.0001), 5′ splice site strength (*light blue*, t-test P = 0.0033), and both 5′ and 3′ splice site strengths (*light blue*, t-test

P = 0.0004). 40-50 nt introns are consistently spliced out slower than other introns, independent of their weaker splice site strengths. **(D)** Distribution of splice site strengths (MaxEnt score, *y-axis*) across both 3′ and 5′ splice sites for introns binned into quantiles of intron length (*x-axis*). **(E)** The distribution of splicing half-lives (*y-axis*) for very long introns greater than 10 kb (*dark blue*), relative to the distributions of 60-70 nt introns matching for the distributions of 10 kb+ 3′ splice site strength (*light blue*, t-test P = 7.093×10^{-14}), 5′ splice site strength (*light blue*, $P < 10^{-16}$), and both 5′ and 3′ splice site strengths (*light blue*, $P < 10^{-16}$). Introns greater than 10 kb in length are consistently spliced more slowly than other introns, independent of their stronger splice site strength.

Supplementary Figure 3. Identifying sites of recursive splicing. **(A)** Schematic indicating two computational approaches used to detect recursive sites: junction split and spanning reads (*top*) and automatic detection of sawtooth patterns (*bottom*). **(B)** Number of recursive sites identified by one of multiple identification pipelines, with the majority of recursive sites identified by both junction reads and sawtooth scores, as well as present in the Duff *et al.* dataset. **(C)** The gene expression levels of genes with recursive introns (TPM, *y-axis*) relative to the junction spanning read support for each recursive intron (read count, *x-axis*), showing the varying power to identify recursive sites with the sawtooth recursive method (*orange*), junction-spanning reads alone (*blue*), or both methods (*black*). **(D)** The probability derived from the sawtooth MCMC model of a site being a recursive site for the final set of recursive sites (*light orange*), all sites with minimal support from any method (*dark orange*), and random sites placed down in the same introns (*grey*). **(E)** The sawtooth score (see Methods) for the final set of recursive sites (*light orange*), all sites with minimal support from any method (*dark orange*), and random sites place down in the same introns (*grey*). **(F)** The cumulative distribution of distances between the recursive site identified with the sawtooth recursive method and the best matching recur-

sive motif (*orange*) and random sites placed down in the same introns (*grey*) are significantly different.

Supplementary Figure 4. Properties of recursively spliced introns. **(A)** Sequence logo for all intronic AG—GT sites (*top*), medium-confidence recursive sites (*middle*) and high-confidence recursive sites (*bottom*). **(B)** Conservation of sequences around all detected recursive sites, with average phastCons scores for medium-confidence recursive sites (*yellow*), high-confidence recursive sites (*gold*), and random AG—GT sites in introns increasingly larger than 1kb (*grey*). **(C)** Full intron length distributions for introns (*y-axis*) with varying numbers of recursive sites (*x-axis*).

Supplementary Figure 5. Rates of recursive splicing. **(A)** Splicing half-lives (*y-axis*) for recursive segments with varying positions across the intron (*x-axis*), where on average, all segments in an intron tend to be spliced out at similar rates. **(B)** The number of splice junction reads (*y-axis*) spanning a $5'$ splice site and recursive site (*blue*), two recursive sites (*gold*), and a recursive site and $3'$ splice site (*yellow*) across the time-points (*x-axis*). **(C)** Distribution of lengths of recursive segments (nucleotides, *x-axis*) for medium-confidence recursive segments (*yellow*) and high-confidence recursive segments (*gold*). **(D)** The distribution of splicing half-lives (*y-axis*) for the longest recursive segments in introns (*gold*) relative to non-recursive introns chosen to match the length of the recursive segments (*grey*). **(E)** The distribution of mean life-times (*y-axis*) for recursively spliced introns (estimated by the maximum of exponentials from constituent recursive segment splicing rates, *gold*) relative to non-reucursive introns chosen to match the length of the recursive introns (*grey*).

Supplementary Figure 6. Variance in splicing half-lives across introns in a gene. **(A)** The proportion of annotated introns detected for each gene (*x-axis*), with 76% of genes having suf-

ficient coverage in 100% of annotated introns. **(B)** The cumulative distribution of variance in splicing half-lives (coefficient of variation, *x-axis*) across introns within a gene (*dark blue*) and introns randomly sampled to match the distribution of lengths of introns within actual genes (*dark grey*). This trend is consistent when excluding the first intron of each gene (*light blue*) and doing a similar sampling strategy excluding the length of the first introns (*light grey*). **(C)** Splicing half-lives (*y-axis*) before (*dark blue*) and after (*light blue*) correcting for the non-linear correlation between splicing half-lives and intron length. Length-correction was done by subtracting a running median of local splicing half-lives, where medians were computed in 50 intron sliding bins. **(D)** The cumulative distribution of variance in length-corrected relative splicing half-lives (coefficient of variation, *x-axis*), within categories of observed and sampled introns as in (B).

Supplementary Figure 7. Correcting for the effects of intron length. **(A)** The distribution of splicing efficiency (median half-lives, *y-axis*) vs. intron length (mean nucleotides, *x-axis*) for introns in different positions across a gene. First introns are longer and more slowly spliced than non-first introns. **(B)** Correcting for length does not account for the slower splicing of first introns, where splicing half-lives (*y-axis*) for first introns are still slower than for non-first introns in both the measured half-lives (*dark blue*) and the length-corrected relative half-lives (*light blue*). **(C)** The percentage of enhancers within an intron (*y-axis*) for first introns (*blue*) and non-first introns (*grey*) binned by quintiles of intron length (*x-axis*). **(D)** The distribution of intron half-lives (*y-axis*) for introns containing an enhancer (*blue*) and without an enhancer (*grey*) binned by quintiles of intron length (*x-axis*).

