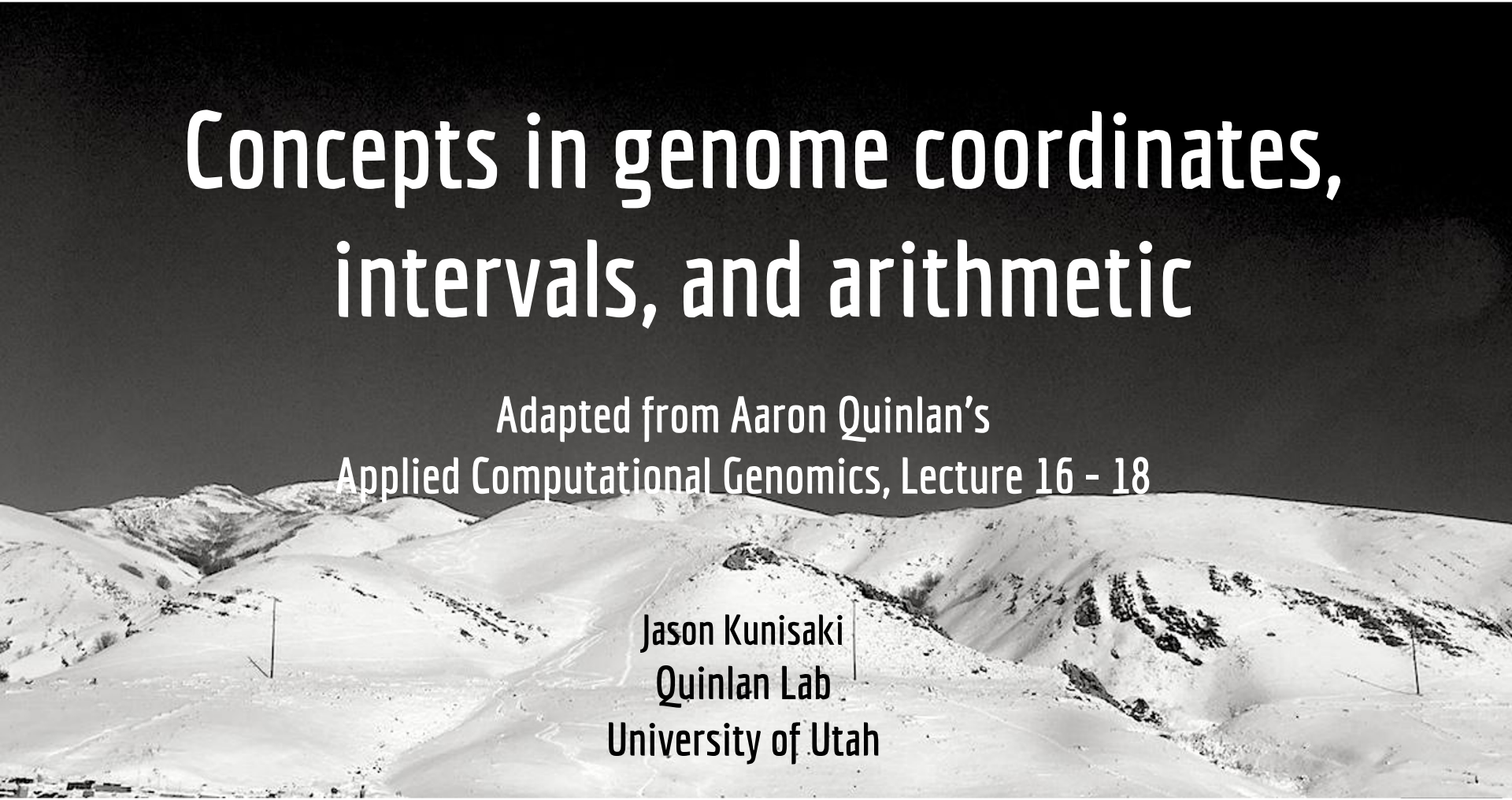
Concepts in genome coordinates, intervals, and arithmetic

Adapted from Aaron Quinlan's
Applied Computational Genomics, Lecture 16 - 18



Jason Kunisaki
Quinlan Lab
University of Utah

Learning objectives

- Understand how “omic” analyses relies on features/annotations with specific genomic intervals or coordinates
- Grasp concepts in genome arithmetic
- Perform “basic” genome arithmetic calculations with the command line tool, **awk**
- Use **bedtools** to perform complex genome arithmetic analyses
- Understand the general utility of **bedtools** to answer many biological questions and problems

