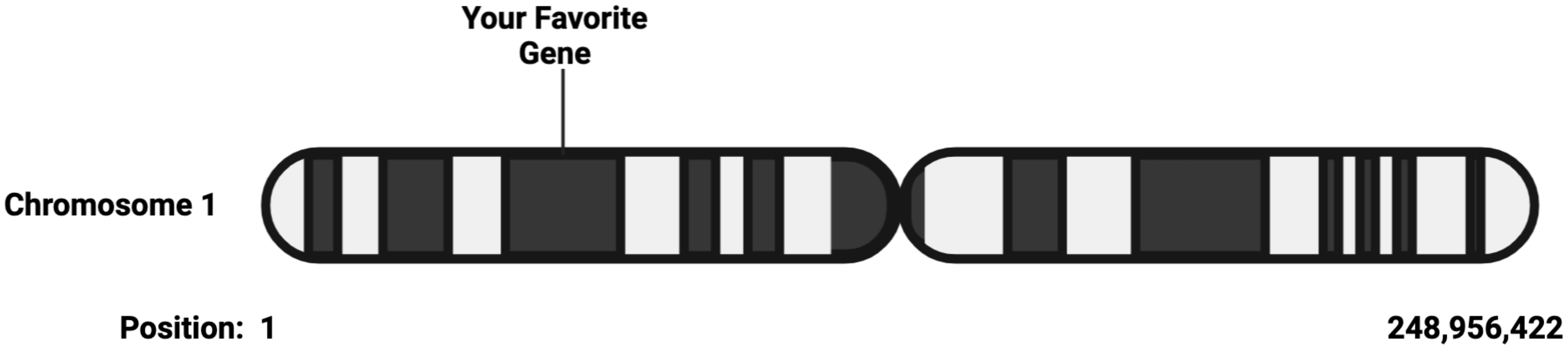
Introduction to Genome Arithmetic

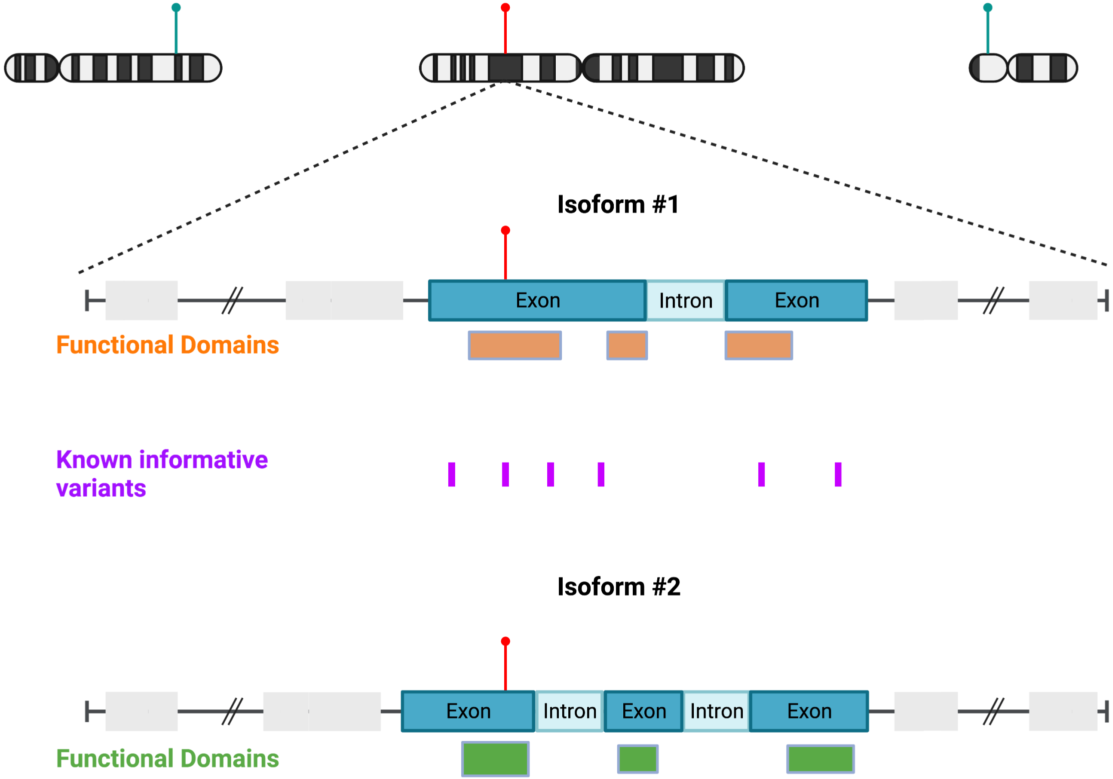
Aaron Quinlan, Joshua Mincer, Jason Kunisaki
CSHL Advanced Sequencing Technologies 2022
11/16/2022

A reference genome is a coordinate system

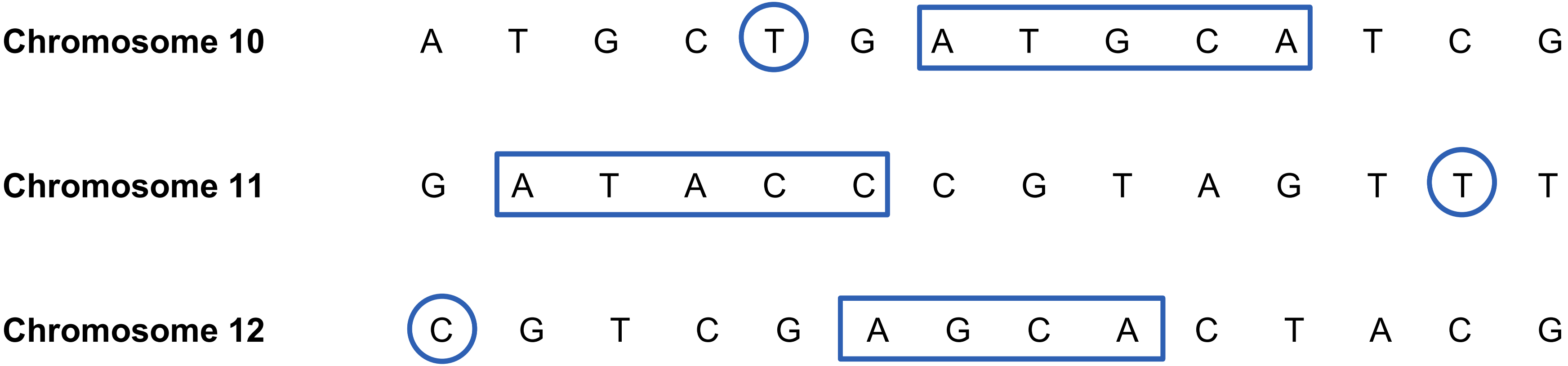


Genome coordinates are essential

- Identifying exact variant position
- Determining functional consequence of a variant
  - Variant in a functional domain?
  - Tumor vs normal comparisons
  - Rare in the population?
- **Designing a targeted sequencing panel**