Supplementary Figure 8. First-intron length and splicing efficiency. Running median (*red*) of local gene-specific median splicing half-lives across the distribution of first intron lengths, for raw splicing half-lives **(A)** and splicing half-lives corrected for intron length **(B)**.

Running median is computed in sliding bins of 50 genes. **(C)** Selected genes with four or five total introns, of which all of the non-first introns are 60-70 nt in length and have low variance across their splicing half-lives. Varying first intron lengths across these genes (nucleotides, *x-axis*) shows a correlation between first intron length and the median half-lives for these genes (*y-axis*).

Supplementary Figure 9. Coefficients from a multiple linear-regression with several parameters (*y-axis*), where the coefficient represents the % change in half-life concordant with a 1% change in each parameter. Bars indicate the standard error and the size of the mean dot indicates the $-log_{10}$ p-value for the significance of the individual parameter.

7 Legends for Supplementary Tables

Supplementary Table 1: Summary of introns analyzed . Summary statistics and information for all introns that were analyzed in this study.

- **Column 1: `intron`**. Coordinates of intron, with chr:start:end:strand for the upstream flanking exon and the chr:start:end:strand for the downstream flanking exon separated with an '@'.
- **Column 2: `gene`**. FlyBase gene symbol for parent gene.
- **Column 3: `TPM`**. Gene expression values calculated using kallisto.
- **Column 4: `intron_position`**. Position of intron relative to other introns in the transcript.
- **Column 5: `intron_len`**. Length of intron (nucleotides).

- **Column 6: `intron_type`**. Regulatory type of intron, where CI is constitutively spliced intron, RI is an annotated retained intron, and SEflanking is an intron that flanks a retained intron.

- **Columns 7-17: `psi_[timepoint]_[replicate]`**. MISO-derived Ψ values of intron across timepoints and replicates.

- **Columns 18-28: `ci_[timepoint]_[replicate]`**. MISO-derived confidence intervals around the Ψ values for each timepoint and replicate.

- **Column 19: `Tau`**. τ value which accounts for time to transcribe the region from the $3'$ splice site of the intron to the polyA site of the transcript.

- **Column 20: `t_inferred`**. Comma separated list of inferred average lifetimes of the intron across experimental timepoints, given the τ value.

- **Column 21: `coefficient`**. Regression coefficient from fit of first-order exponential model to log-transformed Ψ values (linear fit). In this form, the regression coefficient is reciprocal of the decay coefficient λ.

- **Column 22: `r_squared`**. R-squared value for goodness of fit from the first-order exponential model.

- **Column 23: `half_life`**. Half-life of intron computed from the decay coefficient.

- **Column 24: `halflife_error`**. Error of around the half-life estimate.

- **Column 25: `ss5_maxEnt`**. maxEnt-derived splice-site score for the $5'$ splice site of the intron.

- **Column 26: `ss3_maxEnt`**. maxEnt-derived splice-site score for the 3′ splice site of the intron.

- **Column 27: `contains_enhancer`**. Flag for whether the intron contains a transcriptional enhancer as defined by STARR-Seq.

Supplementary Table 2: Recursive sites. Summary statistics and information for all recursive sites that were analyzed in this study.

- **Column 1: `intron`**. Coordinates of intron containing recursive site, with chr:start-end:strand.

- **Column 2: `gene`**. FlyBase gene symbol for parent gene.

- **Column 3: `TPM`**. Gene expression values calculated using kallisto.

- **Column 4: `completed_splicing _junction _reads`**. Number of junction reads supporting completed splicing across the entire intron.

- **Column 5: `recursive _site`**. Coordinate for the recursive site.

- **Column 6: `method`**. Method used for identification of the recursive site, where 'junction' indicates site identified by either 'RachetJunction' or 'RachetPair', 'sawtooth' indicates site identified by 'RachetScan', and 'both' indicates site identified by both methods.

- **Column 7: `in_duff`**. Flag indicating the recursive site was identified in the Duff *et al.* study.

- **Column 8: `high_confidence`**. Flag indicating the recursive site was identified as a high-confidence site (1) or a medium-confidence site (0).

- **Column 9: `junction_reads`**. Comma-separated list of number of junction reads supporting the recursive site in each timepoint (combined across replicates) [5m, 10m, 20m, total].

- **Column 10: `spanning_read _pairs`**. Comma-separated list of number of spanning read-pairs supporting the recursive site in each timepoint (combined across replicates) [5m, 10m, 20m, total].

- **Column 11: `sawtooth _score`**. Sawtooth score for the recursive site, as defined in the Supplementary Methods.

- **Column 12: `mcmc_probability`**. Probability of this site being a recursive site, as derived from the MCMC sampling procedure described in the Supplementary Methods.

- **Column 13: `recursive_index`**. Recursive index for the recursive site, as defined in the Supplementary methods.

- **Column 14: `motif`**. Sequence found around the recursive site.

- **Column 15: `motif_score`**. Motif score for the recursive site.

- **Column 16: `downstream_reads`**. Number of splice junction reads originating from the downstream end of the exon.

