Additional Figures

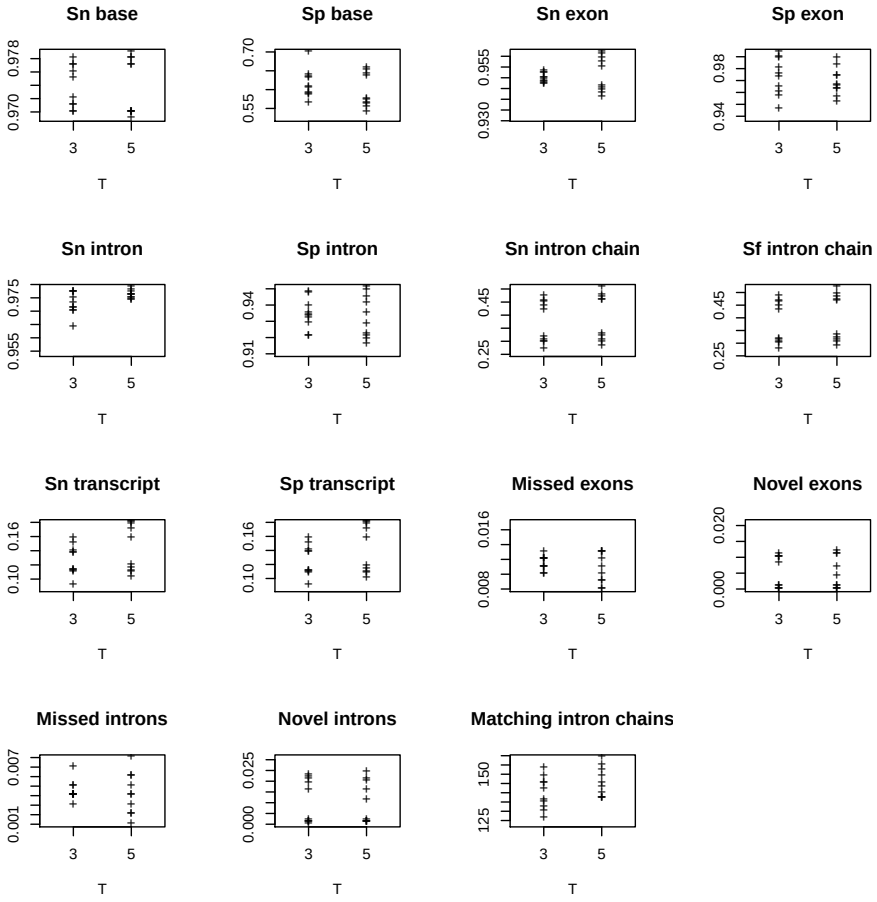

Figure 1: **Assessing the convergence of `altra` in transcript reconstruction**: the different panels show plots of different performance measures on 5 different runs of `altra` as a function of T, the number of alternative transcripts per gene in τ_{sim}. All metrics, except the number of matching intron chains - last plot - are constrained between 0 and 1 (e.g., the highest sensitivity Sn is 1, the lowest sensitivity is 0).

Representation of reads

In `altra`, reads are represented by a set of sensible coordinates. Since at each transcript proposal the compatibility between the proposed transcripts and all reads must be computed, a sensible representation is crucial.

As depicted in Figure 1a of the main paper, `altra` partitions each locus in a set of adjacent intervals using sensible coordinates. These coordinates are collected as follows.

First, the set of positive 3' and 5' splice site (i.e. splice sites on the forward strand), $S_+^{(3)}$ and $S_+^{(5)}$, respectively, are collected by any or all of the following procedures:

- Identifying junctions (and therefore 3' and 5' splice sites) on the positive (and negative) strand with a splice–aware mapping method; these *de novo* junctions are directional, their directionality being determined by the intronic consensus sequence;
- Collecting annotated junctions, annotated starts and annotated ends;
- Inferring splice sites (ends and starts) by running FLLat (Nowak et al., 2011), a segmentation method, on the set of N coverage patterns in the locus. We use FLLat because it can handle multiple samples simultaneously. In this case the splice site position is affected by an error and the directionality of the splice site is unknown so 3' (5') splice sites are added to both $S_+^{(3)}$ and $S_-^{(3)}$ ($S_+^{(5)}$ and $S_-^{(5)}$). This step can help identifying unannotated starts, ends and junctions.