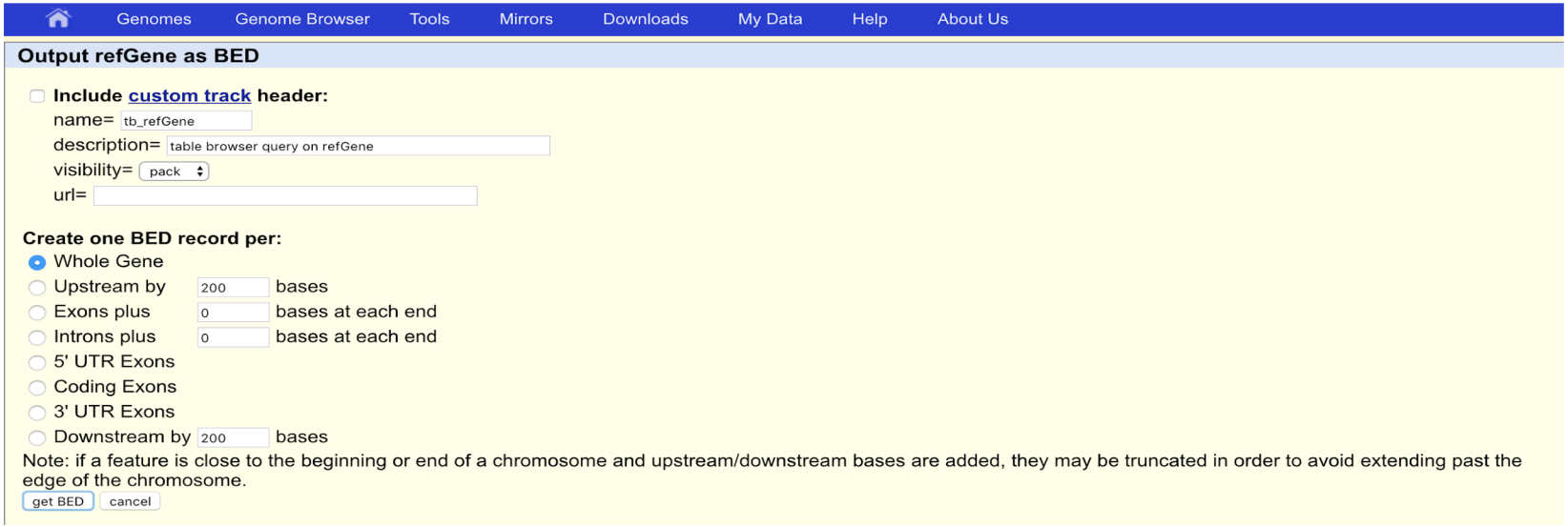
Quick overview of the UCSC Genome Browser

