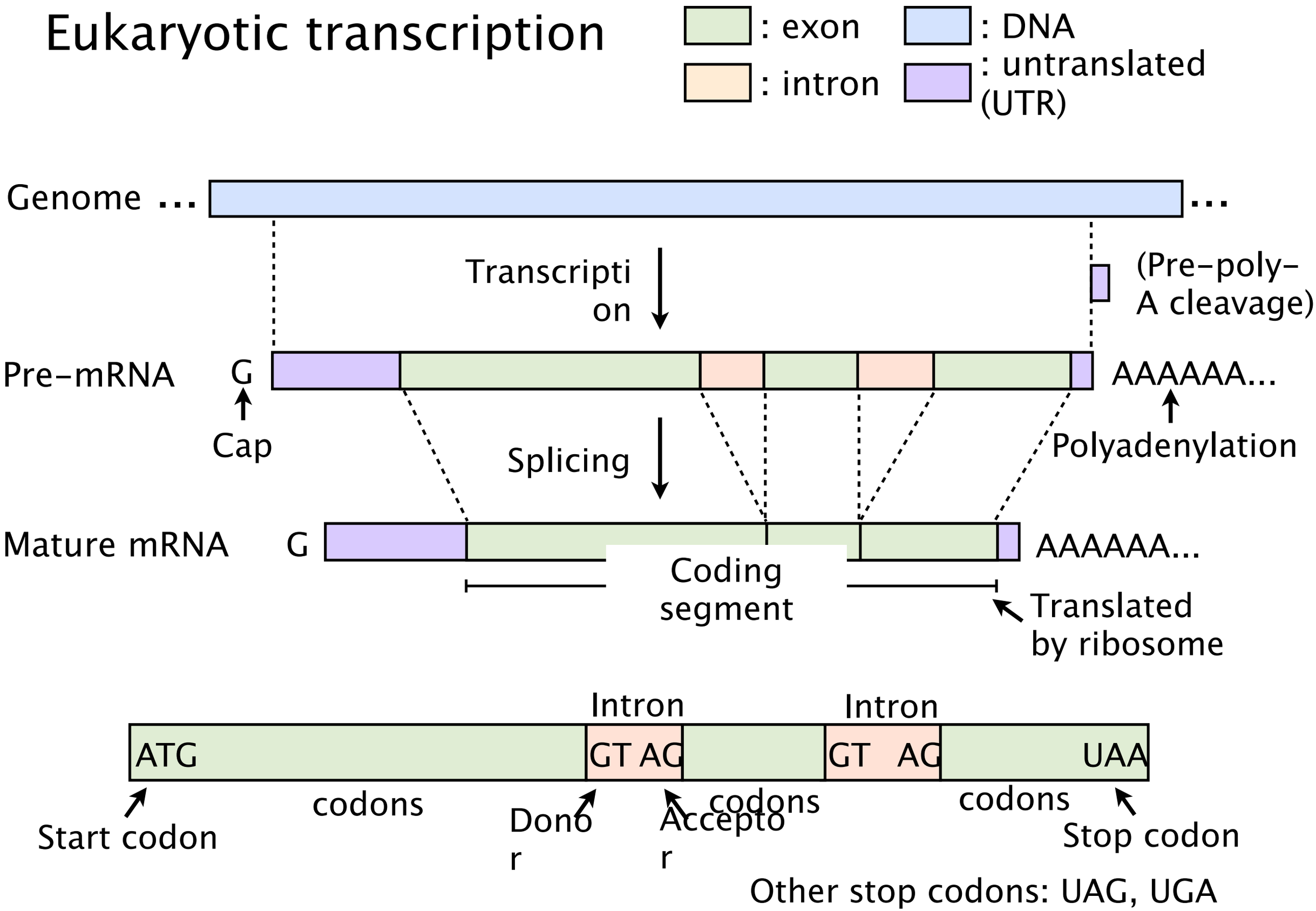
Spliced alignment in Rail-RNA

Ben Langmead et al

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Eukaryotic transcription



Eukaryotic genes

Splicing signals, e.g. donors & acceptors, are more suggestions than rules

There are other signals besides donors and acceptors

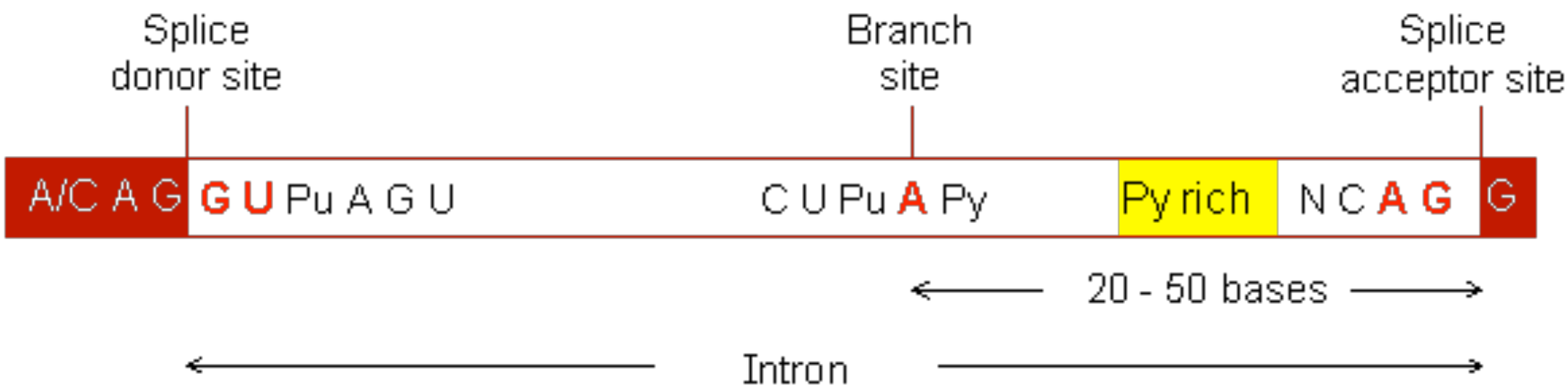
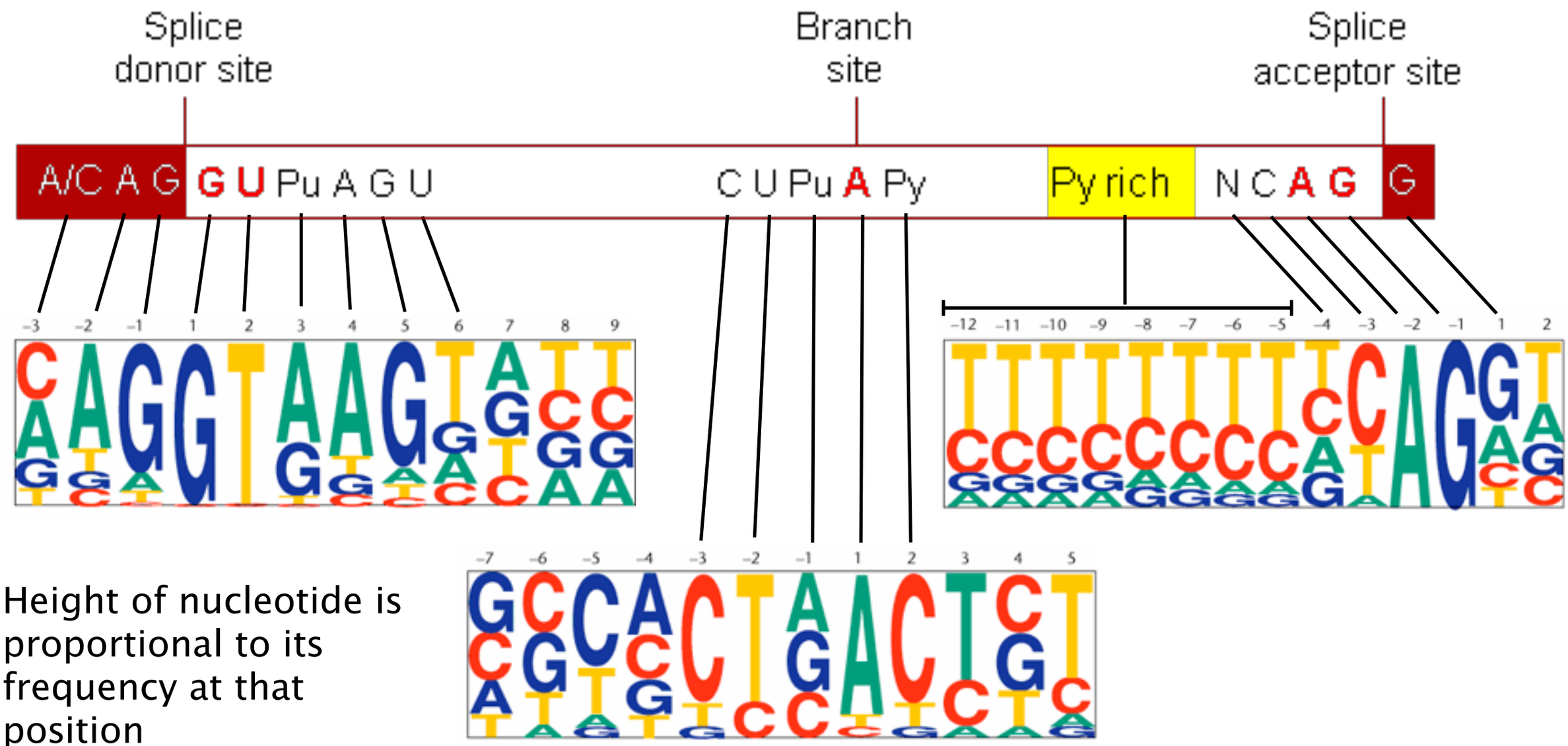