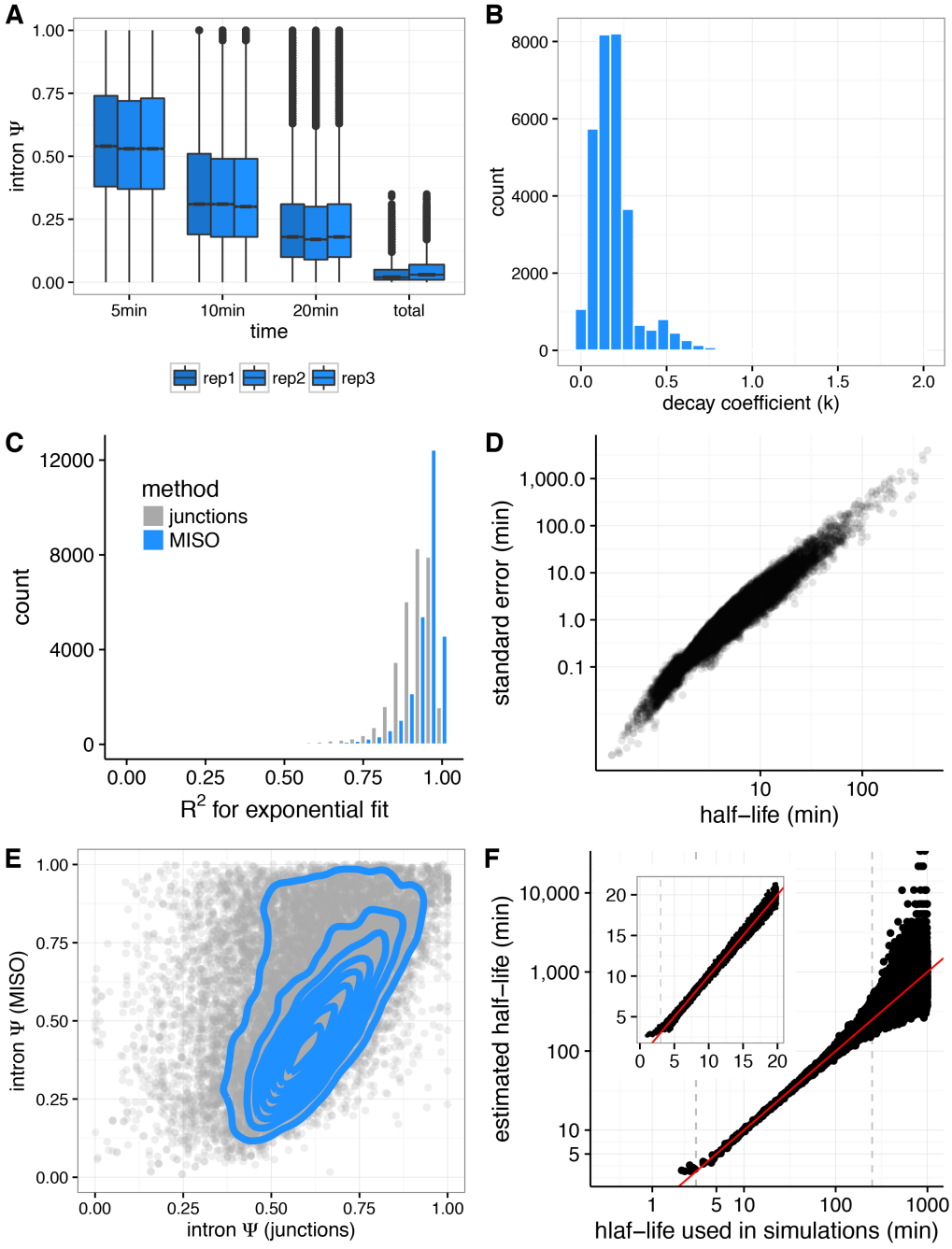
- **Column 17: `intron_body_reads`**. Number of reads in the body of the intron.

Supplementary Table 3: Gene Ontology analyses for recursively spliced introns. Summary output from DAVID Gene Ontology Analyses for significantly enriched biological process gene ontology categories.