Learning Objectives



Chromosome 10 A T G C T G A T G C A T C G

Chromosome 11 G A T A C C C G T A G T T T

Chromosome 12 C G T C G A G C A C T A C G

- What are **genome coordinates** and how are they used?
- How to incorporate **intervals** to analyze specific regions of the genome
- Concepts in **genome arithmetic** – **bedtools**
- High level strategy to generate a targeted sequencing panel
- Figures adapted from Obi Griffith's biostars tutorial and Aaron Quinlan's bedtools tutorial

Genome coordinates identify a specific location of interest in the reference genome

World coordinates:

- 41.8781ºN, 87,6298ºW
- Chicago

Chromosome 10 A T G C T G A T G C A T C G

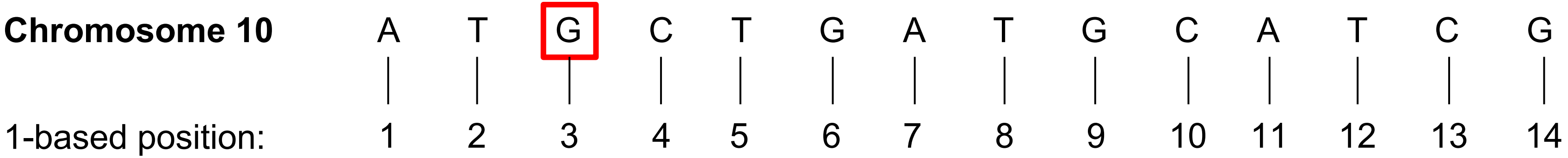
Genome coordinates:

- Chromosome: chr10
- Start: 3
- End: 3
- chr10:3-3

Genome coordinates:

- chr10
- Start: 8
- End: 10
- chr10:8-10

1-based system numbers nucleotides in a sequence

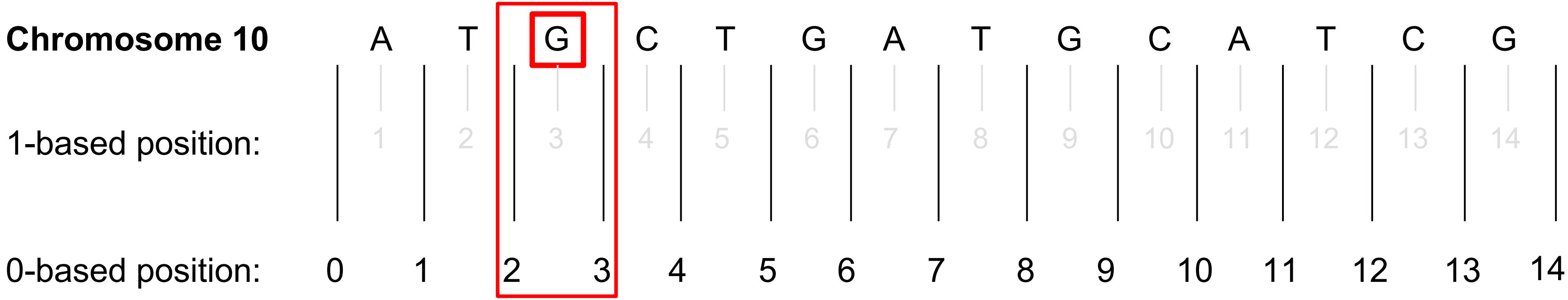


Chromosome 10	A	T	G	C	T	G	A	T	G	C	A	T	C	G
1-based position:	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Genome coordinates (1-based):

- Chromosome: chr10
- Start: 3
- End: 3
- chr10:3-3

0-based system numbers between nucleotides



Chromosome 10 A T G C T G A T G C A T C G

1-based position: 1 2 3 4 5 6 7 8 9 10 11 12 13 14

0-based position: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

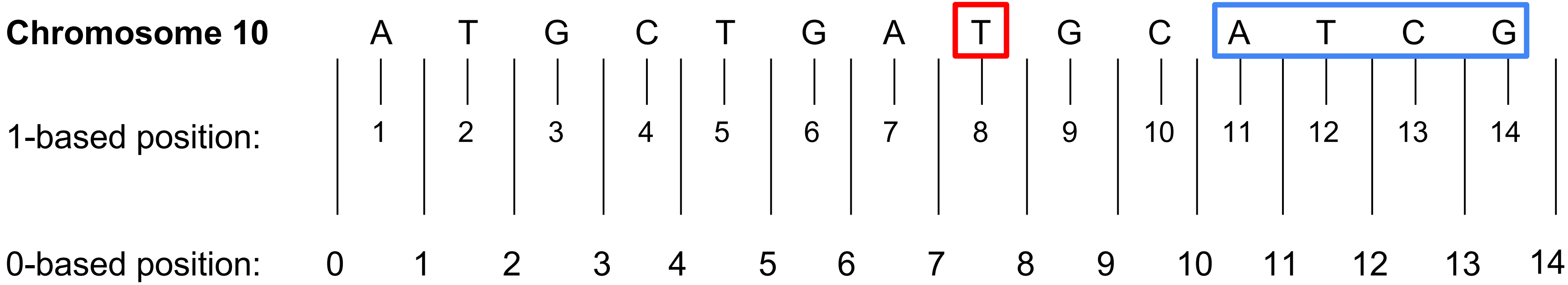
Genome coordinates (1-based):

- Chromosome: chr10
- Start: 3
- End: 3
- chr10:3-3

Genome coordinates (0-based):

- Chromosome: chr10
- Start: 2
- End: 3
- chr10:2-3

Practice exercises in 0 and 1 base coordinates



Chromosome 10 A T G C T G A T G C A T C G

1-based position: 1 2 3 4 5 6 7 8 9 10 11 12 13 14

0-based position: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

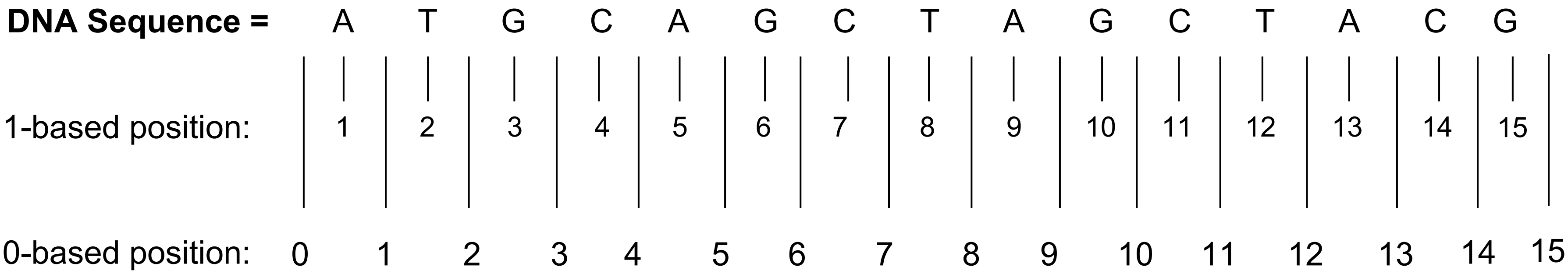
Exercise 1: specify genome coordinates for the T allele in red

- 1-based position = ?
- 0-based position = ?