Image: http://www.web-books.com/MoBio/Free/Ch5A4.htm

Eukaryotic genes

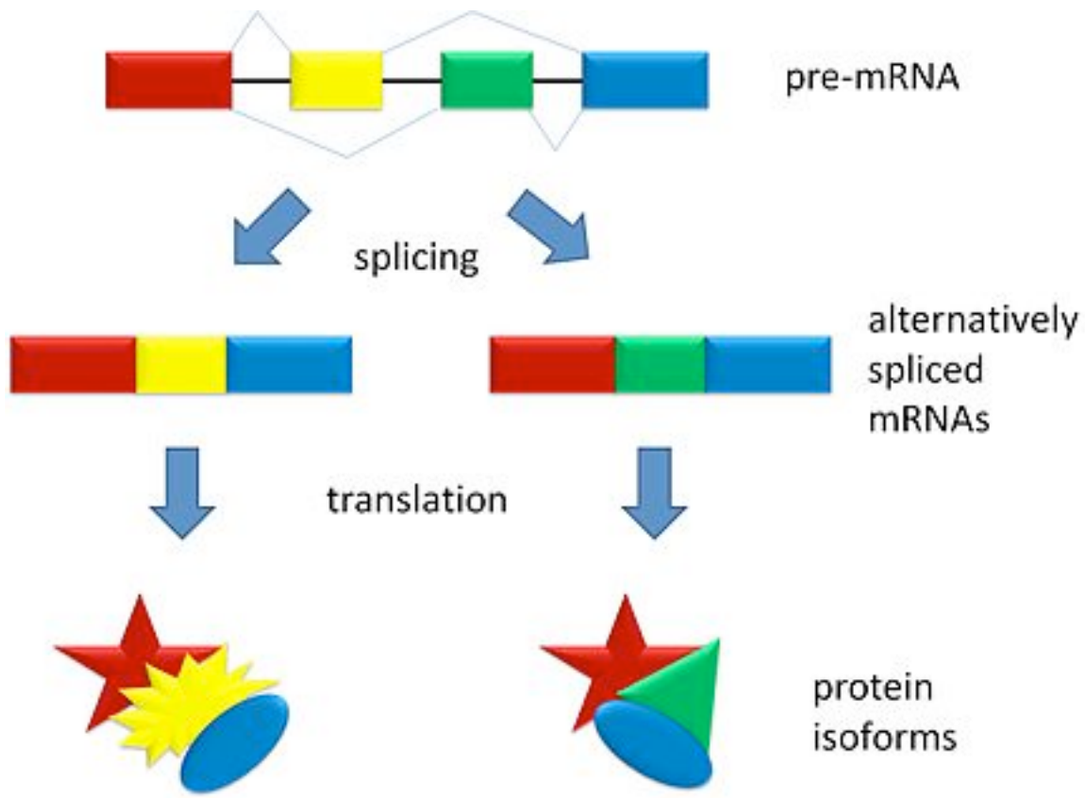


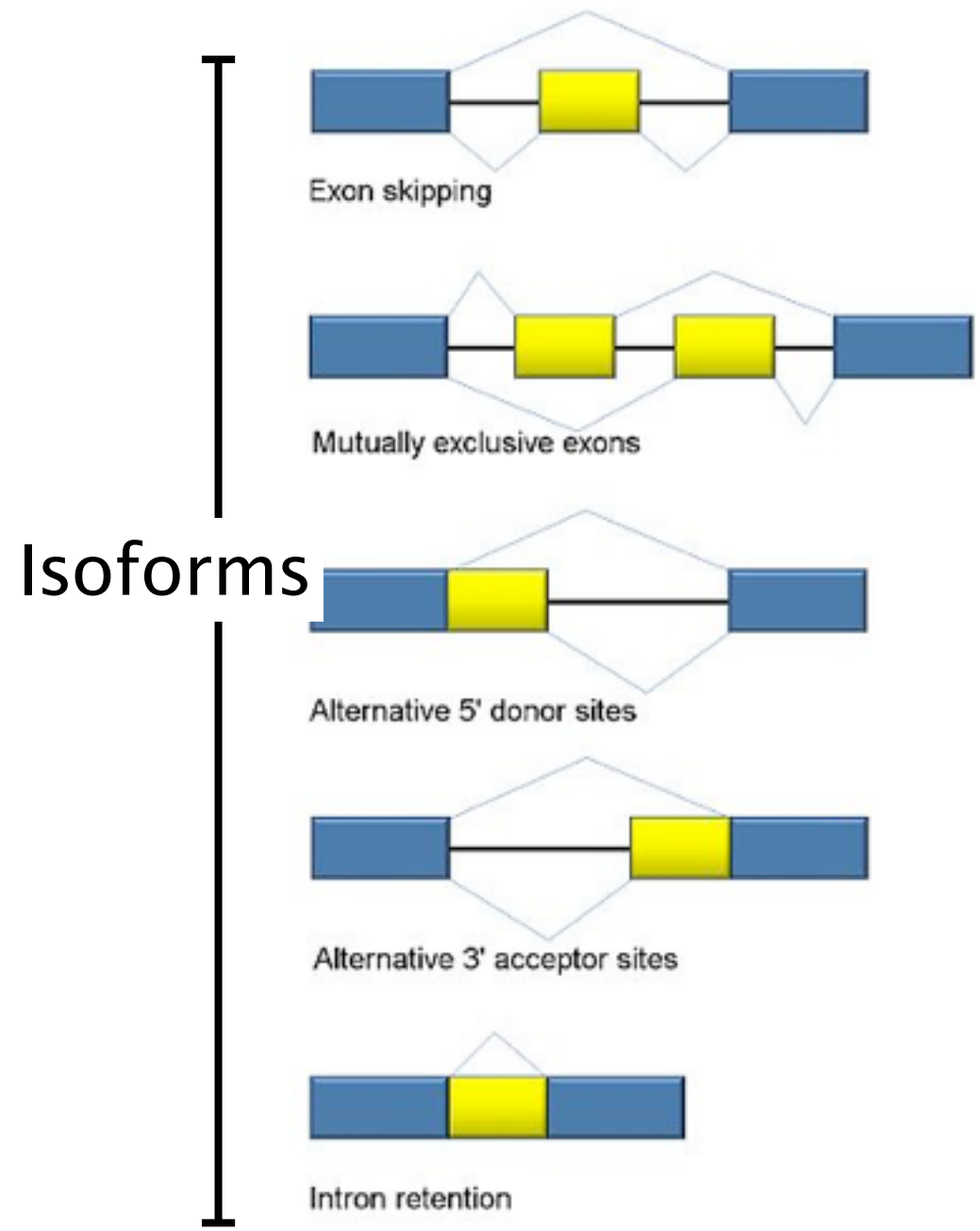
Height of nucleotide is proportional to its frequency at that position

Top image: http://www.web-books.com/MoBio/Free/Ch5A4.htm

Bottom images: Padgett, R. A. and Burge, C. B. 2005. Splice Sites. eLS: http://onlinelibrary.wiley.com/doi/10.1038/npg.els.0005044/full

Alternative splicing





Human genome has ~20K genes, ~95% of genes with >1 exon have >1 isoform, total # distinct isoforms likely >100K.

Pan, Qun, et al. "Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing." Nature genetics 40.12 (2008): 1413-1415.

Images: http://en.wikipedia.org/wiki/Alternative_splicing

Alternative splicing

Dystrophin (DMD) is the largest protein-coding gene in the human reference genome, spanning a total of 2.2 MB, while Titin (TTN) has the longest coding sequence (80,780 bp), the largest number of exons (364), and the longest single exon (17,106 bp). Over the whole genome, the median size of an exon is 122 bp (mean = 145 bp), the median number of exons is 7 (mean = 8.8), and the median coding sequence encodes 367 amino acids (mean = 447 amino acids; Table 21 in [3]).

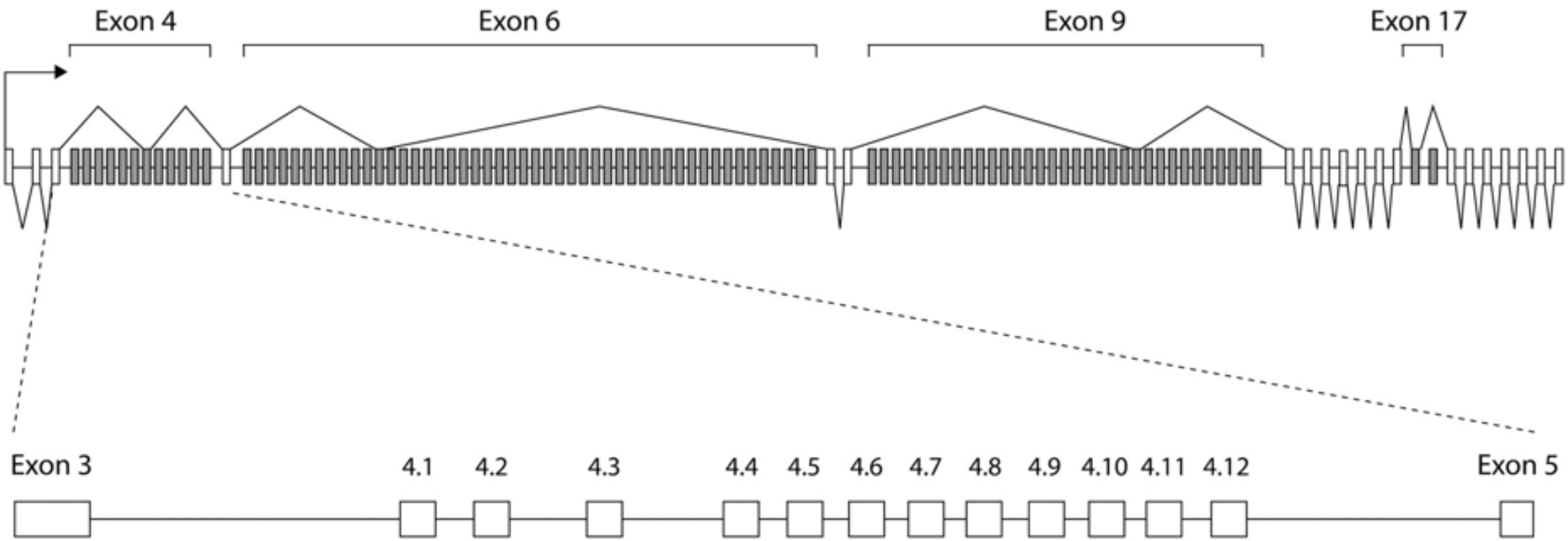
Protein	Chrom	Gene	Length	Exons	Exon length	Intron length	Alt splicing
Breast cancer type 2 susceptibility protein	13	BRCA2	83,736	27	11,386	72,350	yes
Cystic fibrosis transmembrane conductance regulator	7	CFTR	202,881	27	4,440	198,441	yes
Cytochrome b	MT	MTCYB	1,140	1	1,140	0	no
Dystrophin	X	DMD	2,220,381	79	10,500	2,209,881	yes
Glyceraldehyde-3-phosphate dehydrogenase	12	GAPDH	4,444	9	1,425	3,019	yes
Hemoglobin beta subunit	11	HBB	1,605	3	626	979	no
Histone H1A	6	HIST1H1A	781	1	781	0	no
Titin	2	TTN	281,434	364	104,301	177,133	yes

http://en.wikipedia.org/wiki/Human_genome

Human genome has ~20K genes, ~95% of genes with >1 exon have >1 isoform, total # distinct isoforms likely >100K.

Pan, Qun, et al. "Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing." Nature genetics 40.12 (2008): 1413-1415.

Alternative splicing: extreme example



The D. melanogaster Dscam gene contains 115 exons spanning ~60,000 bp. Twenty exons are constitutively spliced (open boxes) and 95 exons are alternatively spliced (shaded boxes). The alternatively spliced exons are organized into four clusters (exons 4, 6, 9, and 17) that contain 12, 48, 33, and 2 alternative exons each. The exons within each cluster are alternatively spliced in a mutually exclusive manner.

Image and caption from: Celotto, Alicia M., and Brenton R. Graveley. "Alternative splicing of the Drosophila Dscam pre-mRNA is both temporally and spatially regulated." Genetics 159.2 (2001): 599-608.