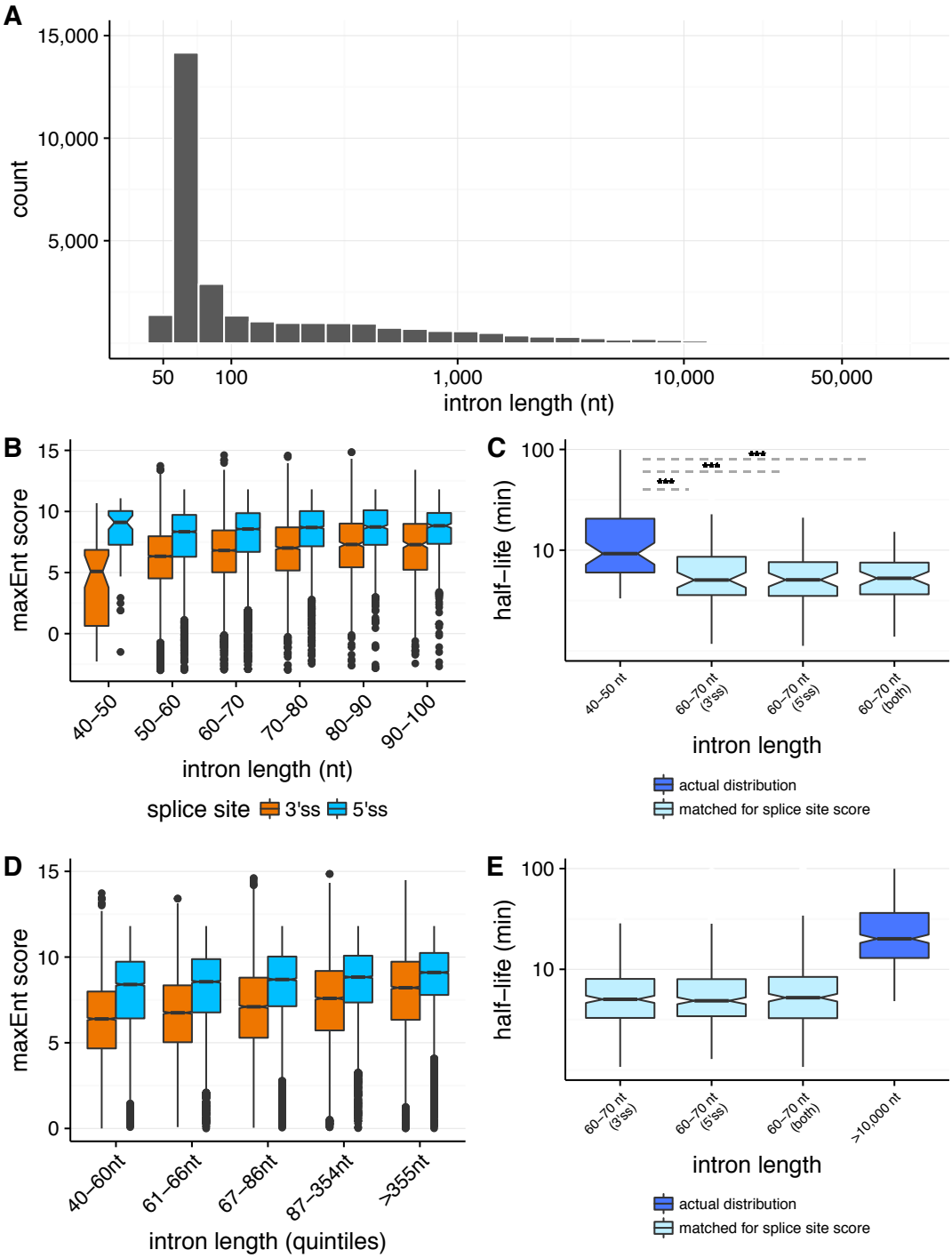
Supplementary Table 4: Gene Ontology analyses for genes with shorter median half-lives. Summary output from DAVID Gene Ontology Analyses for significantly enriched biological process gene ontology categories.