Exercise 2: specify genome coordinates for the ATCG sequence in blue

- 1-based position = ?
- 0-based position = ?

Add example R and python code to go through this



DNA Sequence = A T G C A G C T A G C T A C G

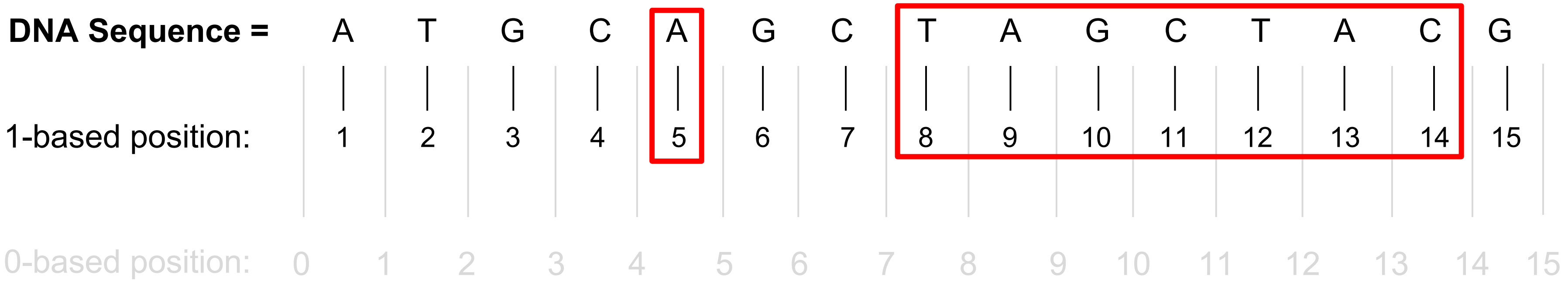
1-based position: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

0-based position: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

5 minute exercise: using R (google “substr”) and python, answer the following questions where DNA_seq = ATGCAGCTAGCTAGC:

- Identify the 5th nucleotide in the sequence
- Identify the sequence of the 8-14th nucleotides

R’s 1-index system is similar to 1-based coordinates

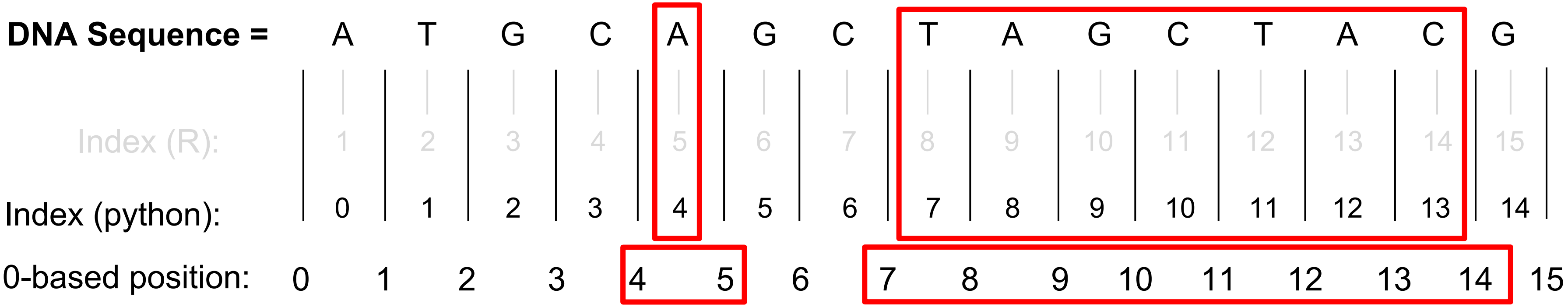


DNA Sequence = A T G C A G C T A G C T A C G

1-based position: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

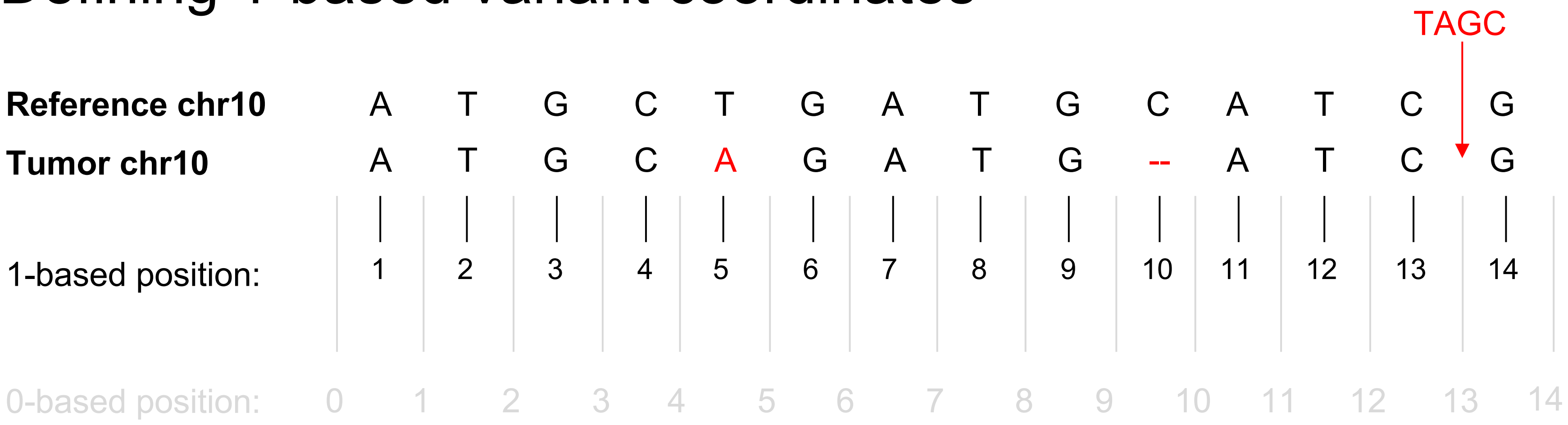
0-based position: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Python's 0-index system is analogous to 0-base coordinates



DNA Sequence = A T G C A G C T A G C T A C G
Index (R): 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Index (python): 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0-based position: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Defining 1-based variant coordinates