RNA sequencing

RNA-seq sequences mature mRNAs (including UTRs and poly-A tail)

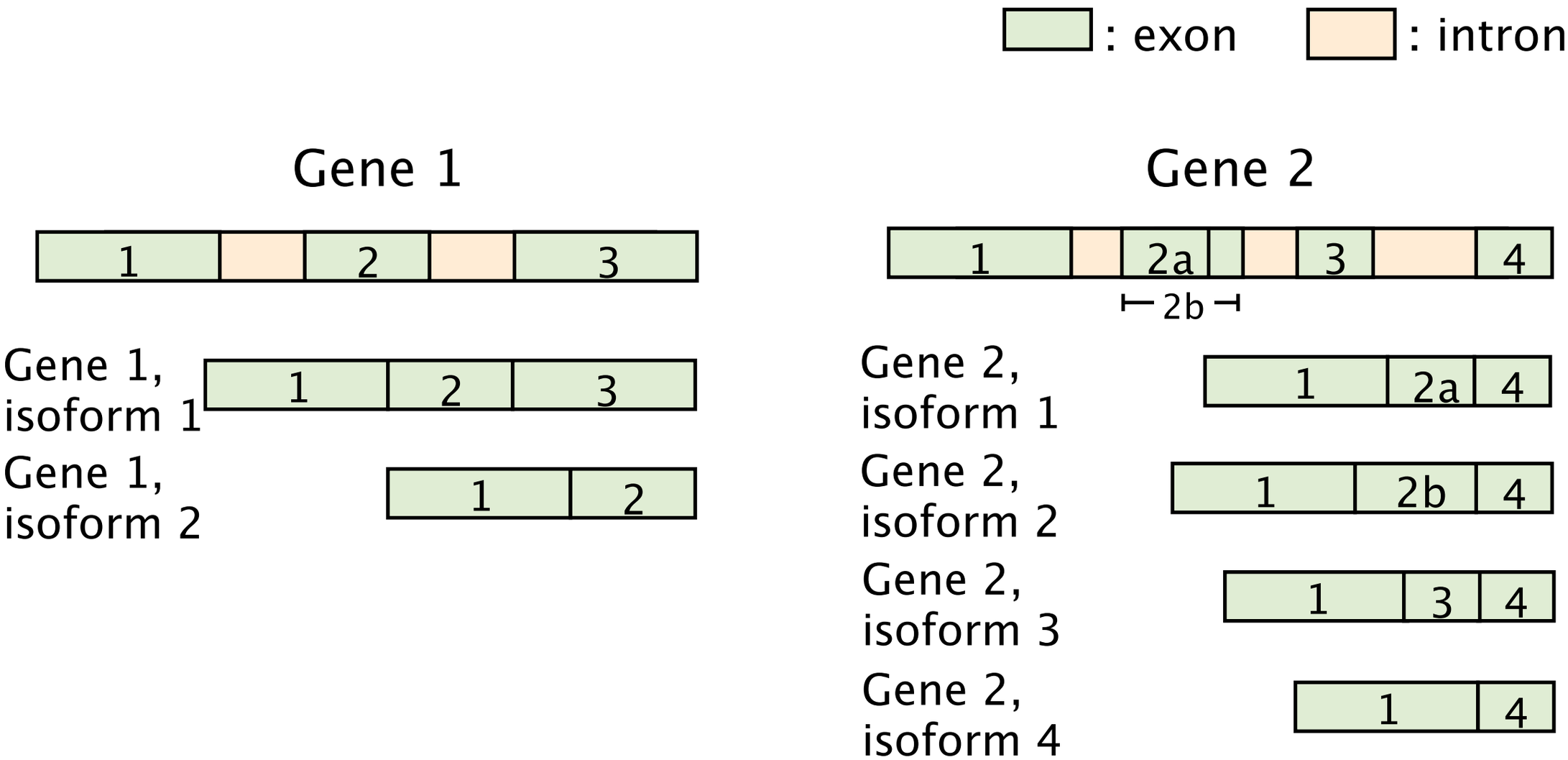
...so now we can measure abundance of particular versions (AKA isoforms, transcripts) of the gene

Isoform: particular concatenation of exons

This is more information, and more relevant information, than just a per-read expression measurement. But can we possibly do a good job of estimating this?

RNA sequencing

To fully measure the transcriptome, we'd like to measure abundance of all isoforms of all genes



RNA sequencing

Q1: What isoforms are there?

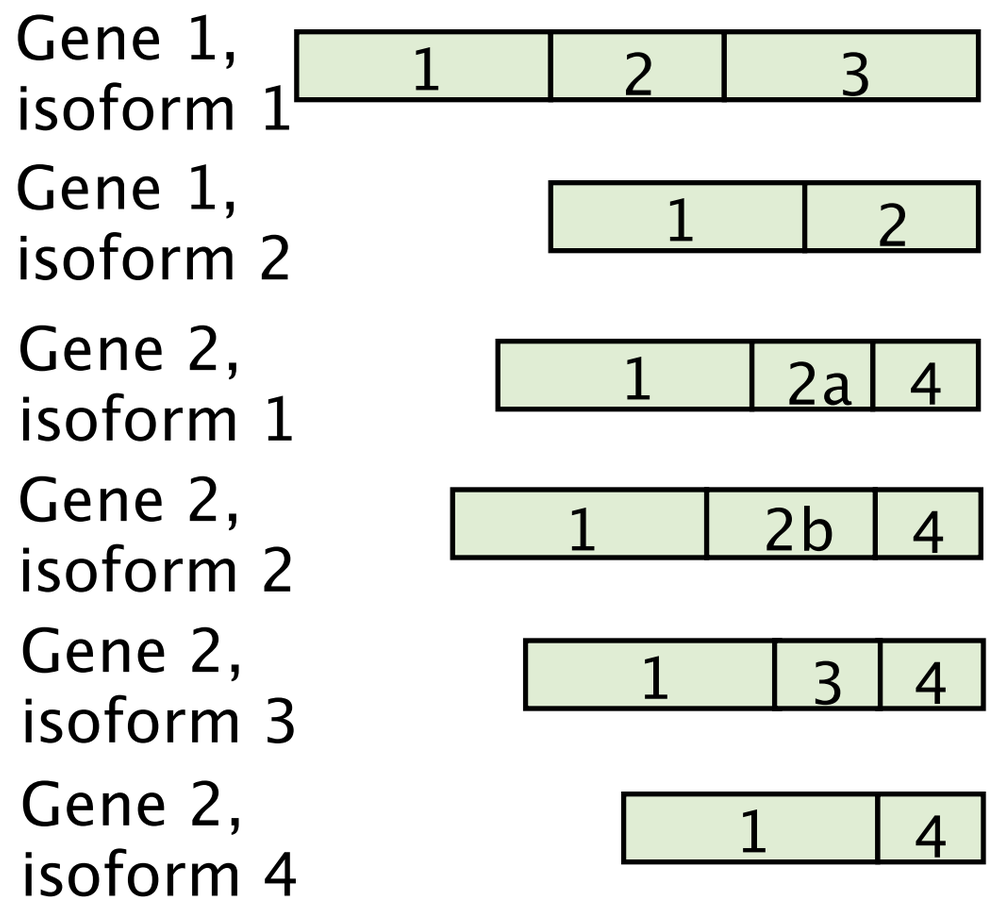
What does a read tell us? Assume paired-end reads

Hard to answer this without first thinking about what prior information we want to use

RNA sequencing: approach 1

Align read to collection of previously-observed and/or hypothetical isoforms

Alignment with P = read, T = isoforms in database
Best alignment is our best guess as to which isoform read came from



Pro: simple, fast

Con: Problematic to assume we know what isoforms are present to begin with. Must disentangle lots of repetitive alignments.

RNA sequencing: approach 2

Align reads to genome

Pro: don't need foreknowledge of what isoforms are relevant

Con: Because of introns, read doesn't necessarily align contiguously. Algorithms for spliced alignment are slower and more complex than those for typical end-to-end alignment.

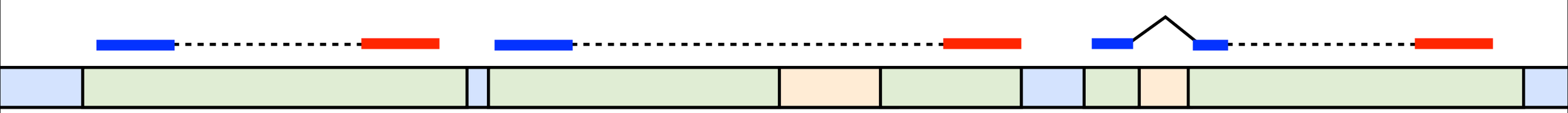
RNA sequencing: approach 2

Align reads to genome

Some fragment don't span introns

Some fragments overlap intron(s), but neither mate overlaps an intron

Some fragments have mates that overlap introns



: exon

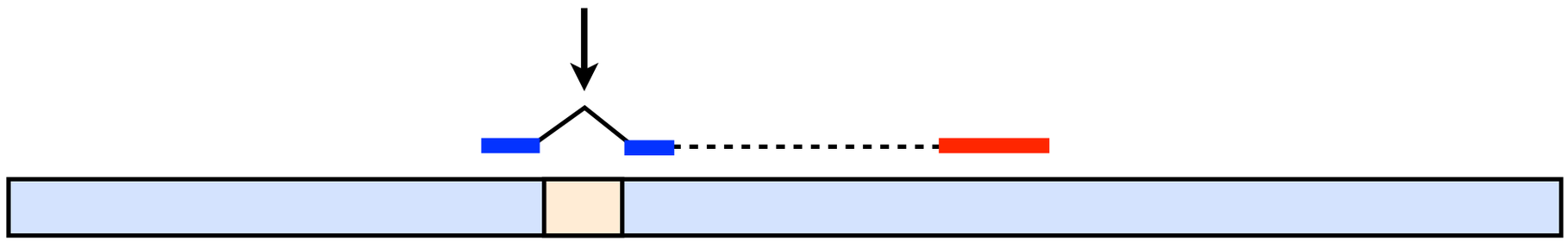
: intron

Distance between mates signals length of spanned intron(s)

Mate aligning "across" a spanned intron reveals intron boundaries (we hope)

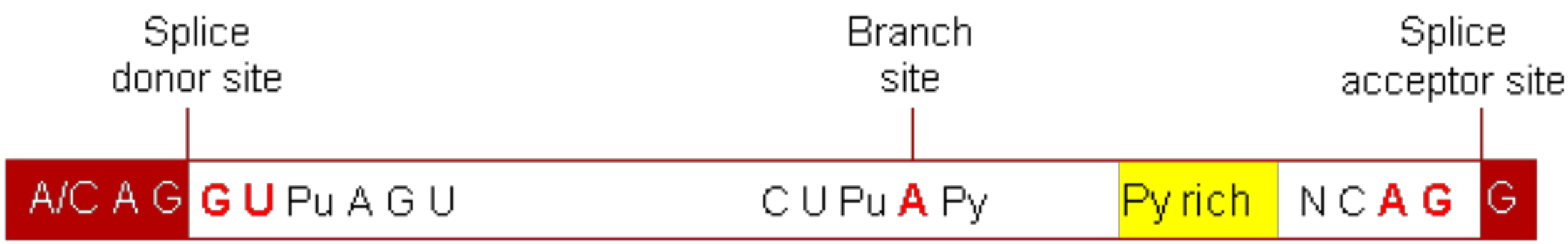
RNA sequencing: approach 2

How to find an alignment like this; i.e. a spliced alignment?



Some approaches (e.g. dynamic programming) naturally allow short gaps, but introns could be quite long (e.g. >10K nucleotides)!