Supplementary Table 5: Gene Ontology analyses for genes with longer median half-lives. Summary output from DAVID Gene Ontology Analyses for significantly enriched biological process gene ontology categories.

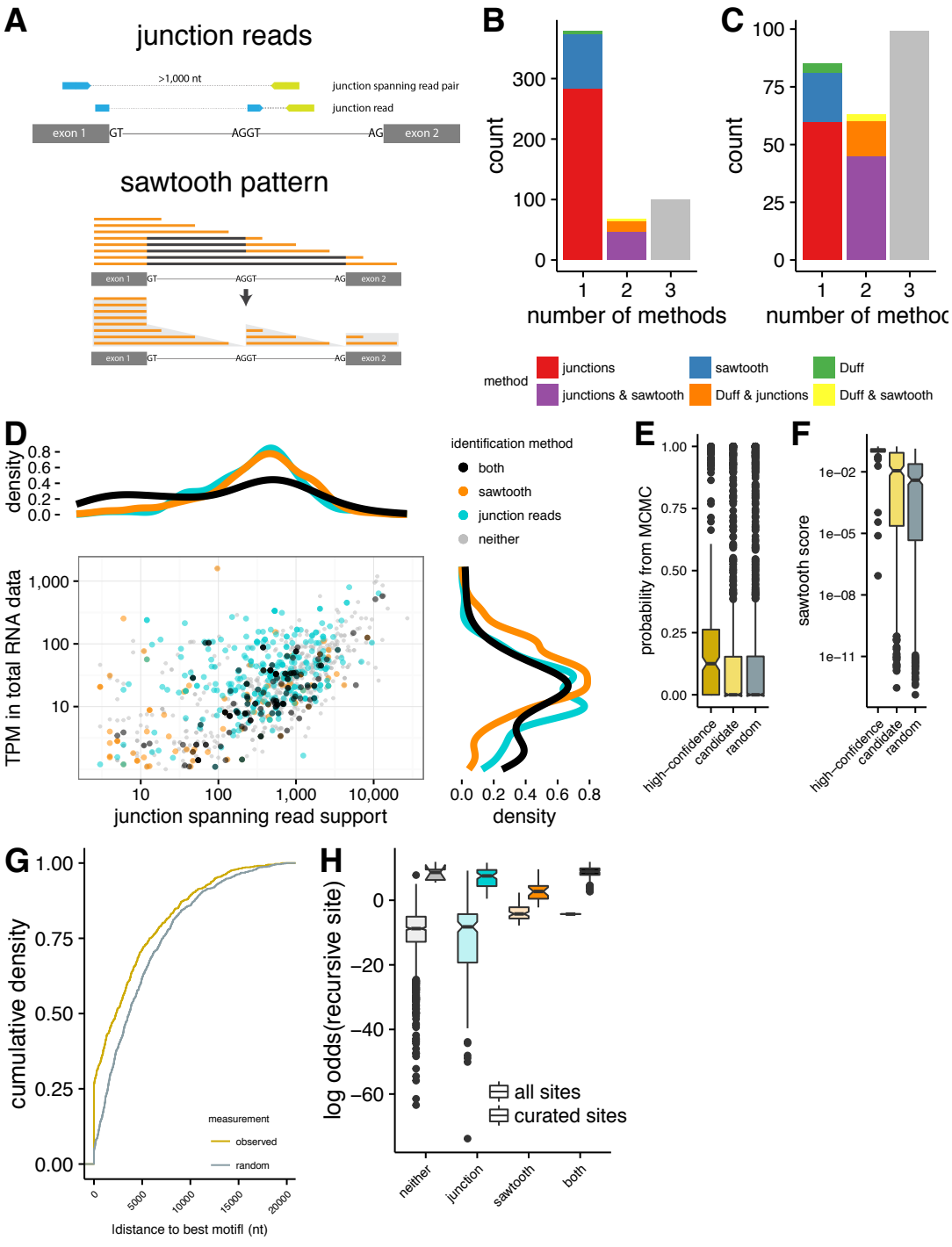
8 Supplementary Figures



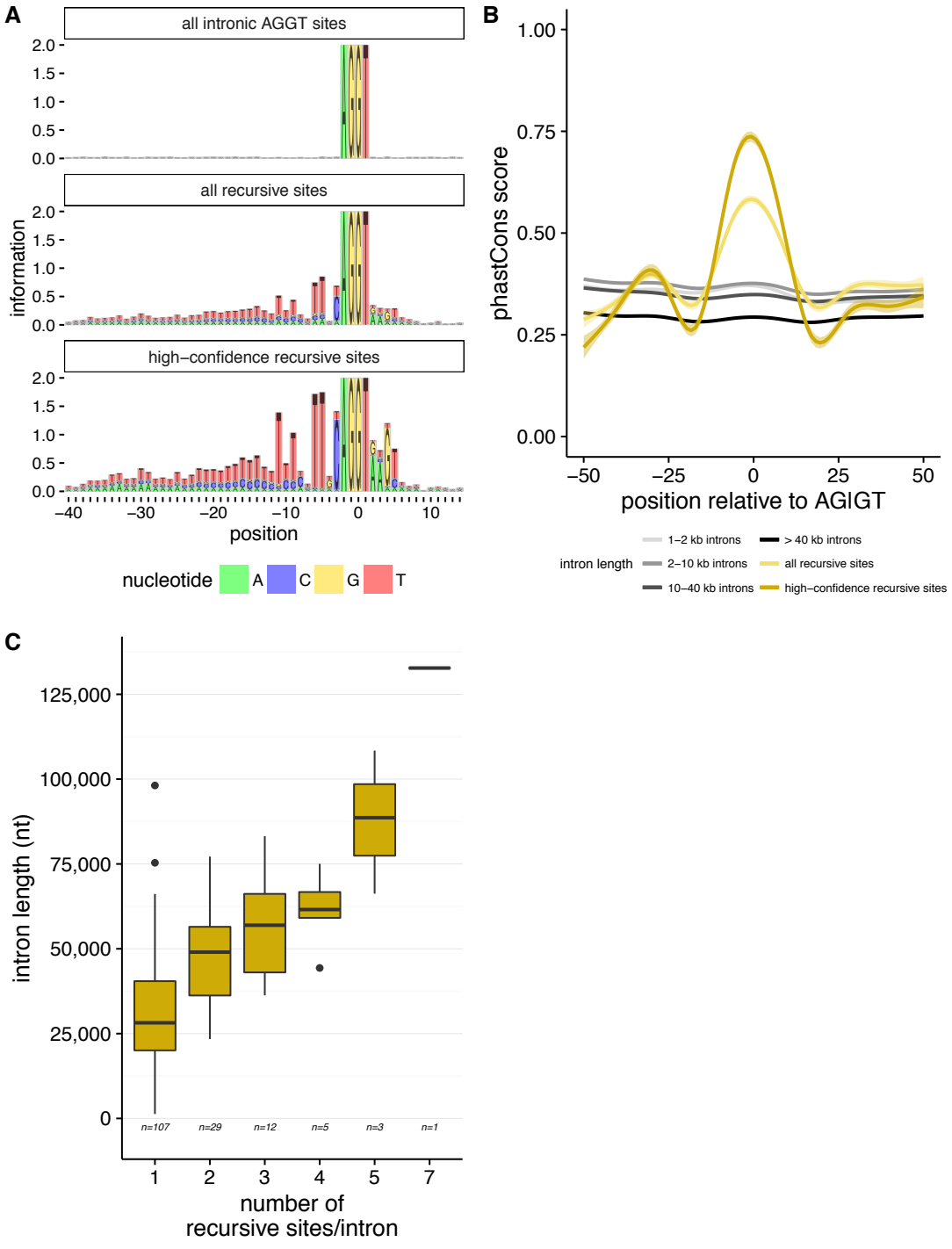
Supplementary Figure 1. Calculating rates of splicing in Drosophila.