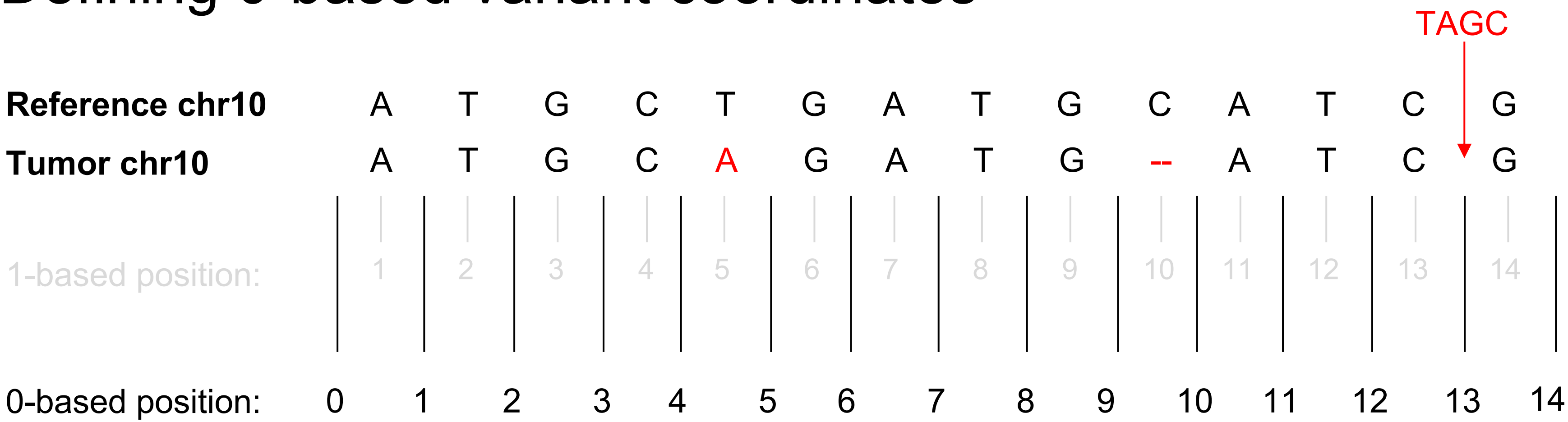
TAGC

Reference chr10	A	T	G	C	T	G	A	T	G	C	A	T	C	G
Tumor chr10	A	T	G	C	A	G	A	T	G	--	A	T	C	G
1-based position:	1	2	3	4	5	6	7	8	9	10	11	12	13	14

0-based position: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

Variant	Genomic Coordinate	Ref>Alt	Variant Coordinate	0 or 1-based
Single nucleotide variant				1 based
Deletion (C deleted)				1 based
Insertion (TAGC inserted)				1 based

Defining 0-based variant coordinates


TAGC
Reference chr10 A T G C T G A T G C A T C G
Tumor chr10 A T G C A G A T G -- A T C G
1-based position: 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0-based position: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

Variant	Genomic Coordinate	Ref>Alt	Variant Coordinate	0 or 1-based
Single nucleotide variant				0 based
Deletion (C deleted)				0 based
Insertion (TAGC inserted)				0 based

Why does 0-based or 1-based matter?

- Widely used genomic file formats use different coordinate systems
- Consistent reference to nucleotides is critical for reproducible research
- Aaron will go through different file formats in the next session

0-based	**1-based**
BAM (alignments)	SAM (alignments)
BED (**start** position only)	BED (**end** position only)
IGV (the file type - *.igv)	IGV (the viewer)
	VCF (variants)
	GFF (genomic features)
	UCSC Genome Browser

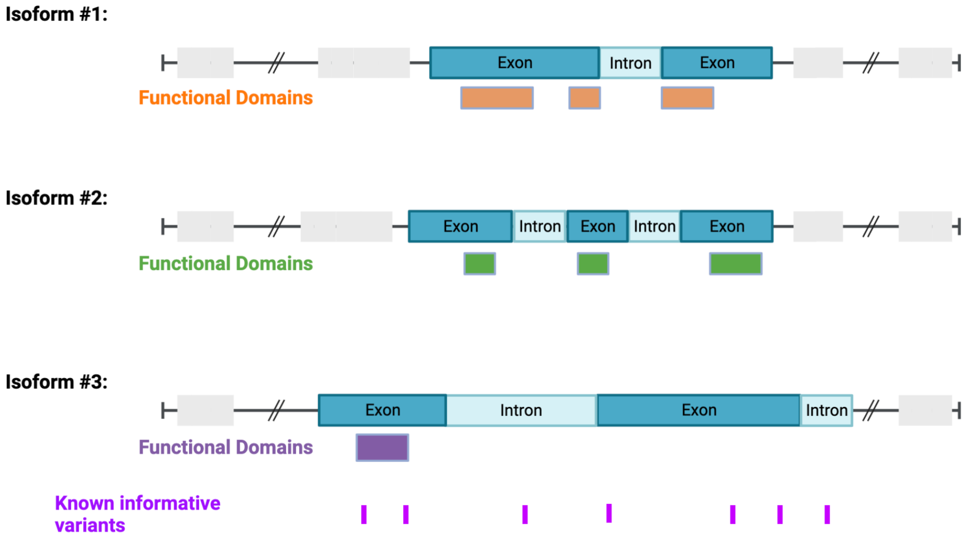
Let’s use IGV to visualize the “fun” of 0 and 1-based coordinates

- We will look at exons in *FGFR3* with the UCSC Genome Browser
  - Genome browser > tools > table browser > specify track > download
  - https://training.incf.org/lesson/how-do-i-get-coordinates-and-sequences-exons-using-ucsc-genome-browser
- Step 1: Download genomic coordinates for exons (BED file)
  - Make a new folder on your Desktop called bedtools
  - mkdir ~/Desktop/bedtools
- Step 2: Open IGV and look at FGFR3
- Step 3: Copy and paste coordinates directly from BED file into IGV
- Step 4: Load BED file into IGV