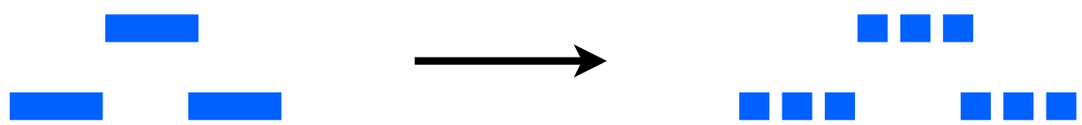
Seems like we should be able to use sequence signals to our advantage



TopHat aligns RNA-Seq reads

Slide by Cole Trapnell

Reads are fragmented into little pieces

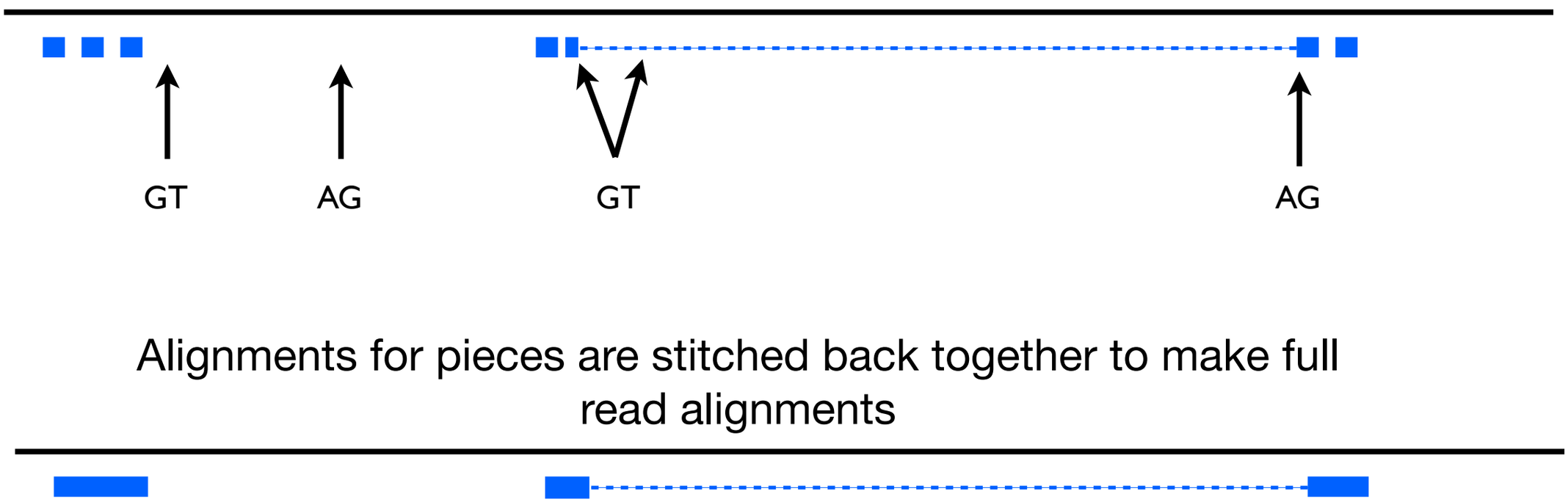


Pieces are aligned, and reads whose pieces map far apart are used to tag possible splice sites

TopHat aligns RNA-Seq reads

Slide by Cole Trapnell

Pieces that didn't align initially are used to confirm splice sites



Alignments for pieces are stitched back together to make full read alignments

RNA sequencing: approach 3

Assemble RNA-seq reads from scratch (de novo)

Rail-RNA's approach

Rail-RNA uses approach #2; it aligns RNA-seq reads to the genome in a spliced fashion

- Uses ideas from the now-extensive literature on spliced alignment. Some relevant papers; MapSplice is highly recommended reading:

  Trapnell C, et al. **TopHat**: discovering splice junctions with RNA-Seq. Bioinformatics. 2009 May 1;25(9):1105-11.

  Wang K, et al. **MapSplice**: accurate mapping of RNA-seq reads for splice junction discovery. Nucleic Acids Res. 2010 Oct;38(18):e178.

  Kim D, et al. **TopHat2**: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biol. 2013 Apr 25;14(4):R36.

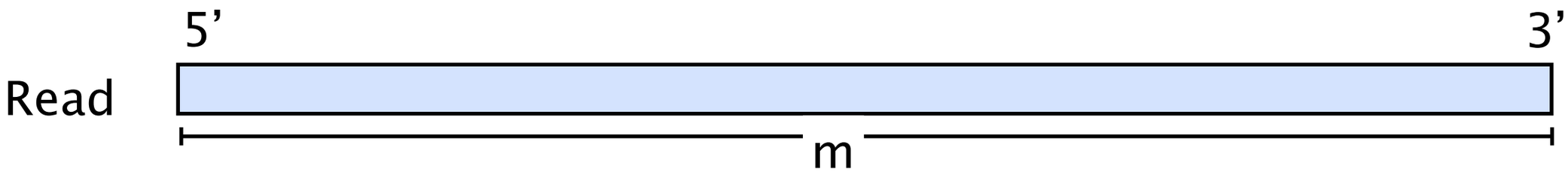
- Older but still rather relevant:

  Zhang M, Gish W. Improved spliced alignment from an information theoretic
  approach. Bioinformatics. 2006 Jan 1;22(1):13-20.

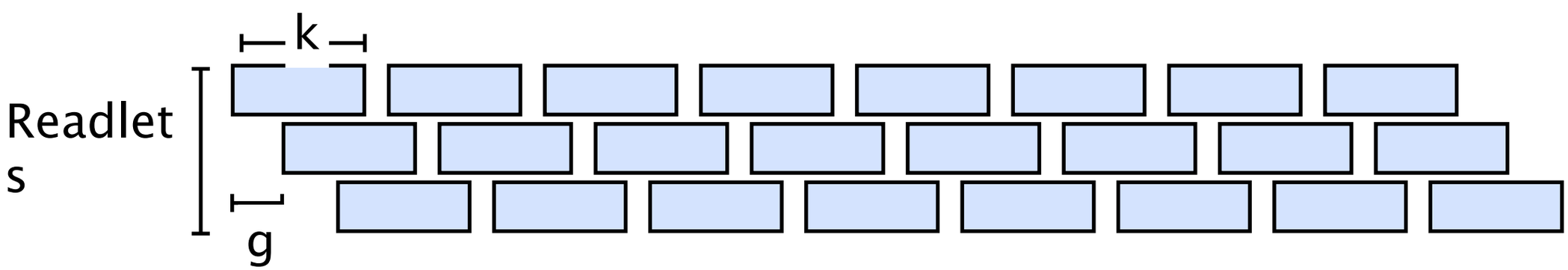
  De Bona F, et al. Optimal spliced alignments of short sequence reads. Bioinformatics. 2008 Aug 15;24(16):i174-80.

Following slides describe what’s happening in align.py

Readlets



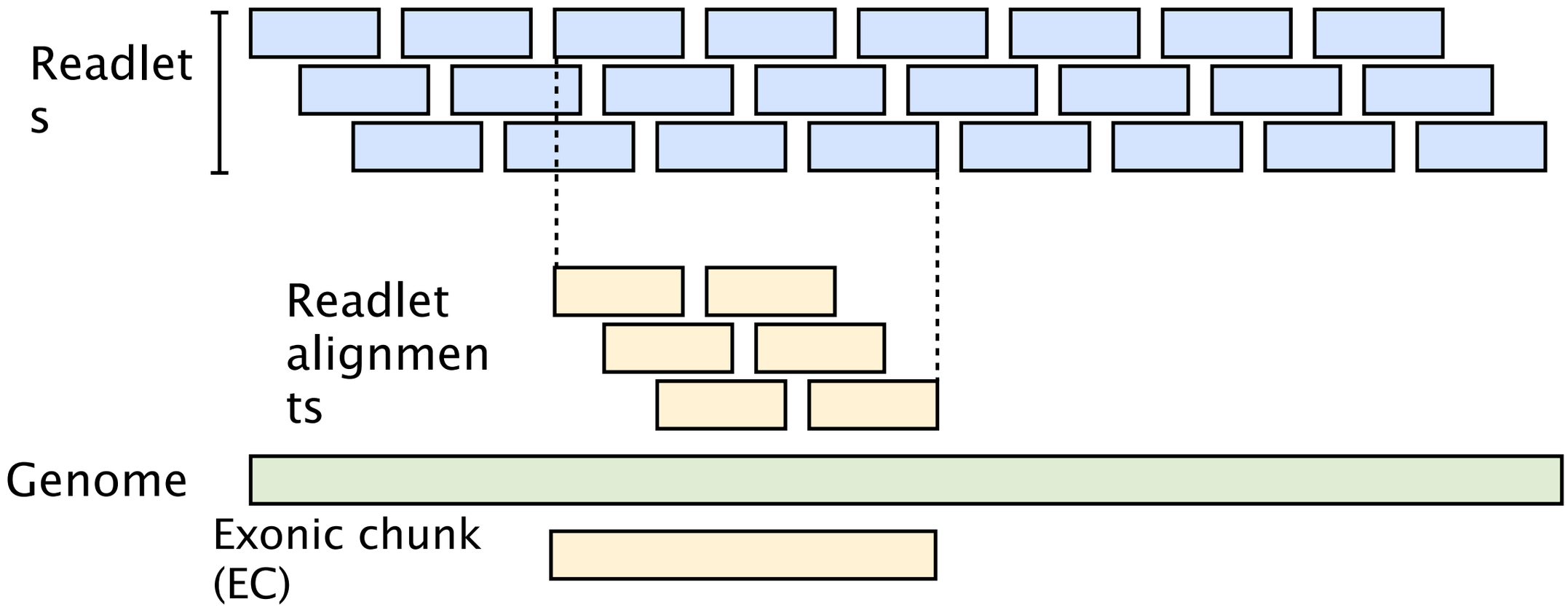
E.g. take a length-k substring every g nucleotides starting at 5' end



Depending on m, k, g, extreme 3' end might not be covered by a readlet -- we'll fix

ECs

Align the readlets, then colaesce (union) the reference intervals aligned to into exonic chunks, ECs for short.



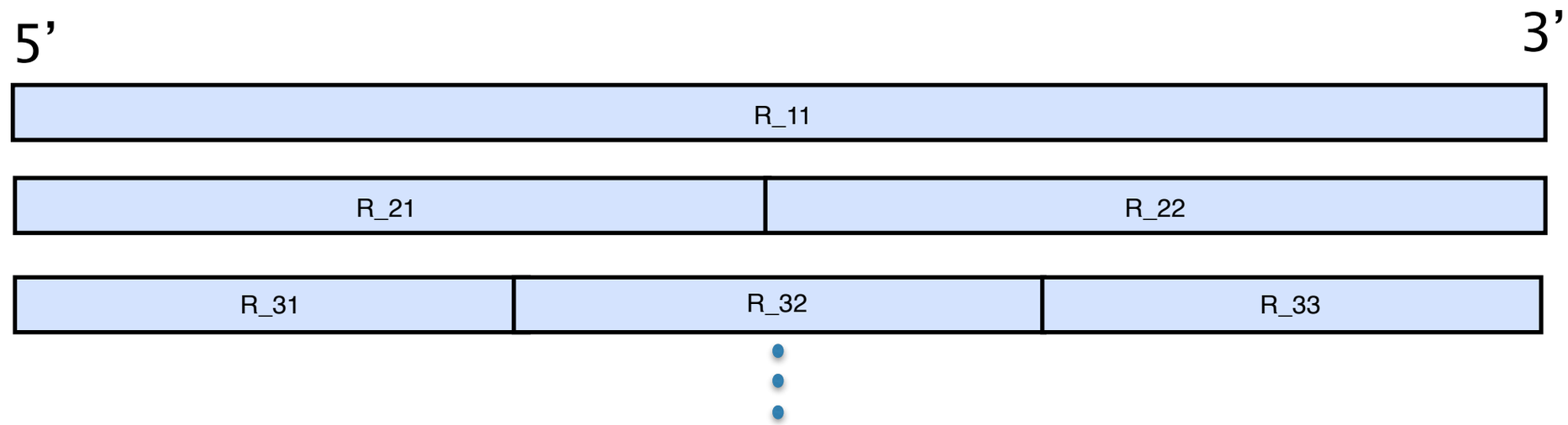
Readlets are not aligned in a spliced fashion. We'll try to infer splicing later.