The UCSC Table Browser

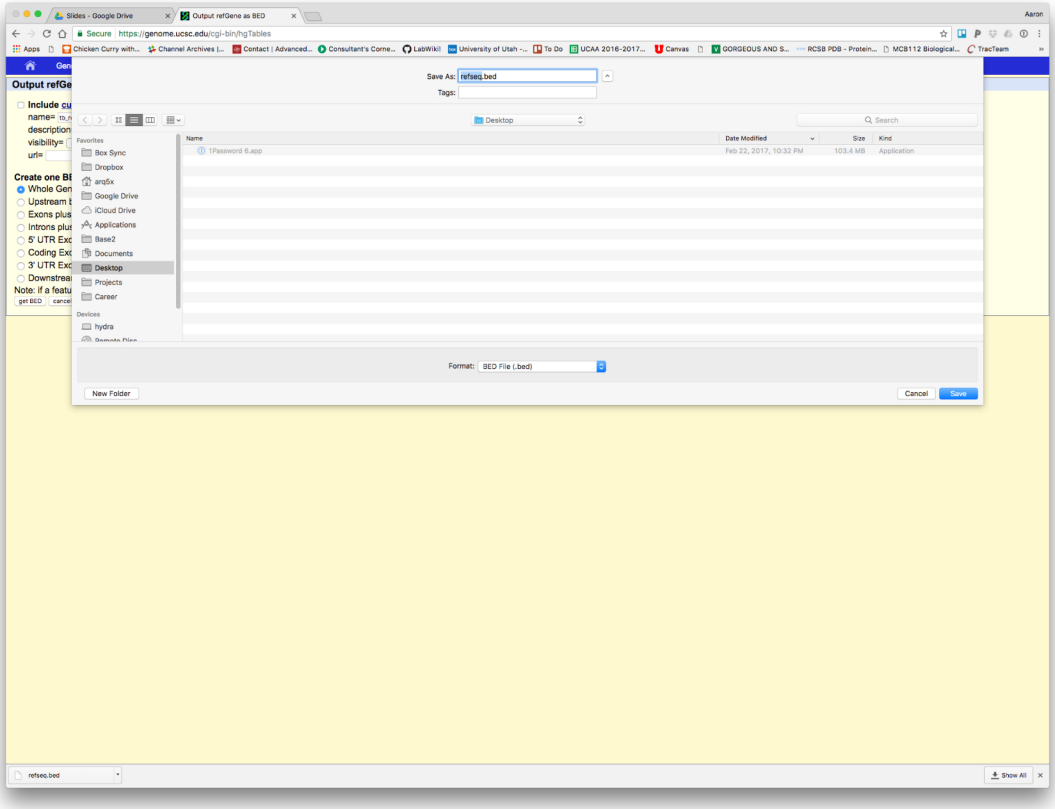
Let's save the annotations as a file called "refseq.bed"

Then click "get output"

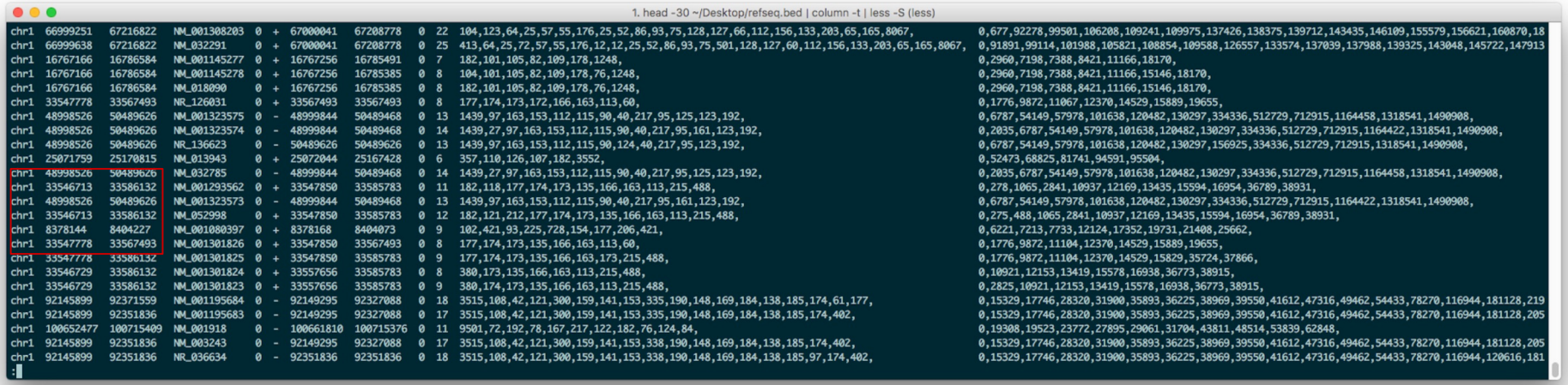
We want the “Whole Gene” in this case. Other options…