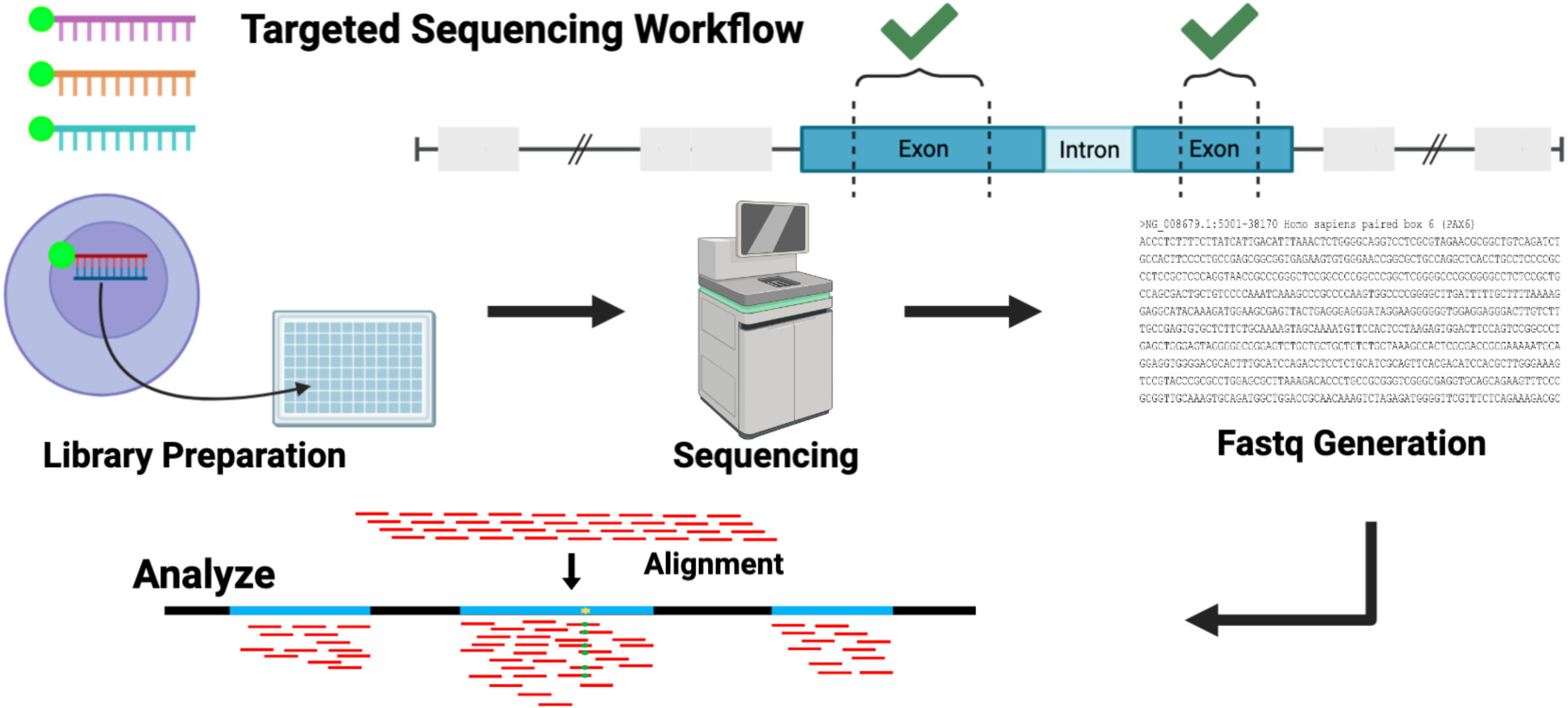
Case study of genome arithmetic: designing a custom sequencing panel

- **Overall goal: identify informative genomic intervals in coding regions for sequencing and subsequent mutation analysis**

- Things to account for:
  - Tissue-specific isoforms
  - Isoform-specific:
    - Exons
    - Functional domains
  - Sites of known mutation hotspots
- Verify intervals included in sequencing panel using IGV



Designing sequencing panel is the first step for targeted sequencing

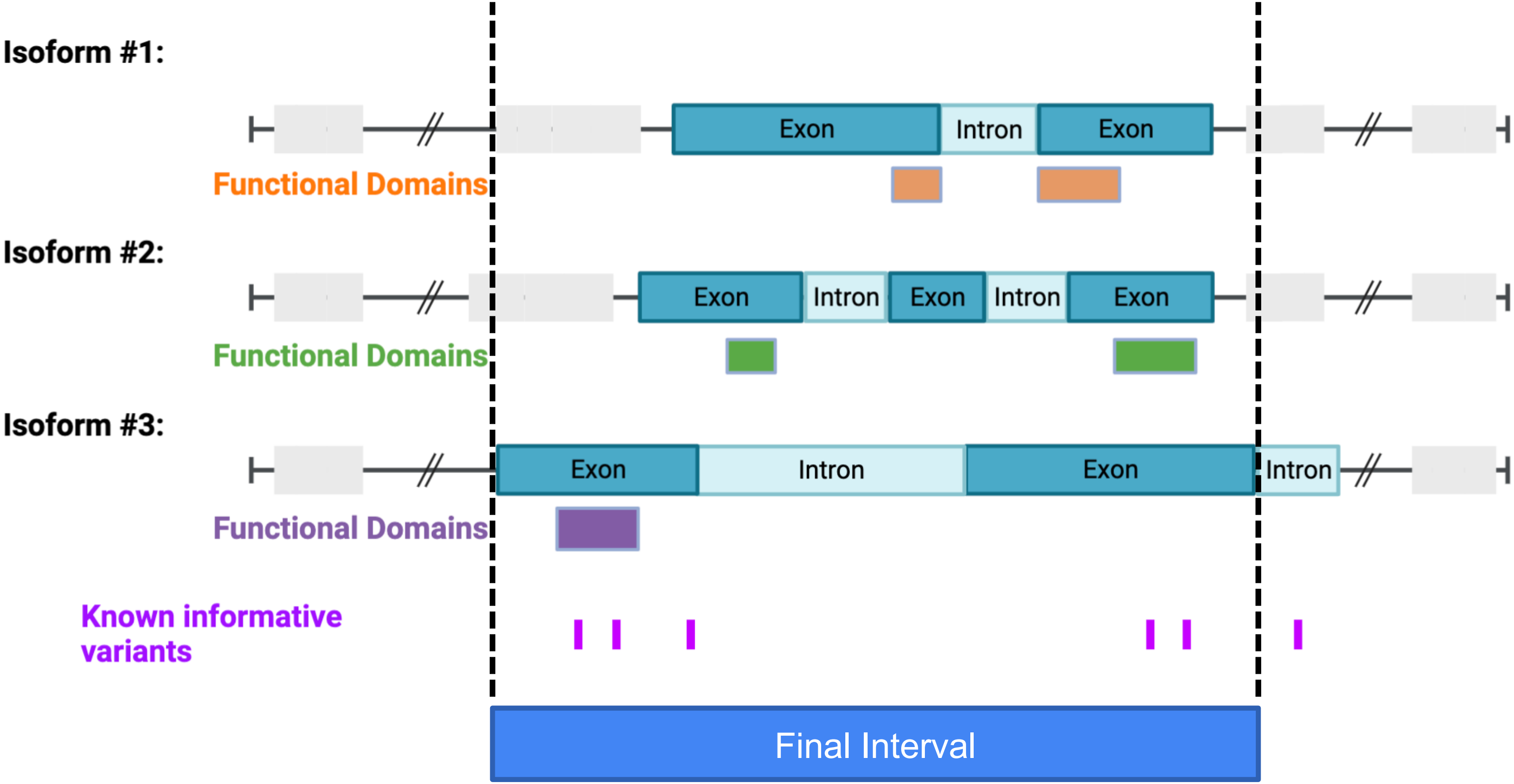


```
>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCCTCTTTCTTATCATTGACATTTAAACTCTGGGGCAGGTCCTCGCGTAGAACGCGGCTGTCAGATCT
GCCACTTCCCCTGCCGAGCGGCGGTGAGAAGTGTGGAACCGGCGCTGCCAGGCTCACCTGCCTCCCCGC
CCTCCGCTCCCAGGTAACCGCCCGGGCTCCGGCCCCGGCCCGGCTCGGGGCCCGCGGGGCCTCTCCGCTG
CCAGCGACTGCTGTCCCCAAATCAAAGCCCGCCCCAAGTGGCCCCGGGGCTTGATTTTTGCTTTTAAAAG
GAGGCATACAAAGATGGAAGCGAGTTACTGAGGGAGGGATAGGAAGGGGGGTGGAGGAGGGACTTGTCTT
TGCCGAGTGTGCTCTTCTGCAAAAGTAGCAAAATGTTCCACTCCTAAGAGTGGACTTCCAGTCCGGCCCT
GAGCTGGGAGTAGGGGGCGGGAGTCTGCTGCTGCTGTCTGCTAAAGCCACTCGCGACCGCGAAAAATGCA
GGAGGTGGGGACGCACTTTGCATCCAGACCTCCTCTGCATCGCAGTTCACGACATCCACGCTTGGGAAAG
TCCGTACCCGCGCCTGGAGCGCTTAAAGACACCCTGCCGCGGGTCGGGCGAGGTGCAGCAGAAGTTTCCC
GCGGTTGCAAAGTGCAGATGGCTGGACCGCAACAAAGTCTAGAGATGGGGTTCGTTTCTCAGAAAGACGC
```