For now, assume we allow a very small number of edits and that we ignore all alignments for readlets that align repetitively

ECs: an alternative

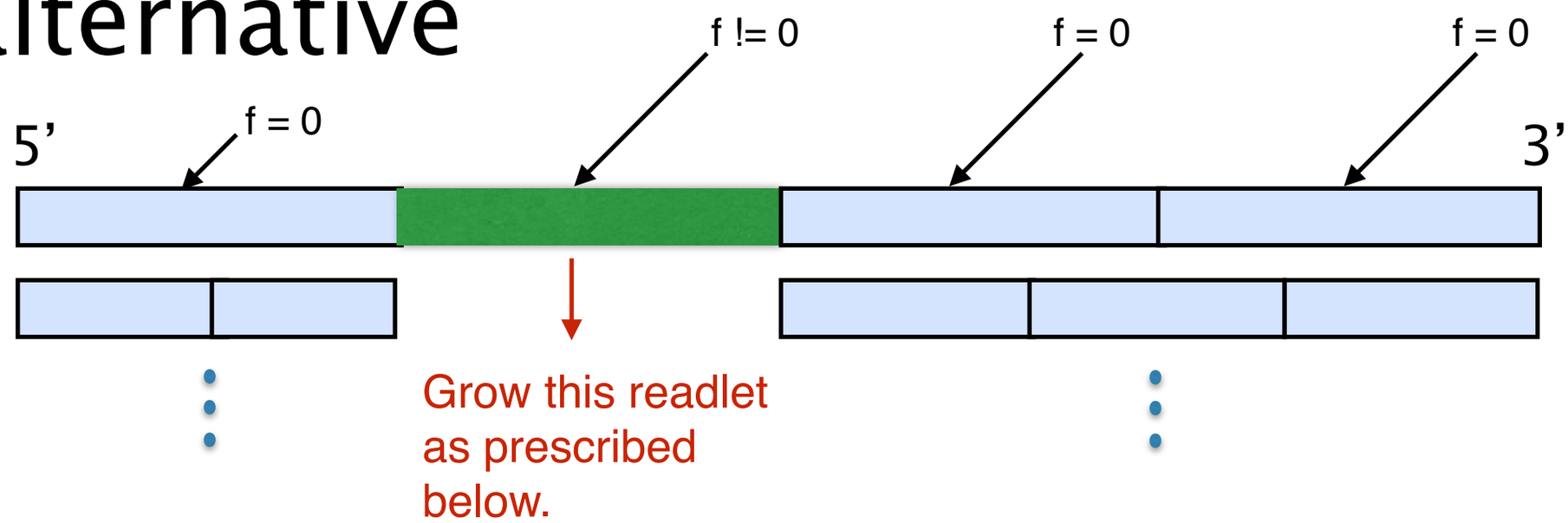
Search for ECs of maximal size by alignment-dependent readletizing as follows.

Let R be a readlet, and f(R) = the number of mappings of that readlet. (max f(R) = N from bowtie -k N .)



- Start with full read R_11.
  - If f(R_11) = 1, assume a full EC with no splice sites.
  - If f(R_11) > 1, EITHER throw out the read (easy for now) OR probabilistically assign to one from set of possible mappings according to coverage. (So one could initially perform alignment of all reads permitting few edits and determine the coverage distribution from unique mappings.)
  - If f(R_11) = 0, split R_11 into two readlets R_21 and R_22 of equal size, and if f(R_21) = f(R_22) = 0, split R_1 into three readlets R_31, R_32, and R_33 of equal size. Continue these splits until f acting on a readlet is nonzero OR some lower bound on size of readlet is reached. If lower bound is reached, throw out R_11.

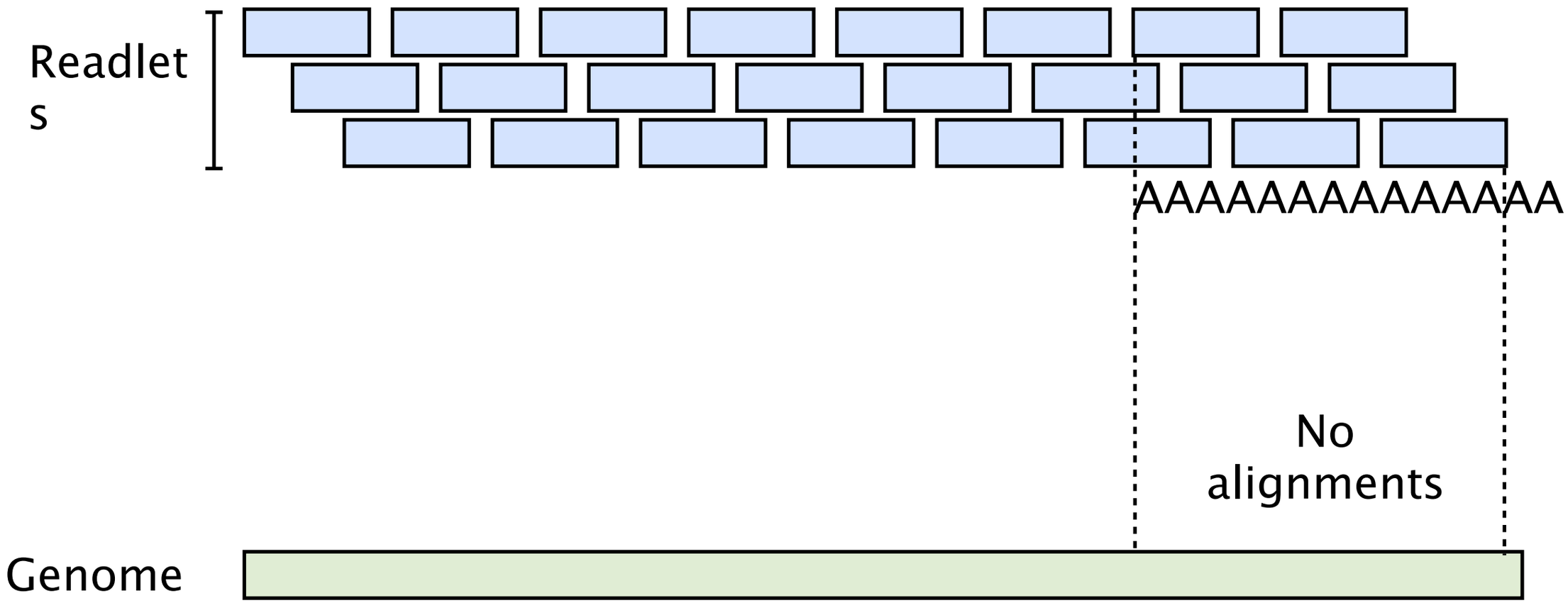
ECs: an alternative



- When f(R) for some readlet R is not zero, keep splitting contiguous readlets for which f != 0 on either side of R, as above. Follow pattern on previous slide.
- Grow R (for which f !=0) by expanding it on either side, performing binary searches for the borders between f = 0 and f != 0 OR leave R alone and label it an EC.

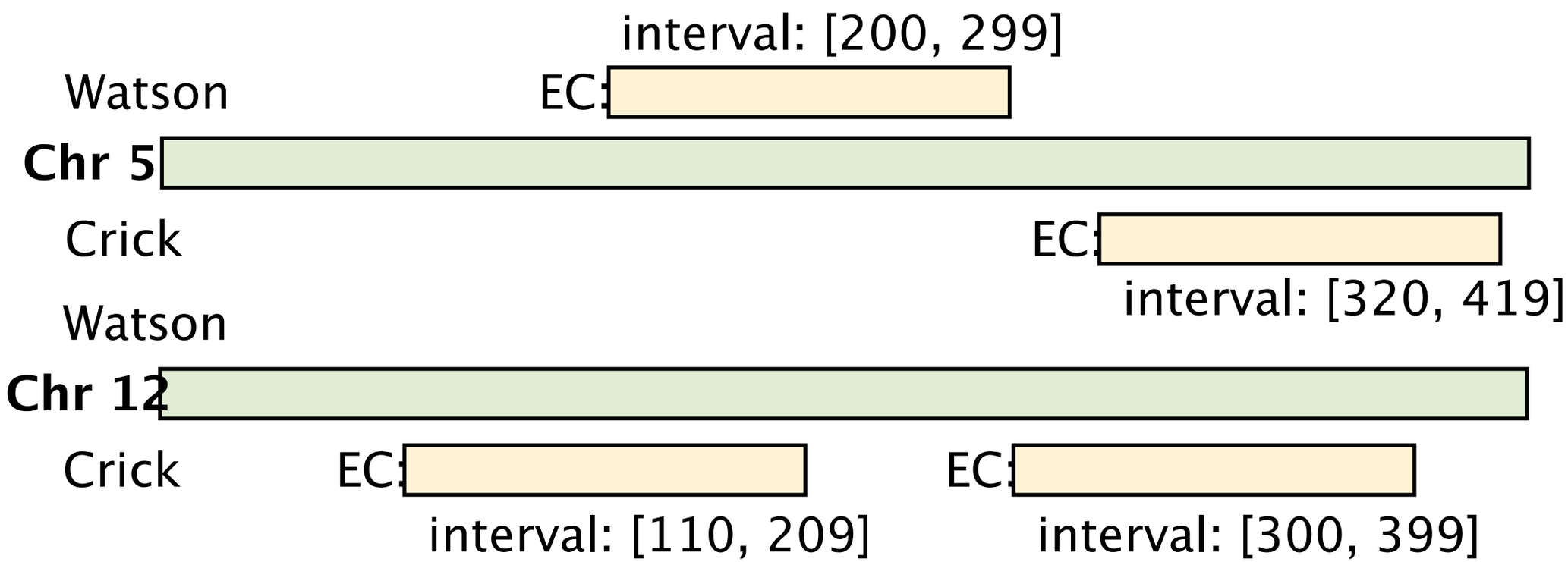
ECs

Note that poly-A tails will align repetitively and won't be included in ECs



ECs

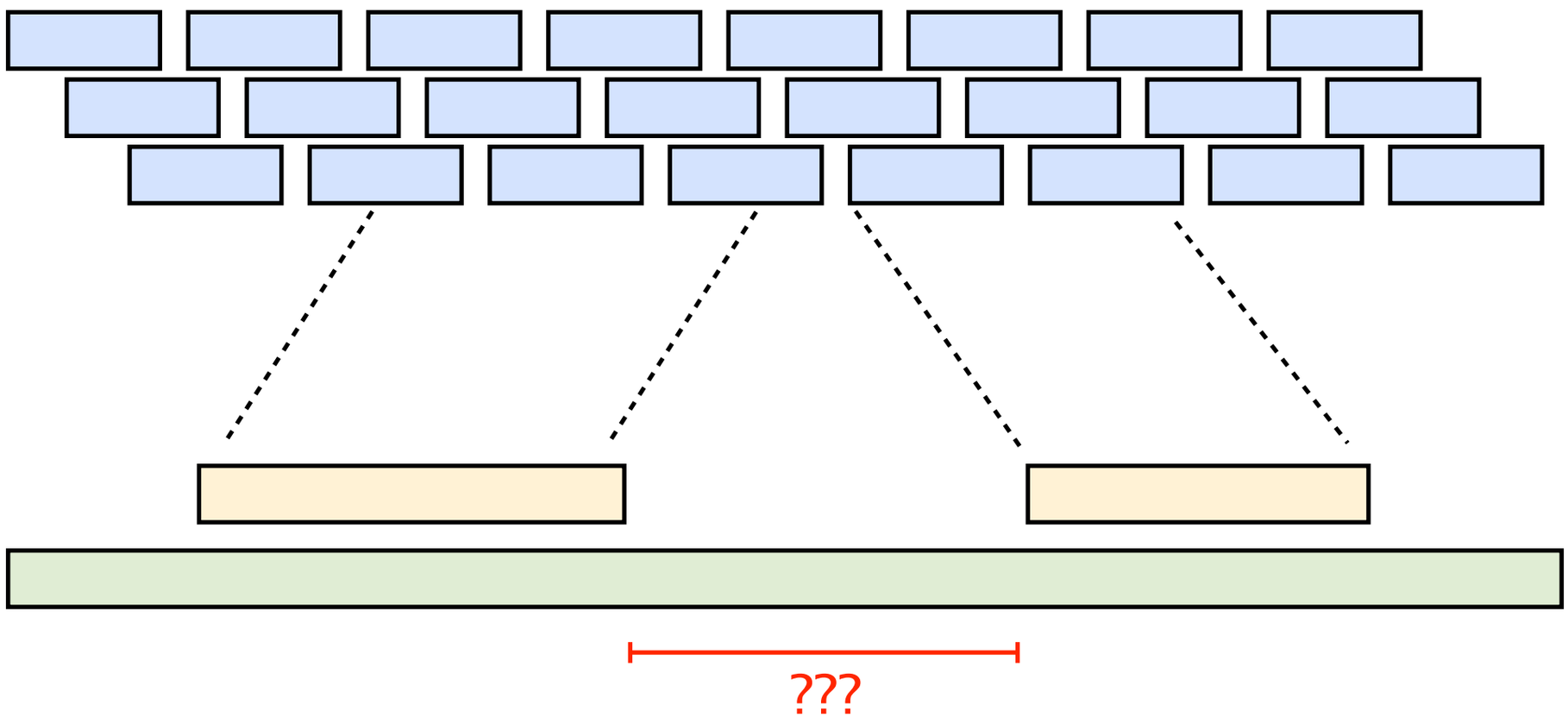
Compose “raw” ECs into per-read (per-mate for paired-end reads) data structure encoding where they all fell:



{ (‘Chr5’, ‘Watson’) => [(200, 299)],
(‘Chr5’, ‘Crick’) => [(320, 419)],
(‘Chr12’, ‘Crick’) => [(110, 209), (300, 399)] }

Refining ECs

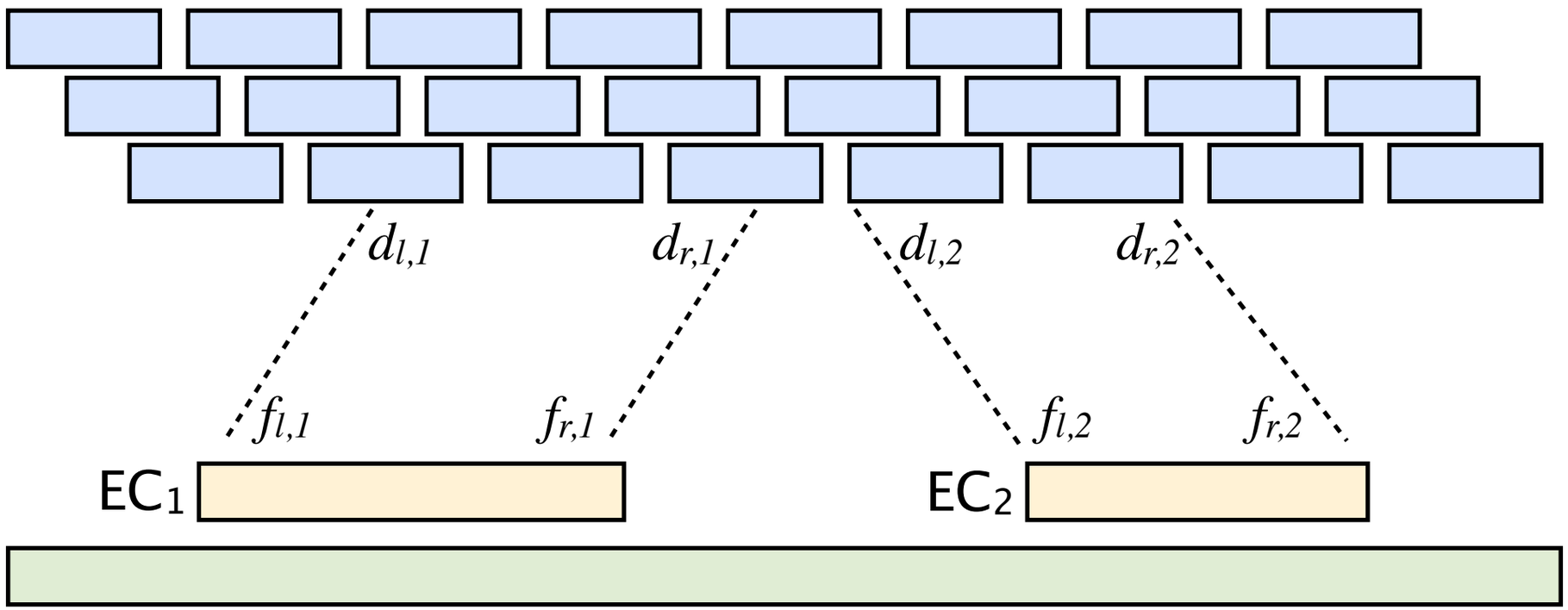
Say we have two ECs that align to the same strand of the same chromosome and are close to each other. What could this mean?



Could be there's an intron between them, or could be intervening readlets failed to align for other reasons (variation, sequencing error, repetitiveness)

Refining ECs

Assume EC_1 is to the left of EC_2. Let $d_{l,1}$ and $d_{r,1}$ be offsets of leftmost and rightmost read characters involved in EC_1. Let $d_{l,2}$ and $d_{r,2}$ be the same for EC_2. Let $f_{l,1}$, $f_{r,1}$, $f_{l,2}$, $f_{r,2}$ be the same for the reference characters involved in EC_1 and EC_2.



The following scenarios are possible:

$d_{l,2} - d_{r,1} \approx f_{l,2} - f_{r,1}$: probably not an intron – intervening readlets failed to align

$d_{l,2} - d_{r,1} < f_{l,2} - f_{r,1}$: probably an intron (but could be sizable deletion w/r/t ref)

$d_{l,2} - d_{r,1} > f_{l,2} - f_{r,1}$: unusual, but could be large insertion w/r/t ref

Refining ECs

What do we do in each of these cases?

$d_{l,2} - d_{r,1} \approx f_{l,2} - f_{r,1}$: probably not an intron – intervening readlets failed to align

Do DP filling, discussed later

$d_{l,2} - d_{r,1} < f_{l,2} - f_{r,1}$: probably an intron (but could be sizable deletion w/r/t ref)

Assume it's an intron and do DP framing, discussed later. If the result from DP framing is suspicious, consider calling it a deletion.

$d_{l,2} - d_{r,1} > f_{l,2} - f_{r,1}$: unusual, but could be large insertion w/r/t ref

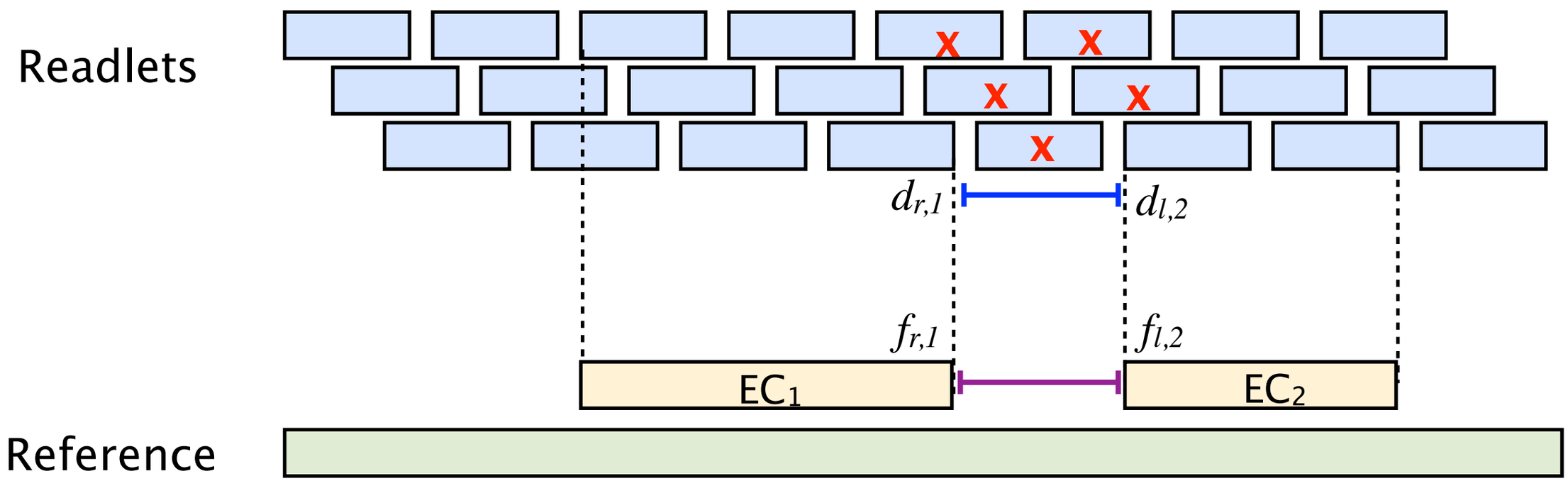
Ignore for now. In theory we could frame a DP to pinpoint the insertion point w/r/t reference.

$d_{l,2} \leq d_{r,1}$: some read nucleotides aligned in both ECs

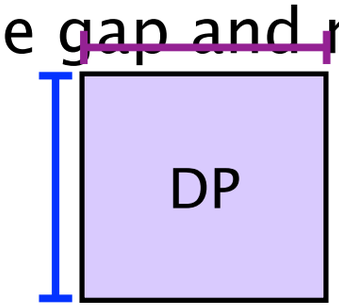
This could coincide with either of the first two cases.

DP filling

In the case where $d_{l,2} - d_{r,1} \approx f_{l,2} - f_{r,1}$, the gap is probably due to intervening readlets failing to align because of variants, sequencing errors, and/or repetitive sequence



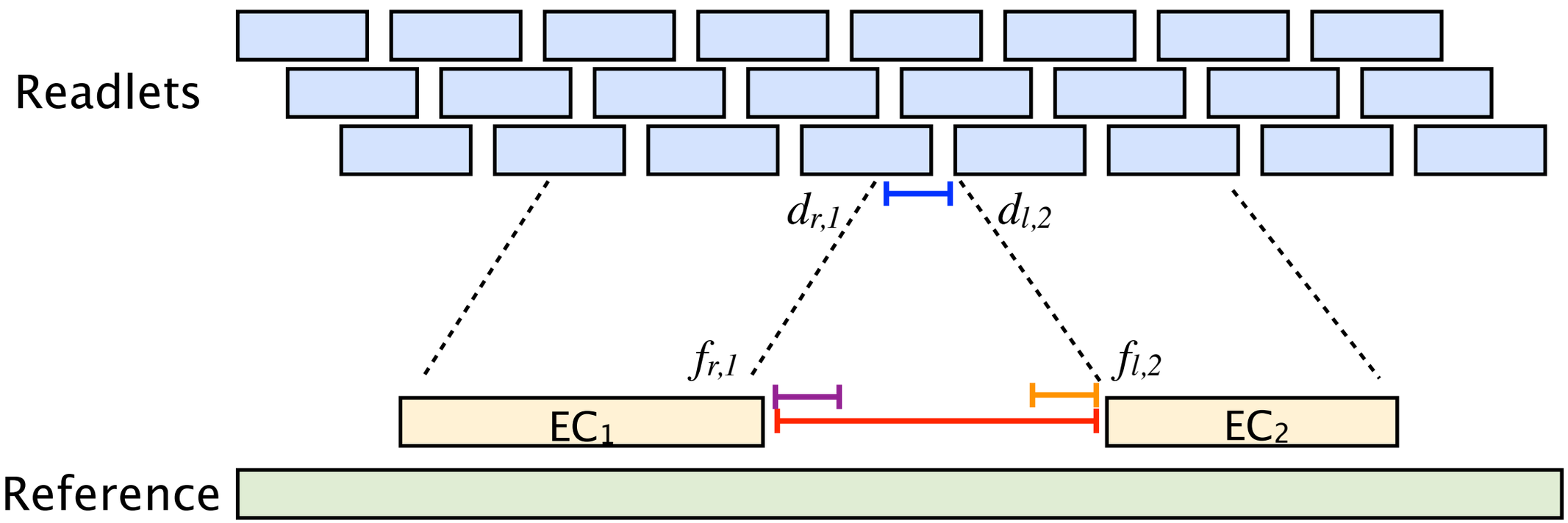
Our goal is to figure out if there's enough sequence similarity between blue read substring and the purple reference substring to justify "filling in" the gap and merging the two ECs.



