Gene Models & Bed format: What they represent.

Gene models are *hypotheses* about the structure of transcripts produced by a gene. Like all models, they may be correct, partly correct, or entirely wrong. Typically, we use information from EST (expressed sequence tag) projects to evaluate or create gene models. It's important to remember at all times that a gene model is only that: a model.

In this class, you will spend a lot of time working with gene models and with file formats used to represent them, mostly the "bed" (browser extensible format) format, which was developed by the UCSC Genome Bioinformatics group for displaying gene models in their on-line browser.

To understand what a gene model represents, you need to refresh your memory about how transcription, RNA splicing, and polyadenylation operate.

Most protein-coding genes in eukaryotic organisms (like humans, the research plant *Arabidopsis thaliana*, fruit flies, etc.) are transcribed into RNA by an enzyme complex called **RNA polymerase II**, which binds to the five prime end of a gene in its so-called promoter region. The promoter region typically contains binding sites for transcription factors that help the RNA polymerase complex recognize the position in the genomic DNA where it should begin transcription. Many genes have multiple places in the genomic DNA where transcription can begin, and so transcripts arising from the same gene may have different five-prime ends. Transcripts arising from the same gene that have different transcription start sites are said to come from **alternative promoters.**

Once the RNA polymerase complex binds to the five prime end of gene, it can begin building an RNA copy of the DNA sense strand via the process known as **transcription**. The ultimate product of transcription is thus called a **transcript**. During and after transcription, another large complex of proteins and non-coding RNAs called the **spliceosome** attaches to the growing RNA molecule, cuts out segments of RNA called **introns**, and joins together (splices) the flanking sequences, which are called **exons**. Not every newly synthesized transcript is processed in this way; sometimes no introns are removed at all. Genes whose products do not undergo splicing are often called **single-exon genes**. Also, splicing may remove different segments from transcripts arising from the same gene. This variability in splicing patterns is called **alternative splicing**.

In addition to splicing, RNA transcripts undergo another processing reaction called **polyadenylation**. In polyadenylation, a segment of sequence at the 3-prime end of the RNA transcript is cut off, and a polymer consisting of adenosine residues called a polyA tail is attached to the 3-prime end of the transcript. The length of polyA tail may vary a lot from transcript to transcript, and the position where it is added may also differ. Genes whose transcripts can receive a polyA tail at more than one location are said to be subject to **alternative polyadenylation** or **alternative 3-prime end processing**.

These processing reactions are believed to take place in the nucleus. Ultimately, the mature or maturing RNA transcript is exported from the nucleus into the cytoplasm, where it will be translated by **ribosomes** into proteins, chains of amino acids that perform work in the cell (such as enzymes) or that provide form and structure (like actin in the cytoskeleton).

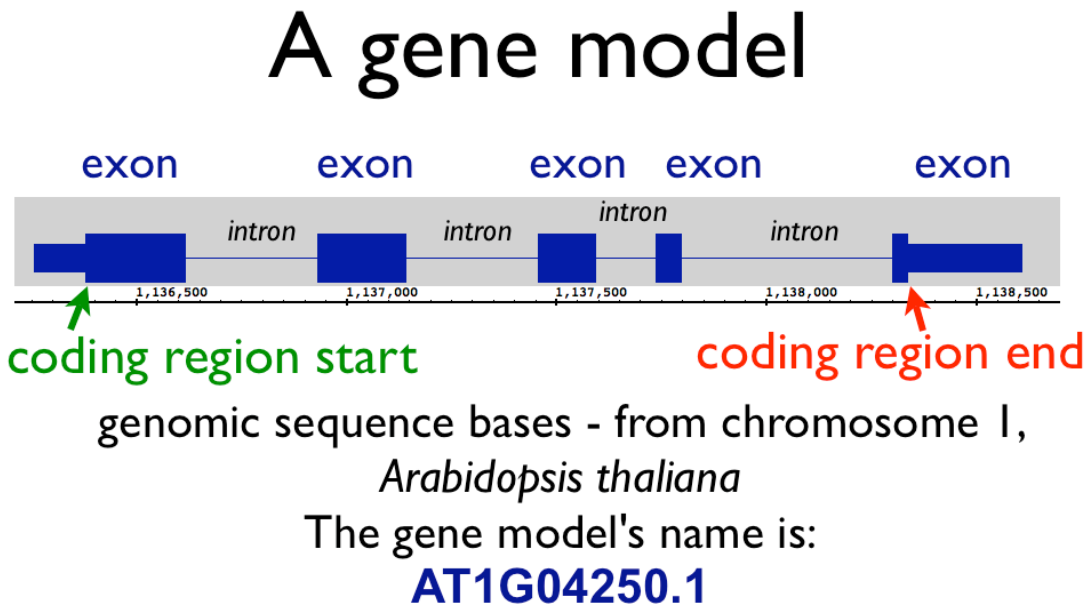
The continuous sequence of bases in an RNA that encode a protein is called a **coding region**, and the coding typically starts with an AUG codon and terminates with one of three possible stop codons. The segments of sequence that comprise a coding region are called **CDSs** and they generally occupy the same sequences as the exons, apart from the regions five and three prime of the start and stop codons, respectively.