“Verbs” in Genome Arithmetic

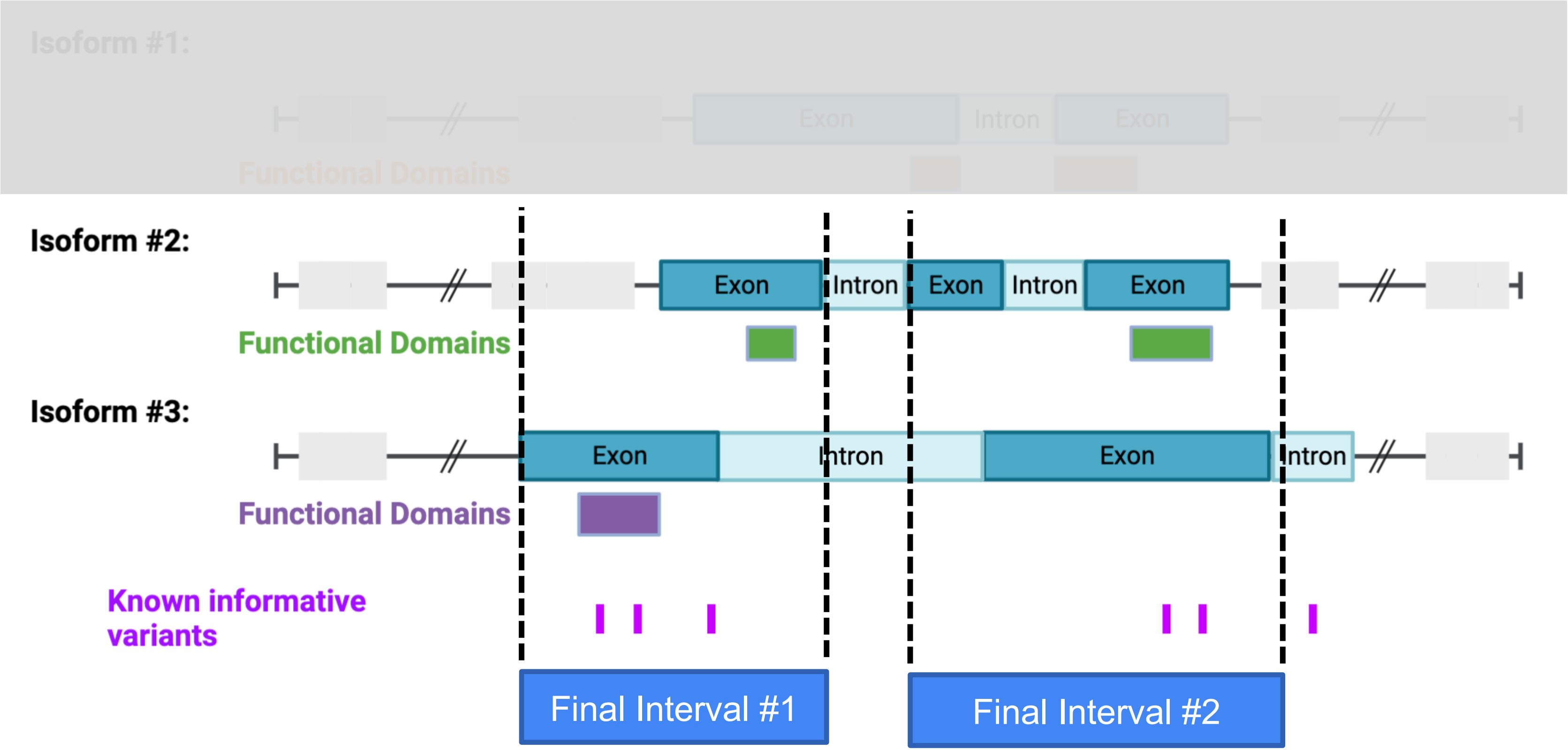
Merge: *combine* overlapping intervals

Capture all coding exons across all isoforms



Merge: *combine* overlapping intervals

Capture all coding regions across isoforms #1 and #2

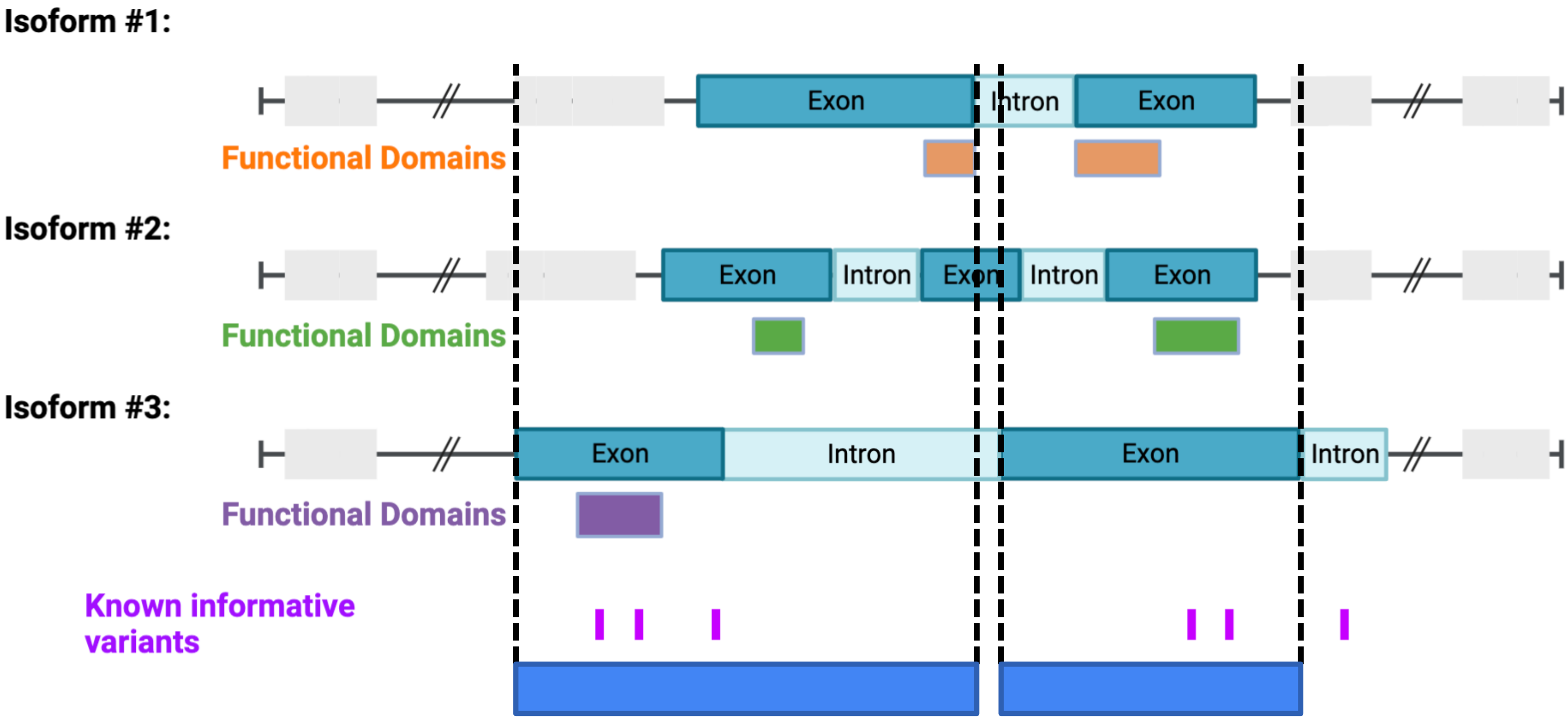


How would we do this in R/python??