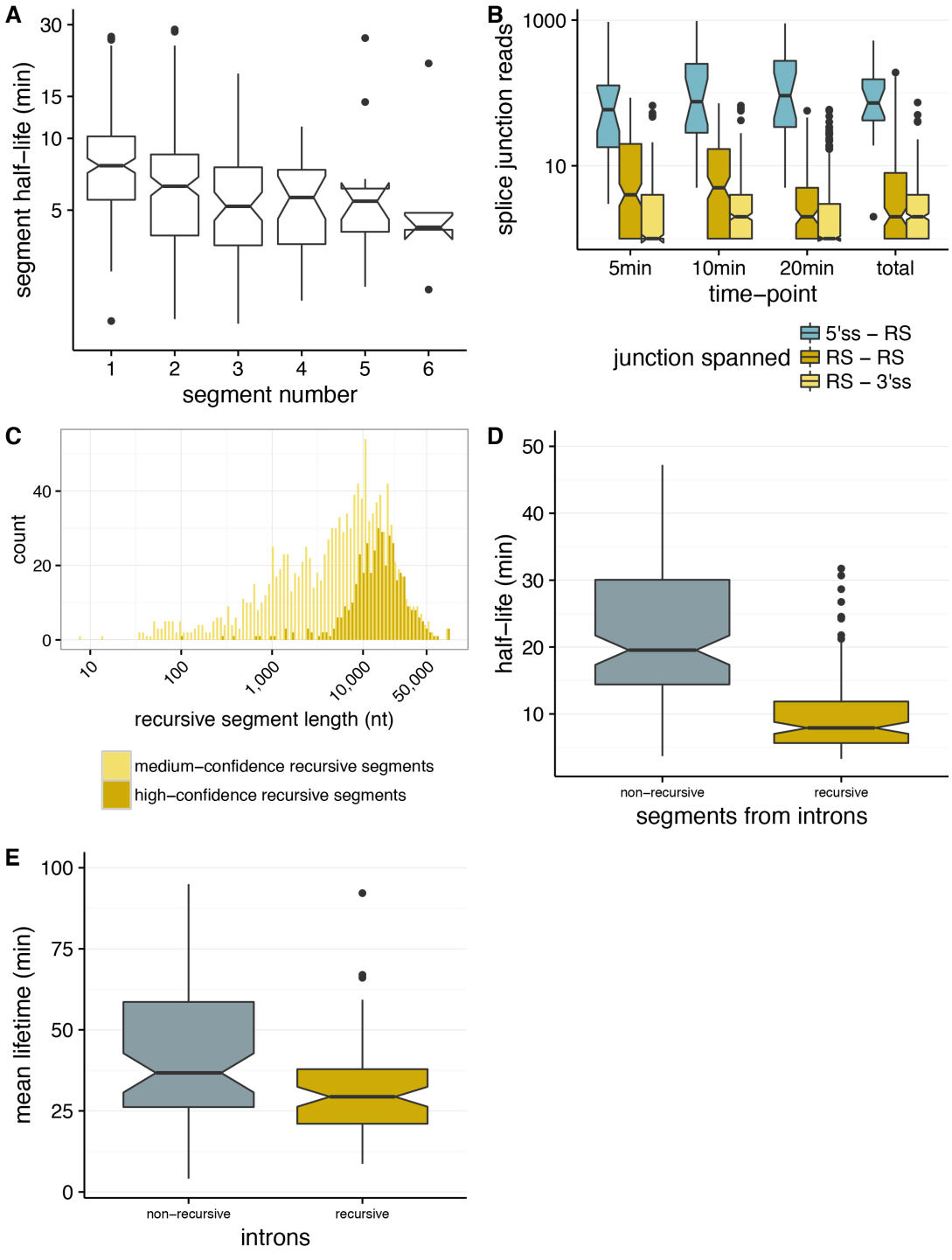
Supplementary Figure 2. Properties of splicing efficiency across varying intron lengths.



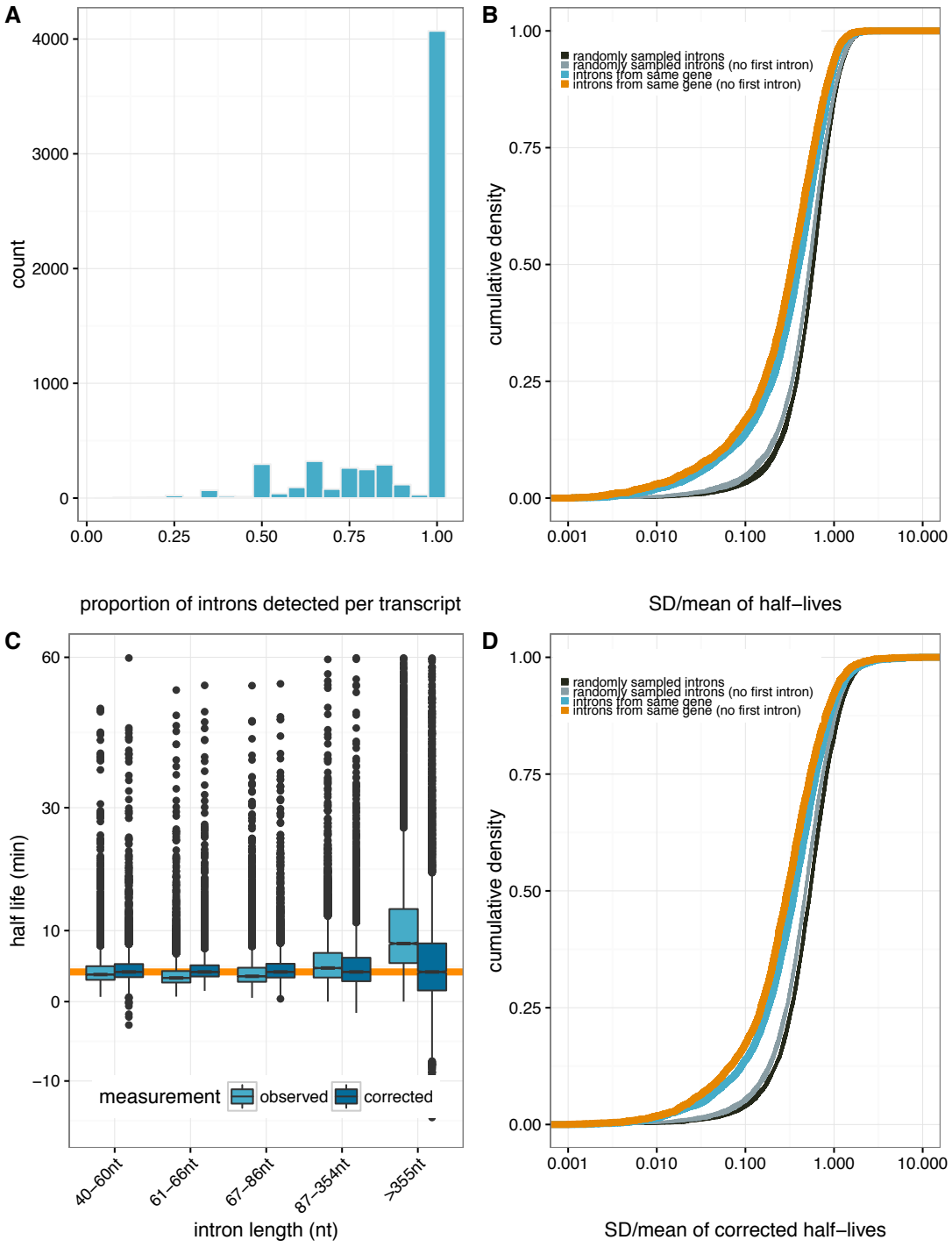
Supplementary Figure 3. Identifying sites of recursive splicing.