Most RNAs code for one protein sequence, but there are some interesting exceptions in which one mature mRNA may contain more than one translated open reading frame. The three bases where the ribosome initiates translation are called a **start codon** and the triplet of bases immediately following the last translated codon are called the **stop codon**. The start codon encodes the amino acid methionine, typically, and the stop codon doesn't code for any amino acid.

A gene model thus consists of a collection of introns and exons and their locations in the genomic sequence, as well as the location of the translated region or region. Thus, a gene model implies a theory about where the RNA polymerase started transcription, as well as the location of the polyadenylation site and the starts and stops of translation.

Usually, we draw gene models as showing the location of introns and exons relative to the genomic sequence, as if we are mapping the RNA copy back onto the genomic DNA itself.

Consider the following gene model diagram, which comes from a genome visualization program called the Integrated Genome Browser. (To run it yourself, visit igb.bioviz.org.)



This diagram can also be represented in simple text using the "bed" format, a format developed at UCSC for displaying gene models in the UCSC Genome Browser. Scientists who want to use the UCSC Genome Browser to display gene models or other annotations can create "bed" format files and upload these to the Genome Browser site, which will then display them alongside the other annotations that are part of the UCSC database system. This is why fields in "bed" format files refer to things like scores and color schemes, which the UCSC Genome Browser can display.

The data file you'll work with for your second assignment on using Unix tools in bioinformatics uses the "bed" format to represent gene models from *Arabidopsis thaliana*, including the gene model shown above. To find the line of text representing the gene model shown above, after downloading the file for Assignment Two, use the Unix command `grep` like so:

```
$ zcat TAIR9_mRNA.bed.gz | grep AT1G04250.1
```

The result is the following line of text:

```
chr1 1136257   1138613   AT1G04250.1    0    +    1136381   1138340   0
     363,212,139,62,311, 0,676,1202,1482,2045,
```

This line of text contains several fields of data that are separated by tab characters. The first three fields give the genomic location for the gene model – the name of the sequence where it resides, and its start (chromStart) and end (chromEnd) positions on that chromosome. The next field gives the gene model's name. The next field indicates a score associated with the gene model; it can be any value between 0 and 1000, inclusive. In this case, the gene model has no score and so the value in this field is 0. The next field defines the gene model's strand. In this case, the gene model is located on the plus (also called top) strand of the DNA sequence named chr1. The strand of the gene model indicates which strand of chr1 is the sense strand for the gene. The sense strand is what ultimately is copied into mRNA during transcription. The sense strand of the gene is identical in sequence to the processed mRNA transcript, except that all T residues are replaced with U's in mRNA and the mRNA lacks introns. The next field (field 9) should contain an RGB value (e.g., 255,0,0) or 0. When non-zero, it indicates a color a browser should use to represent the model when presented visually. No color is specified in the example above, and so here the ninth contains the value "0." The next field indicates the number of exon blocks (5 in this case) and the two fields after that contain comma-separated lists of exon sizes and start postions, respectively. Note that the exon start positions are relative to the position given in the second field. To get the start position of an exon, you would have to add the number in the last field to the number the second field.