- Copy and paste the R code from slack into Rstudio
- What if we could do this in one single line with three words:
- `bedtools merge [file]`

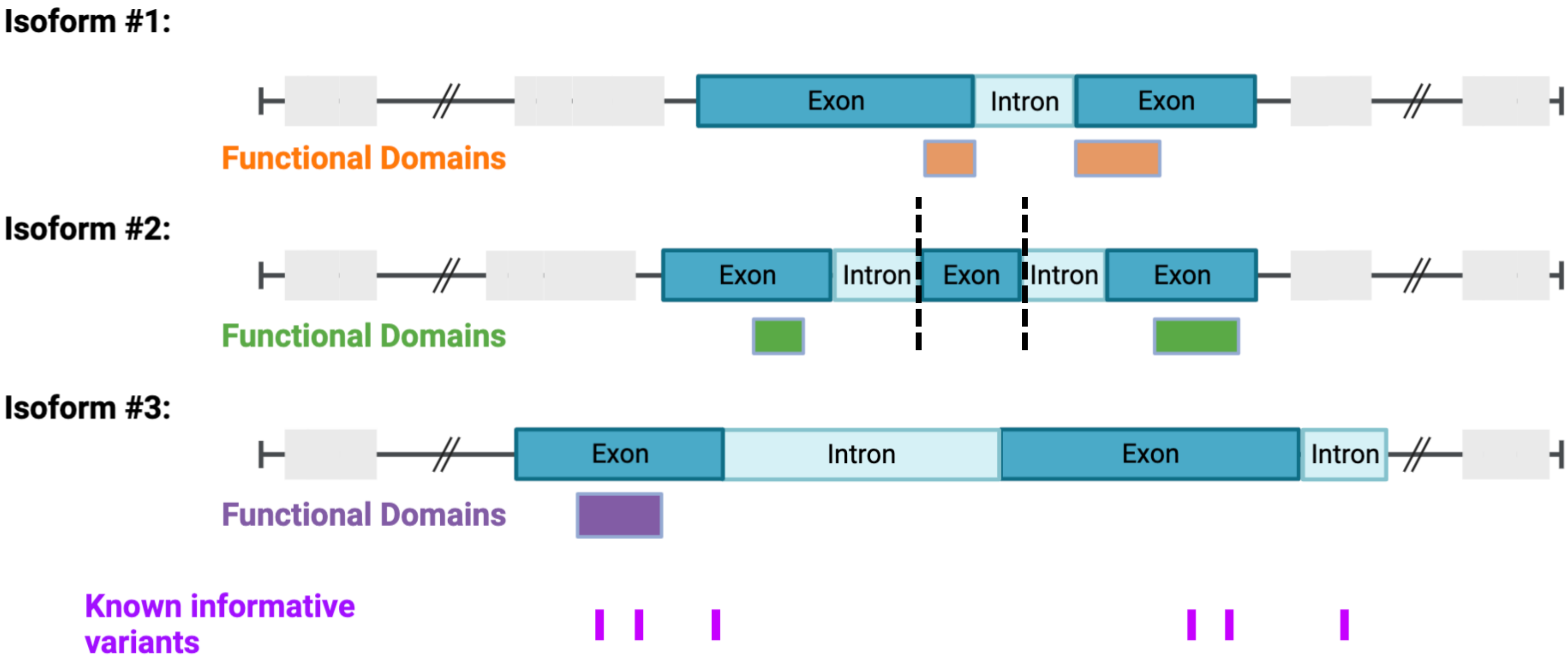
Intersection: *identify* and isolate overlapping features

Identify exons harboring informative variants (1+ variant must be in the exon) → then merge across all isoforms