Negative splice sites are collected in the same way. Note that, in our model, a transcript end is equivalent to a 5' splice site and a transcript start is equivalent to a 3' splice site.

After collecting coordinates, `altra` partitions reads into categories: positive, negative, unsigned; and then into classes: if reads cover different set of intervals, they fall into different classes.

A *positive (negative) read* is a spliced read that contains a junction with a positive (negative) sense splice signal and an *unsigned read* is a read that is not spliced and that could have been emitted both by a transcript on the forward strand and by a transcript on the reverse strand. This partitioning can speed up computation because only transcripts on the same strand (for example the forward strand) are modified at each proposal. Therefore compatibility can be recomputed between the proposed transcripts and reads in the appropriate category (for example, positive and unsigned reads if the proposal involves positive transcripts).

A *read class* is a vector of coordinates $(\boldsymbol{c}5, \boldsymbol{c}3)$ where $\boldsymbol{c}5$ is the largest coordinate upstream of the 5' end of the read and $\boldsymbol{c}3$ is the smallest coordinate downstream of the 3' end of the read (see Figure 1 of the main paper). Note that, in the general case, $\boldsymbol{c}5$ and $\boldsymbol{c}3$ are vectors because a spliced read can have more than one 5' and 3' ends. `Altra` only stores classes found in the data. For each class, `altra` stores the number of reads of that class in each of the N samples.

Representation of transcripts

At each proposal to update the gene model `altra` would need to recompute the compatibility of all reads with the proposed transcripts. To speed up computation, `altra` represents transcripts as C-dimensional *bitsets* (vectors of 1-s and 0-s: if transcript t covers interval c the bitset's c-th element is 1, otherwise it is 0) where C is the number of intervals, and keeps track of which intervals of each transcript are modified. By doing so, `altra` can

recompute the compatibility of only those reads that overlap regions where transcripts have been modified.

Construction of flexons

At each locus, `altra` constructs *positive (negative) flexons* using $S_+^{(3)}$ and $S_+^{(5)}$ ($S_-^{(3)}$ and $S_-^{(5)}$), the sets of positive (negative) 3' and 5' splice sites, respectively, given a minimum and a maximum exon length. Here we describe the procedure to construct positive flexons. The procedure for negative flexons is equivalent and independent.

First all pairs $(s3, s5)$ are identified, where $s3 \in S_+^{(3)}$ is a positive 3' splice site, $s5 \in S_+^{(5)}$ is a positive 5' splice site, and:

- $s3$ is upstream of $s5$;
- No other 5' splice sites in $S_+^{(5)}$ or 3' splice site in $S_+^{(3)}$ fall in between them;
- There is at least a pair of splice sites $\bar{s3} \in S_+^{(3)}$ and $\bar{s5} \in S_+^{(5)}$ such that $\bar{s3}$ is upstream or equal to $s3$ and $\bar{s5}$ is downstream or equal to $s5$ and such that the distance between $\bar{s3}$ and $\bar{s5}$ is larger than the minimum exon length;
- The distance between $s3$ and $s5$ is less than the maximum exon length.

Then, a flexon is defined as the set of 3' splice sites upstream of $s5$ and the set of 5' splice sites downstream of $s3$ that can generate exons shorter than the maximum exon length (see flexon 1 in Figure 1b of the main paper).

A transcript is generated by including/excluding flexons. *A priori* flexon f is included in any transcript with the same probability p_f with p_f uniform on $[0, 1]$:

$$\gamma_{ft}|p_f \sim Bern(p_f) \qquad p_f \sim Beta(1,1) \qquad \text{for } f \in \{1,\dots,F\}, t \in \{1,\dots,T\},$$

where γ_{ft} is an indicator of value 1 if flexon f is included in transcript t and of value 0 otherwise. If flexon f is included, its 3' splice site is chosen according to $\boldsymbol{p}_f^{(3)}$ and its 5' splice site is chosen according to $\boldsymbol{p}_f^{(5)}$. If $\boldsymbol{p}^{(3)} = (p_{fs}^{(3)})_{f\in\{1,\dots,F\}\ s\in S_f^{(3)}}$ where s is a 3' splice site in $S_f^{(3)}$ and $p_{fs}^{(3)}$ is the probability of using splice site s at flexon f then:

$$p(s_{ft}^{(3)} = s) = p_{fs}^{(3)} \qquad \boldsymbol{p}_f^{(3)} \sim Dir(\boldsymbol{\alpha}) \qquad \text{for } f \in \{1,\dots,F\}, t \in \{1,\dots,T\},$$

with $\boldsymbol{\alpha} = \mathbf{1}$. The same assumptions are made on $\boldsymbol{p}^{(5)} = (p_{fr}^{(5)})_{f\in\{1,\dots,F\}\ r\in S_f^{(5)}}$.