We can set up a corresponding dynamic programming problem, looking for e.g. edit distance. Then say, if % identity is > threshold then we fill in the gap.

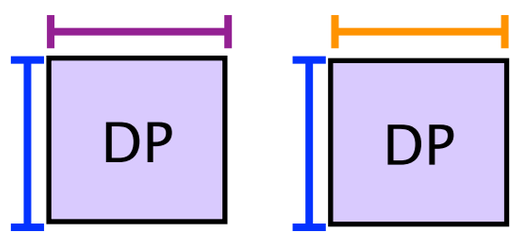
DP framing

In the case where $d_{l,2} - d_{r,1} < f_{l,2} - f_{r,1}$, the difference could be due to an intron. But we don't yet have a precise guess for the intron boundaries.

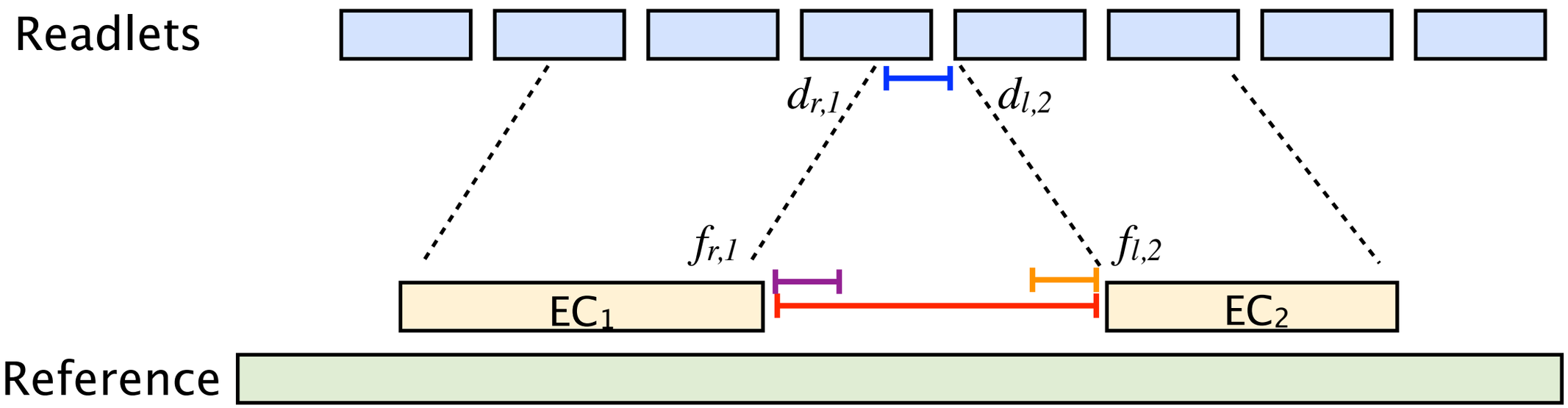


We need to take the as-yet-unaligned read characters (between $d_{l,2}$ and $d_{r,1}$) and allocate them to EC_1 & EC_2 in a way that maximizes overall similarily.

One method is to solve two dynamic programming alignment problems, corresponding to the blue, purple and orange intervals above:



DP framing



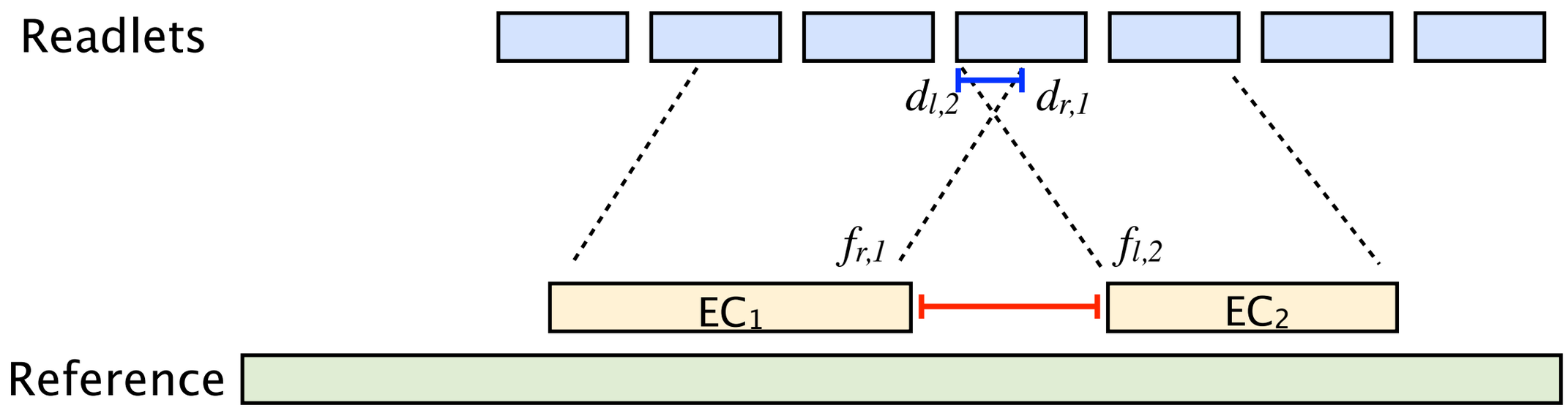
Fill these DP matrices using, say, the typical rules for edit distance. Fill the left matrix from upper-left to lower-right as usual. Fill the right matrix from lower-right to upper left.

might be beneficial to throw in some extra characters on either end of these intervals

Now locate the best row jump, i.e. the i such that jumping from row i of the left matrix to row i+1 of the right matrix minimizes edit distance (or maximizes some other score). i then gives the number of characters between $d_{l,2}$ and $d_{r,1}$ that should be added to EC_1 and remainder get added to EC_2.

DP framing

Note that DP might be overkill; most of the time there won't be gaps and we can do Hamming distance per spliced-align on p4 of MapSplice paper



Also note: $d_{l,2} \leq d_{r,1}$ is possible

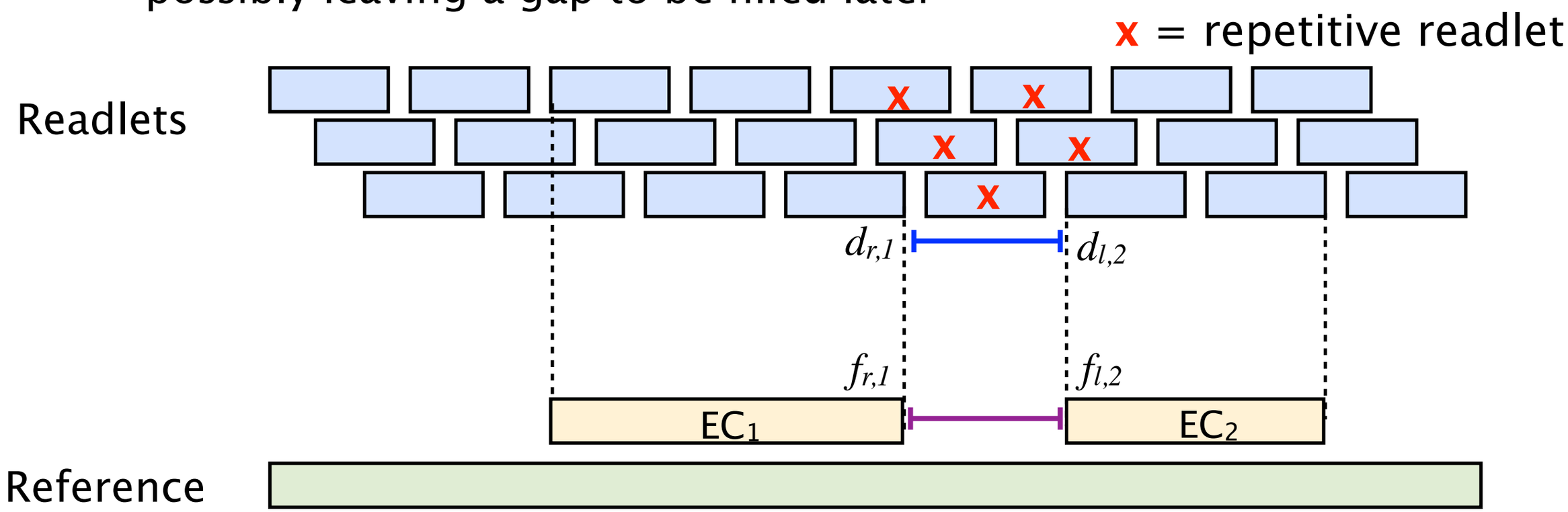
Readlet alignment

How should we align the readlets?

- How should we deal with readlets that align repetitively?
- What if the extreme ends of the read aren't covered by readlets?
- Are our methods efficient?

Repetitive readlets

Currently, we align the readlets with bowtie using the -m 1 option, meaning that a read that aligns to >1 location will be ignored, possibly leaving a gap to be filled later

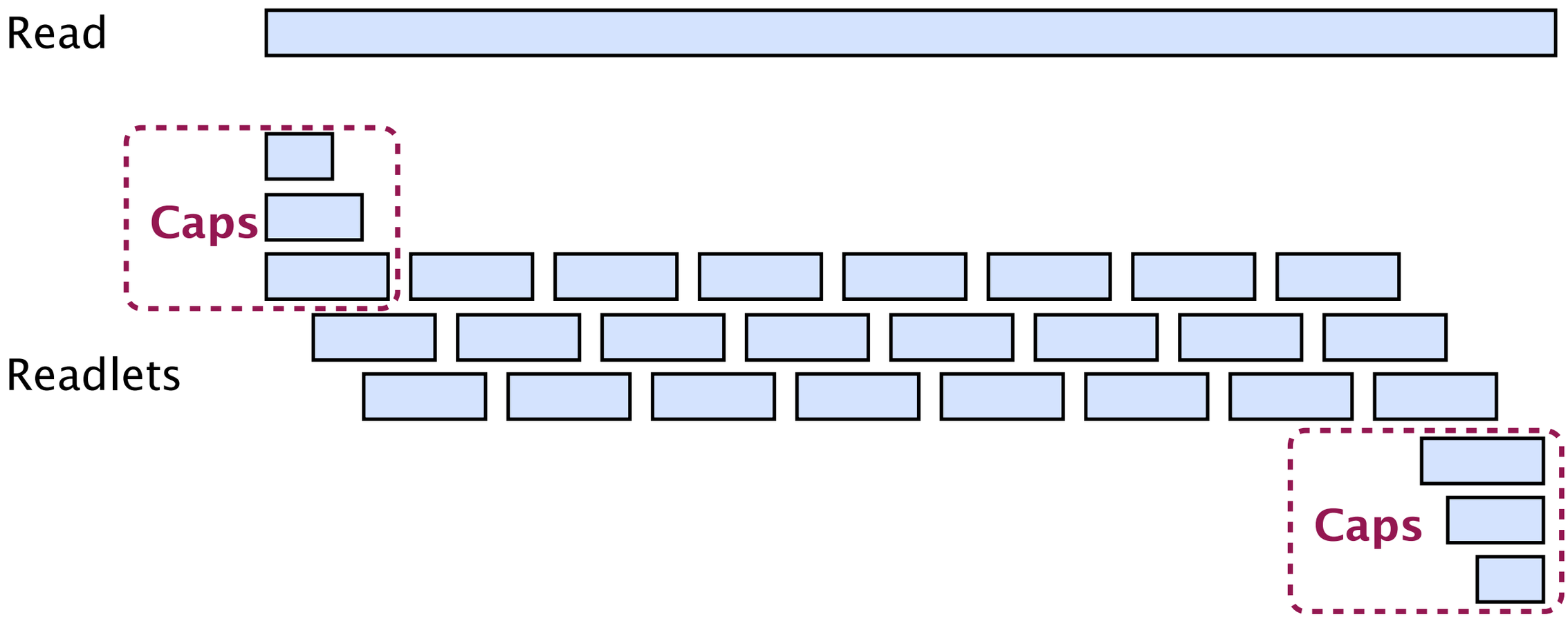


One alternative: Like TopHat, align using bowtie with -k N where N is fairly small (say, 3, 5, 10). Then sift through readlet alignments. We might find that the entire read aligns multiple places, in which case we might want to ignore it. Or we might find that several readlets align uniquely and the remaining readlets align many places, including a place near the uniquely-aligned ones.