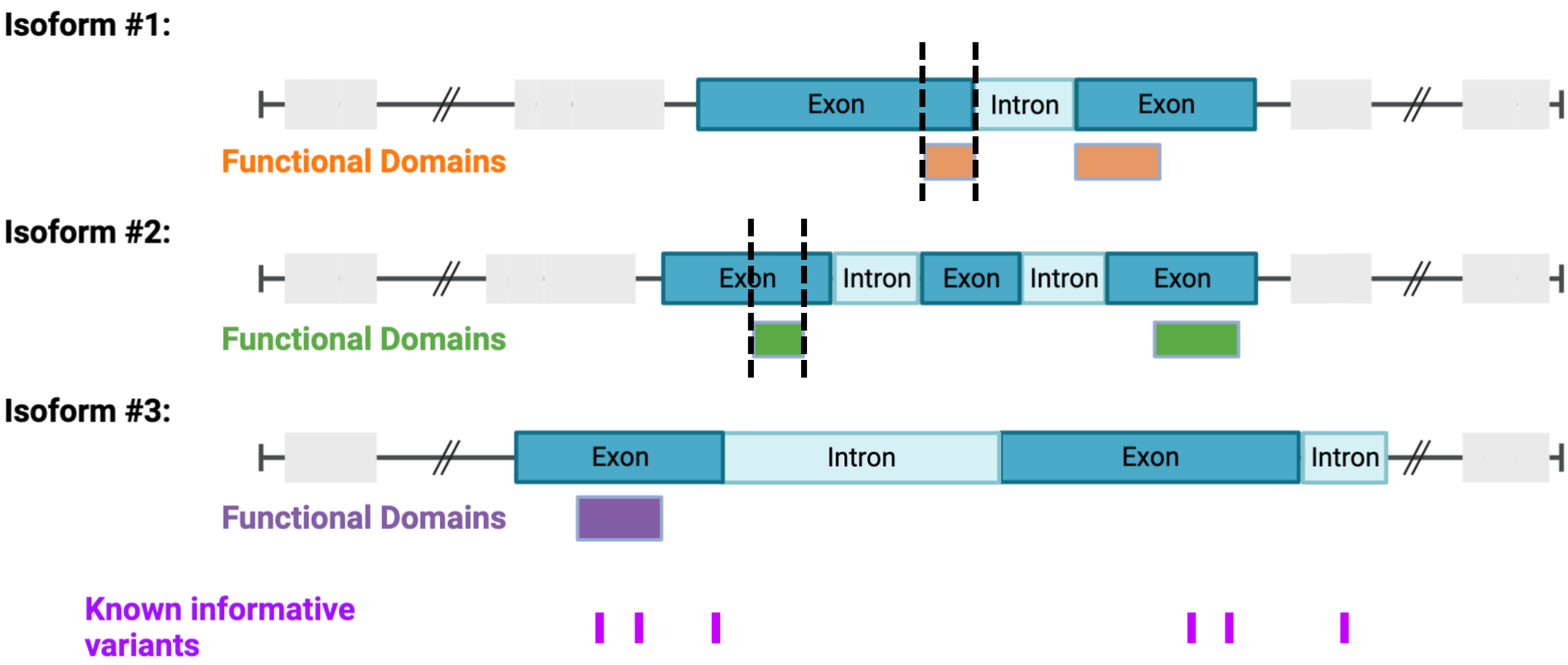


Intersection: *identify* and isolate overlapping features

Identify any exons in individual isoforms without informative variants (no variant can be in the exon at any position)

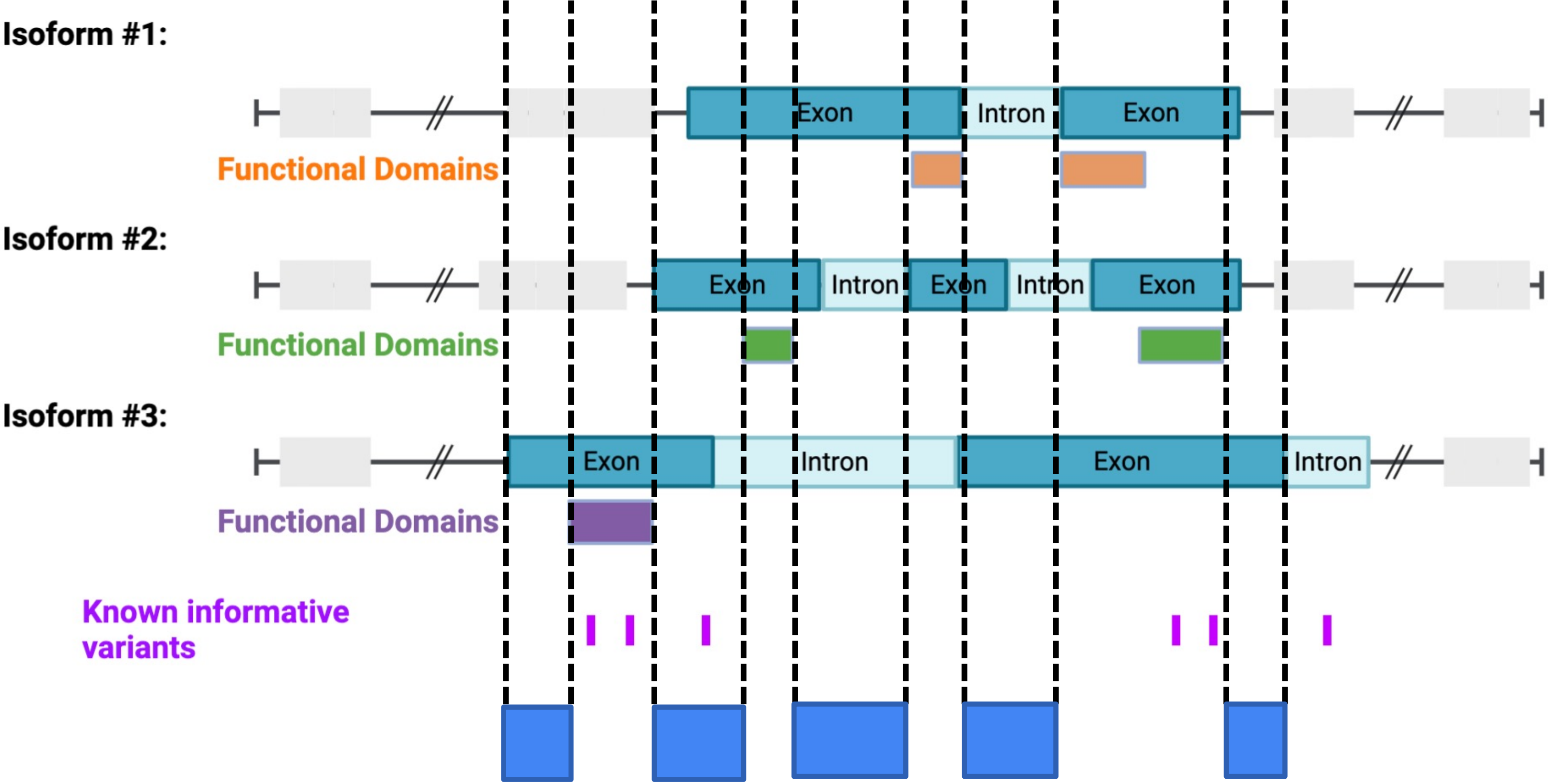


Intersection: identify portions of exons from any isoform without informative variants and overlaps with a functional domain (functional domain cannot harbor informative variant)



Complement: identify intervals *not* covered by genomic features

Get non-functional domain regions across all isoforms (if any isoform has a FD, exclude)


Isoform #1:
Exon
Intron
Exon
Functional Domains
Isoform #2:
Exon
Intron
Exon
Intron
Exon
Functional Domains
Isoform #3:
Exon
Intron
Exon
Intron
Functional Domains
Known informative variants