With these assumptions alternative transcripts are encouraged to share a similar structure.

The Metropolis Hastings algorithm

`Altra` uses a Metropolis Hastings (MH) algorithm to explore the posterior distribution $p(\tau, \boldsymbol{\theta}, \delta|\boldsymbol{x})$ given the data $\boldsymbol{x}$ and priors on $\tau, \boldsymbol{\theta}$, and δ.

`Altra` initializes the MH algorithm by sampling $\boldsymbol{\theta}$ and δ from the prior and by randomly sampling T transcripts. If an annotation is available, `altra` chooses a random set of T annotated transcripts as a sensible start.

After the initialization, `altra` updates θ_t^i with an independent random walk Metropolis algorithm (RWM) given $\boldsymbol{x}$, τ, δ and given $\bar{\theta}_t$ and $\sigma^2_{\theta,t}$ which are drawn from

$$\begin{aligned}\bar{\theta}_t|\theta_t^1,\ldots,\theta_t^N,\sigma^2_{\theta,t},c &\sim \mathcal{N}\left(\tfrac{cN}{1+cN}m_t, \tfrac{c}{1+cN}\sigma^2_{\theta,t}\right)\\ \sigma^2_{\theta,t}|\theta_t^1,\ldots,\theta_t^N,\alpha,\beta &\sim IG(\alpha+a,\beta+b_t),\end{aligned}$$

where $m_t = \frac{\sum_{i=1}^N \theta_t^i}{N}$, $a = \frac{N}{2}$ and $b_t = \frac{1}{2}\left[(N-1)s_t^2 + \frac{N}{1+cN}m_t\right]$ with $s_t^2 = \frac{\sum_{i=1}^N(\theta_t^i - m_t)^2}{N-1}$ and updates δ with a RWM given $\boldsymbol{x}$, τ and δ.

Then, `altra` updates the gene model τ. To update τ, `altra` selects a random transcript t_0 in τ and a random flexon f_0 (uniformly). Then for transcript t_0 at flexon f_0, it proposes one of the following moves:

1. *To change a splice site/to include the flexon/to exclude a flexon* by drawing p_{f_0} from the posterior $p_{f_0}|n_{f_0} \sim Beta(1+n_{f_0}, 1+T-n_{f_0})$ and $\gamma'_{f_0t_0}$ from the prior $\gamma'_{f_0t_0}|p_{f_0} \sim Bern(p_{f_0})$, where n_{f_0} is the number of transcripts using flexon f_0. Then:

   - If $\gamma_{f_0t_0} = \gamma'_{f_0t_0} = 1$: with probability $\frac{|S^{(3)}_{f_0}|}{|S^{(3)}_{f_0}|+|S^{(5)}_{f_0}|}$ it proposes to *change* the 3' *splice site* $s^{(3)}_{f_0t_0}$ (see below), otherwise it proposest to change the 5' splice site $s^{(5)}_{f_0t_0}$ (see below);
   - If $\gamma_{f_0t_0} = 0$ and $\gamma'_{f_0t_0} = 1$: it proposes to *include the flexon* and chooses a 3' and a 5' splice site (see below);
   - If $\gamma_{f_0t_0} = 1$ and $\gamma'_{f_0t_0} = 0$ it proposes to *exclude the flexon*.

2. *To swap* the configuration of flexon f_0 at transcript t_0 with the configuration of f_0 at a random transcript in τ;

3. *To recombine* transcript t_0 with a random transcript in τ.

We choose the ratio of these three moves to be 0.8:0.15:0.05.