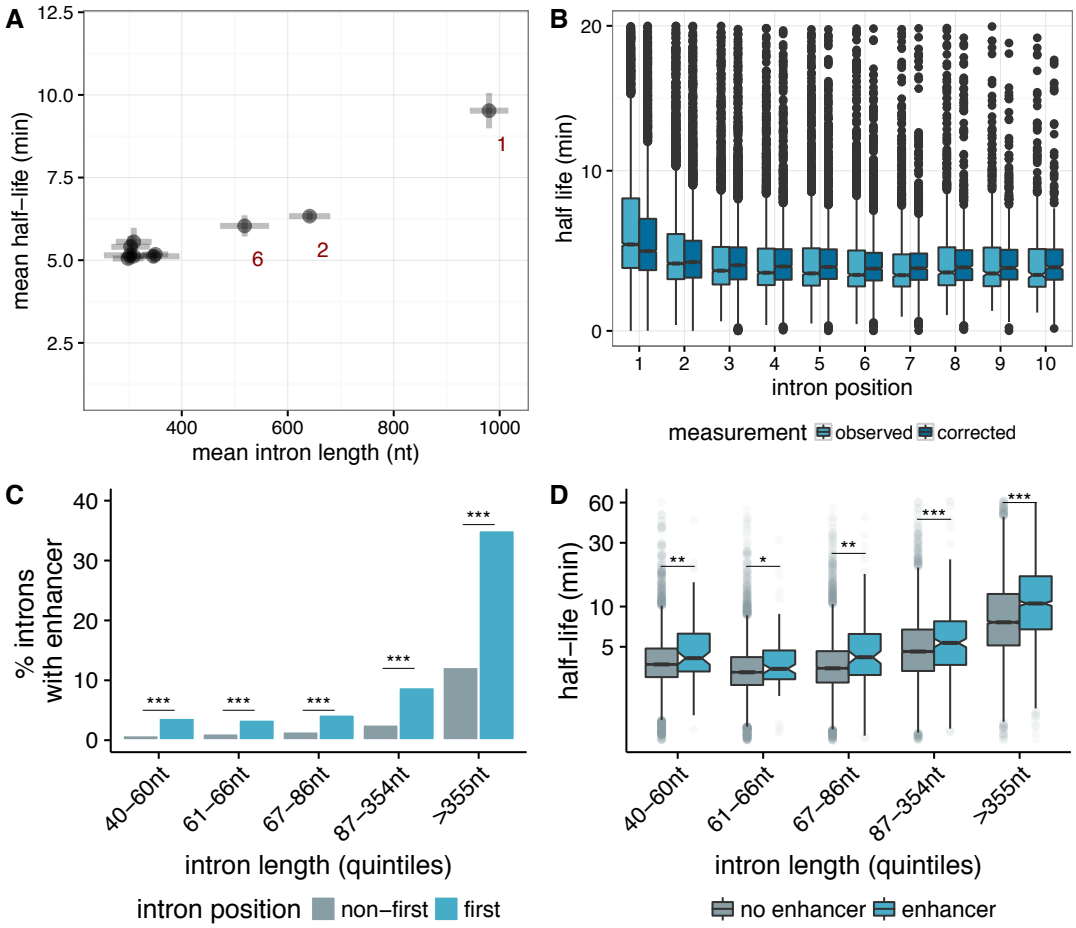
Supplementary Figure 4. Properties of recursively spliced introns.