In most situations, you can safely ignore the score and color fields – fields five and nine. Also, the sixth field is redundant; if you know the start positions and sizes of all the exons, you can also count them.

To review, examine the following screen capture from the page at: http://genome.ucsc.edu/FAQ/FAQformat#format1

BED format provides a flexible way to define the data lines that are displayed in an annotation track. BED lines have three required fields and nine additional optional fields. The number of fields per line must be consistent throughout any single set of data in an annotation track. The order of the optional fields is binding: lower-numbered fields must always be populated if higher-numbered fields are used.

The first three required BED fields are:

1. **chrom** - The name of the chromosome (e.g. chr3, chrY, chr2_random) or scaffold (e.g. scaffold10671).
2. **chromStart** - The starting position of the feature in the chromosome or scaffold. The first base in a chromosome is numbered 0.
3. **chromEnd** - The ending position of the feature in the chromosome or scaffold. The *chromEnd* base is not included in the display of the feature. For example, the first 100 bases of a chromosome are defined as *chromStart=0, chromEnd=100*, and span the bases numbered 0-99.

The 9 additional optional BED fields are:

4. **name** - Defines the name of the BED line. This label is displayed to the left of the BED line in the Genome Browser window when the track is open to full display mode or directly to the left of the item in pack mode.
5. **score** - A score between 0 and 1000. If the track line *useScore* attribute is set to 1 for this annotation data set, the *score* value will determine the level of gray in which this feature is displayed (higher numbers = darker gray). This table shows the Genome Browser's translation of BED score values into shades of gray:

shade								
score in range	≤ 166	167-277	278-388	389-499	500-611	612-722	723-833	834-944	≥ 945

6. **strand** - Defines the strand - either '+' or '-'.
7. **thickStart** - The starting position at which the feature is drawn thickly (for example, the start codon in gene displays).
8. **thickEnd** - The ending position at which the feature is drawn thickly (for example, the stop codon in gene displays).
9. **itemRgb** - An RGB value of the form R,G,B (e.g. 255,0,0). If the track line *itemRgb* attribute is set to "On", this RBG value will determine the display color of the data contained in this BED line. NOTE: It is recommended that a simple color scheme (eight colors or less) be used with this attribute to avoid overwhelming the color resources of the Genome Browser and your Internet browser.
10. **blockCount** - The number of blocks (exons) in the BED line.
11. **blockSizes** - A comma-separated list of the block sizes. The number of items in this list should correspond to *blockCount*.
12. **blockStarts** - A comma-separated list of block starts. All of the *blockStart* positions should be calculated relative to *chromStart*. The number of items in this list should correspond to *blockCount*.

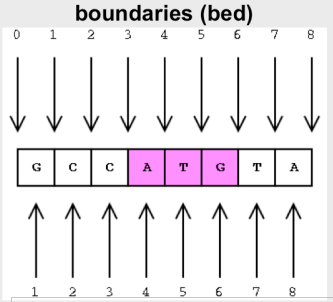
Example:
Here's an example of an annotation track that uses a complete BED definition:

```
track name=pairedReads description="Clone Paired Reads" useScore=1
chr22 1000 5000 cloneA 960 + 1000 5000 0 2 567,488, 0,3512
chr22 2000 6000 cloneB 900 - 2000 6000 0 2 433,399, 0,3601
```

The bed format indicates positions using **interbase** notation, which is different from how programs like blast and many other sequence analysis programs indicate locations within a sequence. Interbase is a way of numbering base positions, as illustrated in the figure below. Note that in interbase, we count the boundaries between bases, not the bases themselves.

"bed" uses interbase coordinates

interbase **numbers**
boundaries (bed)



one-based **numbers bases (used by blast, GFF)**

the pink feature (ATG) in "bed" coordinates:

chromStart: 3
chromEnd: 6
blockCount: 1
blockSizes: 3
blockStarts: 3,

note**: end = size + start**

`http://gmod.org/wiki/Introduction_to_Chado#Interbase_Coordinates`