When a change of splice site is proposed, for example of a 3' splice site s, $\boldsymbol{p}^{(3)}_{f_0}$ is drawn from the posterior $\boldsymbol{p}^{(3)}_{f_0}|\boldsymbol{n}_{f_0} \sim Dir(\boldsymbol{\alpha}_{f_0} + \boldsymbol{m}_{f_0})$ and $s^{(3)}_{f_0t_0}$ from the prior $p(s^{(3)}_{f_0t_0} = s') = p^{(3)}_{f_0s'}$, where m_{f_0s} is the number of transcripts that use 3' splice site s at flexon f_0.

A move is always rejected if intron length is greater than maximum intron length and/or exon length is not within minimum and maximum exon length.

Derivation of the log-likelihood

We derived the log-likelihood as follows. From equation (1) in the main paper and the independence assumption across reads and samples, it follows that we can write the log-likelihood based on the entire data $\boldsymbol{x}$ as:

$$l(\tau, \boldsymbol{\lambda}, \epsilon; \boldsymbol{x}) = \sum_{i=1}^{N} \sum_{r \in \rho} x_r^i \log \left[C^i \left(\sum_{t=1}^{T} z_{rt} \lambda_t^i + \epsilon \right) \right] - \sum_{i=1}^{N} C^i \sum_{r \in \rho} \left[\left(\sum_{t=1}^{T} z_{rt} \lambda_t^i \right) + \epsilon \right] - \sum_{i=1}^{N} \sum_{r \in \rho} \log(x_r^i!)$$

where all variables are defined in the main paper.

We simplify the first term in this sum by grouping together read types that are compatible with the same subset of transcripts, i.e., that share the same "compatibility vector" $z_{r\cdot}$. This generates a partition of ρ, the set of read types, $\rho_{\boldsymbol{z}} = \{r : \boldsymbol{z}_r = \boldsymbol{z}\}$ where $\boldsymbol{z}$ is the compatibility vector and $\boldsymbol{z} \in \mathcal{Z}$ is an element of $\mathcal{Z}$, the set of all possible compatibility vectors (of cardinality 2^T). The overall log-likelihood based on the entire data $\boldsymbol{x}$ simplifies to:

$$l(\tau, \boldsymbol{\lambda}, \epsilon; \boldsymbol{x}) = \sum_{i=1}^{N} \sum_{\boldsymbol{z} \in \mathcal{Z}} \sum_{r : \boldsymbol{z}_r = \boldsymbol{z}} x_r^i \ \log \left[C^i \left(\sum_{t=1}^{T} z_t \lambda_t^i + \epsilon \right) \right] - \sum_{i=1}^{N} C^i \sum_{t=1}^{T} \tilde{L}_t \lambda_t^i - \epsilon \tilde{L} \sum_{i=1}^{N} C^i + const$$

where we defined

$$const = -\sum_{i} \sum_{r \in \rho} log(x_r^i!),$$

a constant that does not depend on τ, $\boldsymbol{\lambda}$, or ϵ,

$$\tilde{L}_t = \sum_{r \in \rho} z_{rt} = L_t + 1 - RL - \mathcal{O}$$

as *the effective length of transcript* t, i.e., the number of possible read types that could be generated by transcript t , and

$$\tilde{L} = \sum_{r \in \rho} 1 = L + 1 - RL + \mathcal{J}$$

as *the effective length of the locus* or the cardinality of ρ, i.e., the number of possible different read types that could be generated in the locus, where

- L_t is the number of basis in transcript t,

- L is the number of bases in the locus,
- RL is the read length,
- M is the overhang constraint,
- term $1 - RL$ accounts for the fact that no reads of length RL can be generated that start in the last $RL - 1$ bases of the transcript/locus,
- $\mathcal{O}$ is a term that accounts for overhang constraints and that - in the simplest case when reads are not long enough to cover an entire exon - is equal to $J_t * 2M$, and J_t is the number of different junction read types that could be generated by transcript t,
- $\mathcal{J}$ is a term that accounts for junction reads that could be generated in the locus which, in the simplest case when junction reads only have one splice site, we approximate with $J * (RL - 2M + 1)$ with J the number of different junction read types, that we approximate by the number of different junction read types observed in the data.

References

Nowak, G., Hastie, T., Pollack, J. R., and Tibshirani, R. (2011). "A fused lasso latent feature model for analyzing multi-sample aCGH data." *Biostatistics Oxford England*, 12, 4, 776–791.