Extreme ends of reads

Currently we simply ignore the extreme ends of reads that are not covered by readlets.

Suggestion: always place some additional “cap” readlets, perhaps of a few different lengths, at both extreme ends



Efficiency

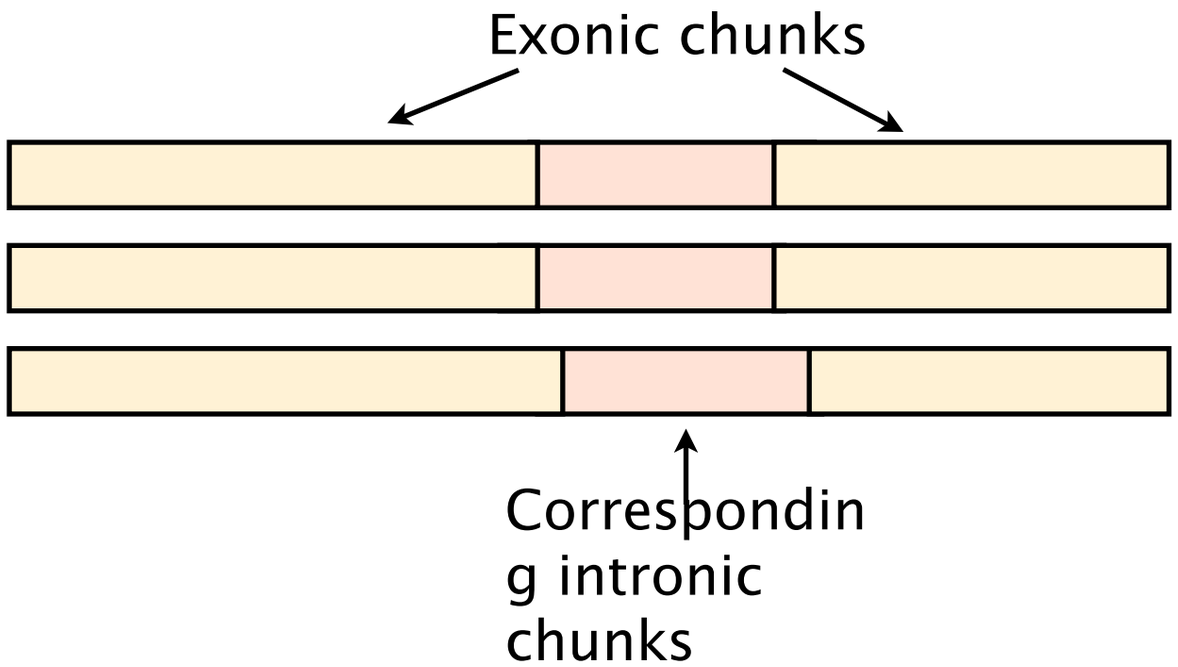
Our method of piping readlets to bowtie and then parsing its output is not terrible, but could be improved in a couple ways

- **Easy**: first try to align entire read; only if it does not have a convincing end-to-end alignment do we attempt readlet alignments
- **Hard**: push all of what happens in align.py into bowtie (or bowtie2)

Following slides describe what’s happening in intron2.py

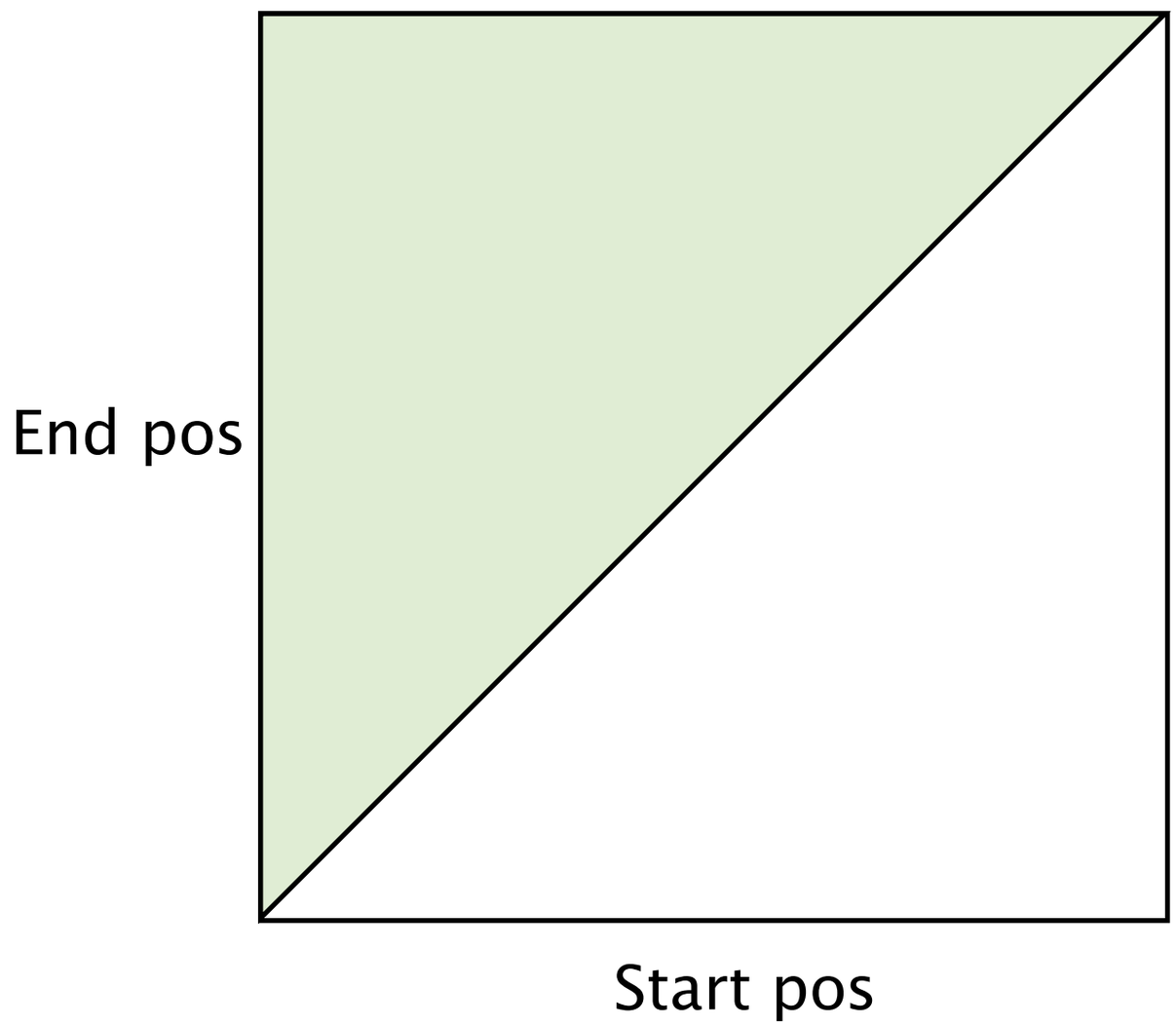
Picking the introns

Now we have intronic chunks which don't necessarily line up with the junction precisely, but hopefully come pretty close.



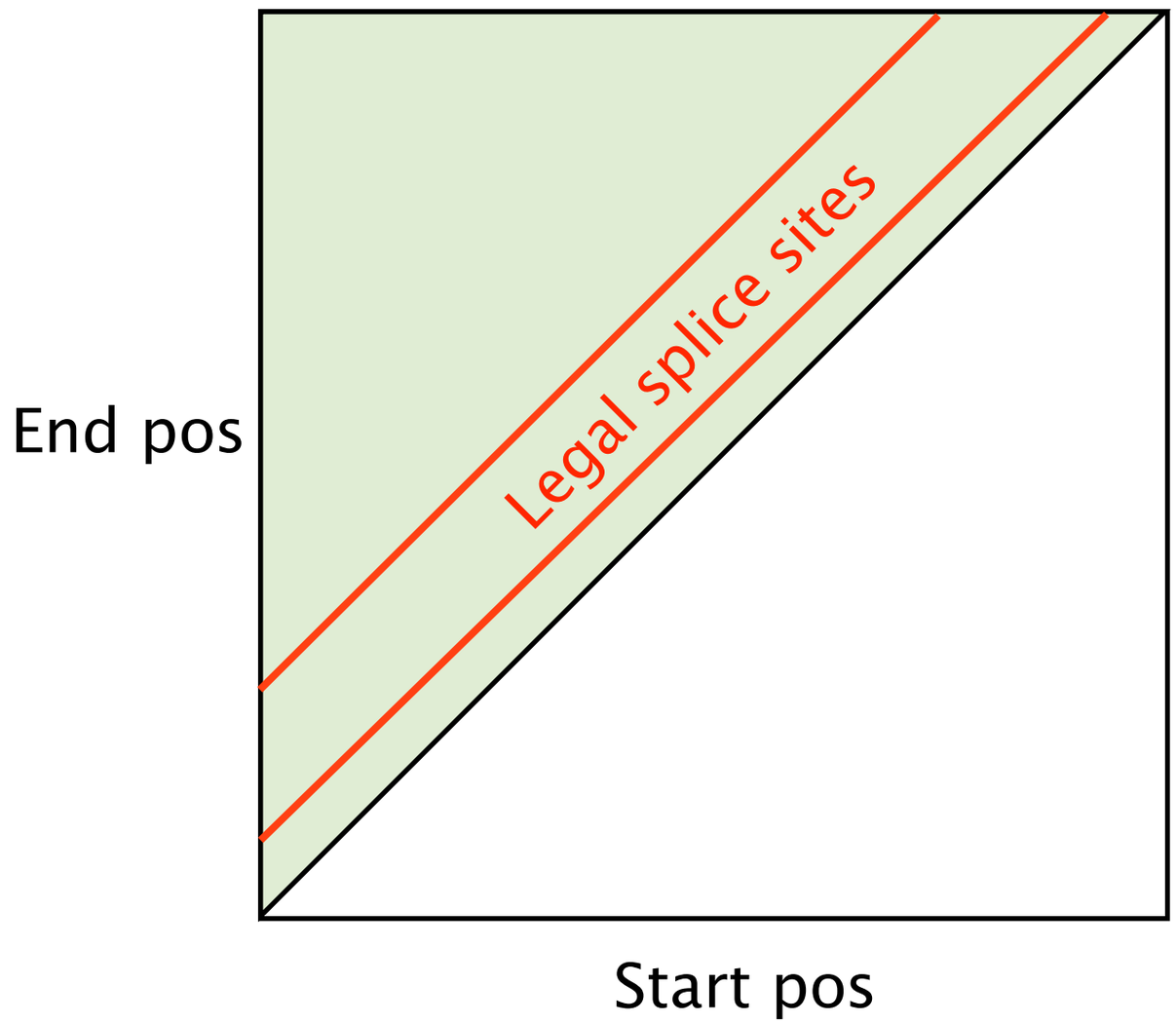
Picking the introns

Imagine the space of all possible intronic chunks, i.e. all possible <start, end> pairs s.t. start < end. These live in the upper triangle.



Picking the introns

Say we're only interested in introns with some minimum length (which could be very small) and some maximum length (say, 1M nucleotides). This limits us to diagonal strip near but above x = y.



Picking the introns

We can parallelize the intron selection process by dividing this band up into, say, vertical strips and giving different strips to different workers