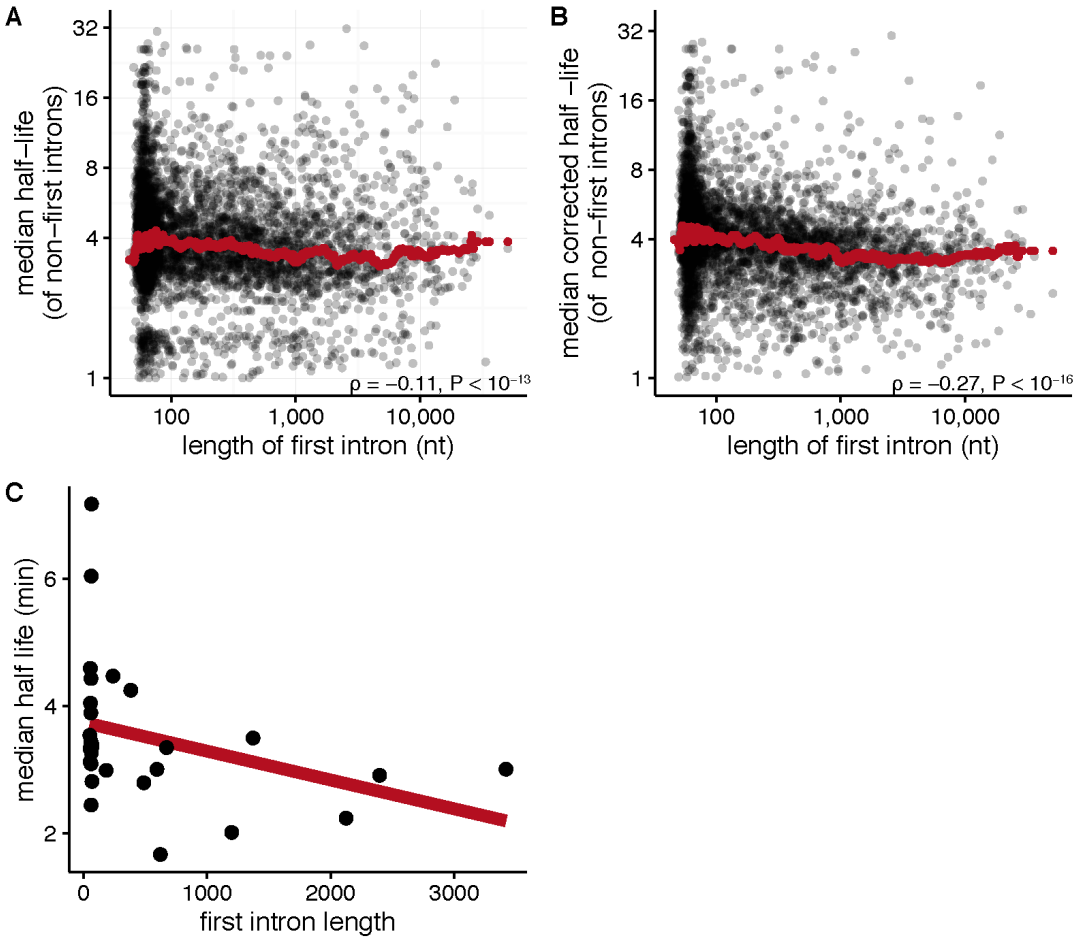
Supplementary Figure 5. Rates of recursive splicing.



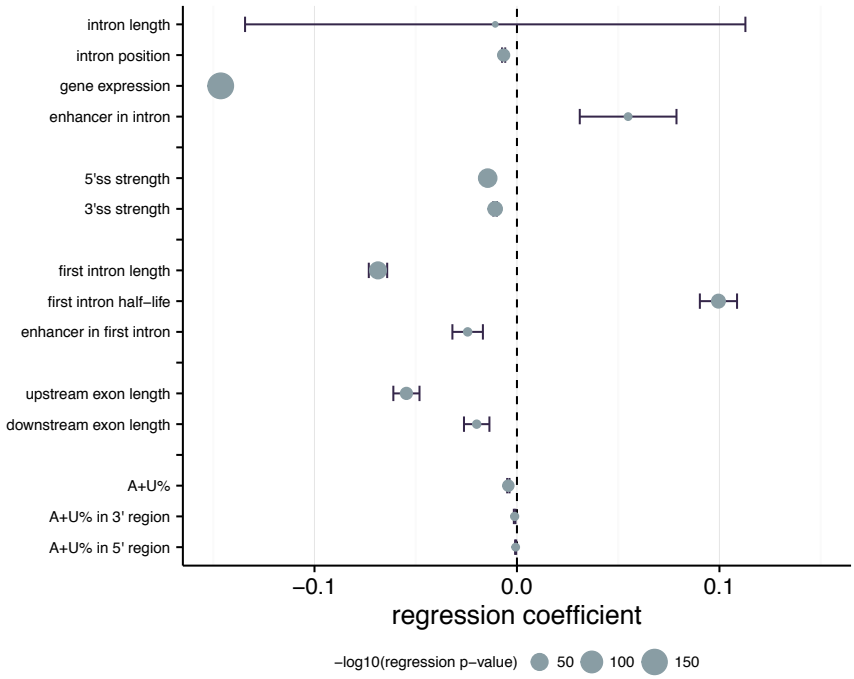
Supplementary Figure 6. Variance in splicing half-lives across introns in a gene.



Supplementary Figure 7. Correcting for the effects of intron length.



Supplementary Figure 8. First-intron length and splicing efficiency.



Supplementary Figure 9.

References

[1] Grömping, U. (2006). Relative Importance for Linear Regression in R: The Package relaimpo. Journal of Statistical Software 17, Issue 1.

[2] Zare-Mirakabad F, Ahrabian H, Sadeghi M, Nowzari-Dalini A, Goliaei B. (2009) New scoring schema for finding motifs in DNA Sequences. *BMC Bioinformatics* 10(1):93.

[3] Kim D, Langmead B, Salzberg SL. (2015) HISAT: a fast spliced aligner with low memory requirements. *Nature Methods* 12:357-360.

[4] Heger A, Jacobs K. (2016) pysam: htslib interface for python. *https://github.com/pysam-developers/pysam*

[5] Guo Y, Mahony S, Gifford DK. (2012) High Resolution Genome Wide Binding Event Finding and Motif Discovery Reveals Transcription Factor Spatial Binding Constraints. *PLOS Comp. Bio.* 8(8):e1002638.

[6] Duff MO, Olson S, Wei X, Garrett SC, Osman A, Bolisetty M, Plocik A, Celniker SE, Graveley BR. (2015) Genome-wide identification of zero nucleotide recursive splicing in Drosophila. *Nature* 521(7552):376-379.

[7] Smit AFA, Hubley R, Green P. (2013-2015) RepeatMasker Open-4.0. *http://www.repeatmasker.org*

[8] Jones E, Oliphant E, Peterson P, et al. (2001-) SciPy: Open Source Scientific Tools for Python. *http://www.scipy.org/*

[9] Dobin A, Davis CA, Schlesinger F, Drendow J, Zaleski C, Jha S, Batut P, Chaisson M, Gingeras TR. (2013) STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*