Choose the location to which to download refseq.bed



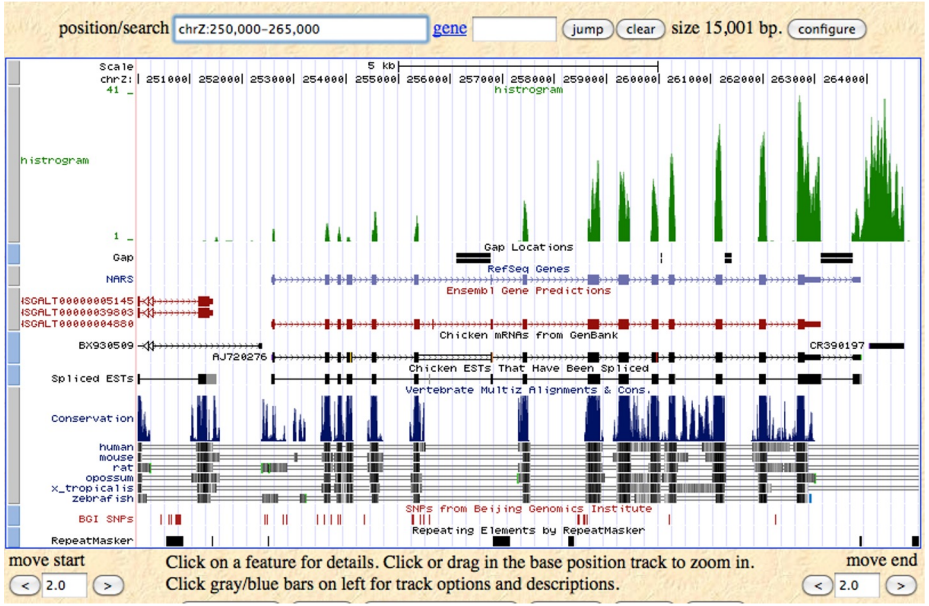
What is in the file?



Quick demo on downloading this file from the start

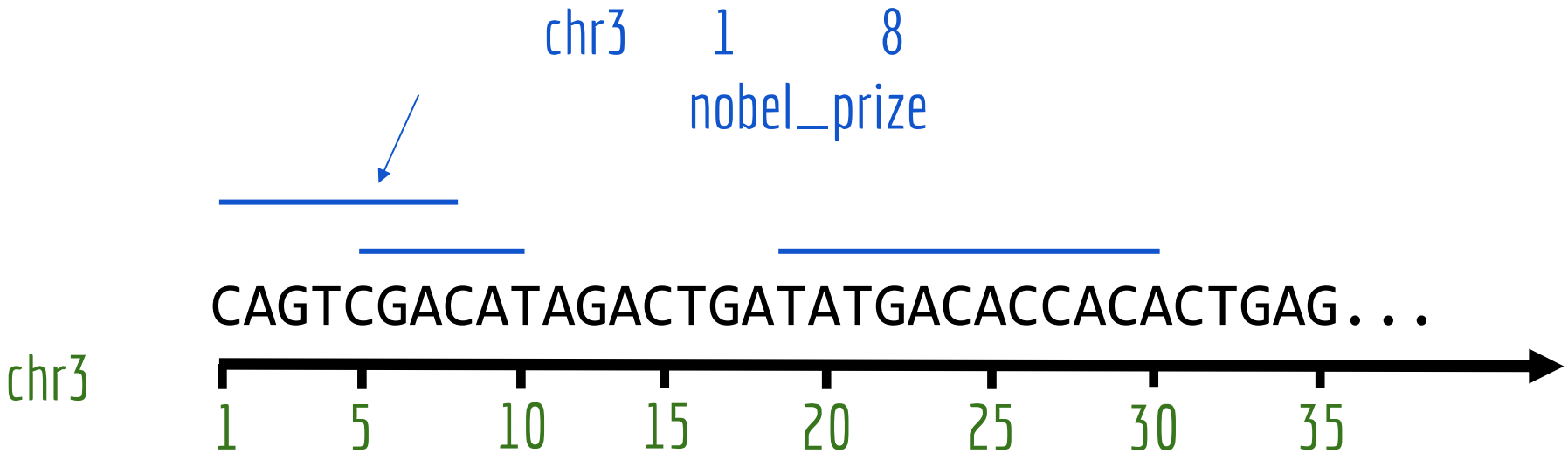
Performing “omic” analyses requires an understanding of genome coordinates

- Genes: exons, introns, UTRs, promoters (BED, GFF, GTF
- Genetic variation (VCF)
- Transcription factor binding sites (BED, BEDGRAPH)
- CpG islands (BED)
- Chromatin annotations (BED)
- Gene expression data (WIG, BIGWIG, BEDGRAPH)
- **Your own observations: put them in context**



What is a coordinate system in genomics?

The reference genome as a coordinate system



Great, genome formats using a common genome coordinate system! My life is going to be easy.

No.

BED (browser extensible data) format

BED format

Index ▷

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.

If your data set is BED-like, but it is very large (over 50MB) and you would like to keep it on your own server, you should use the bigBed data format.

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.

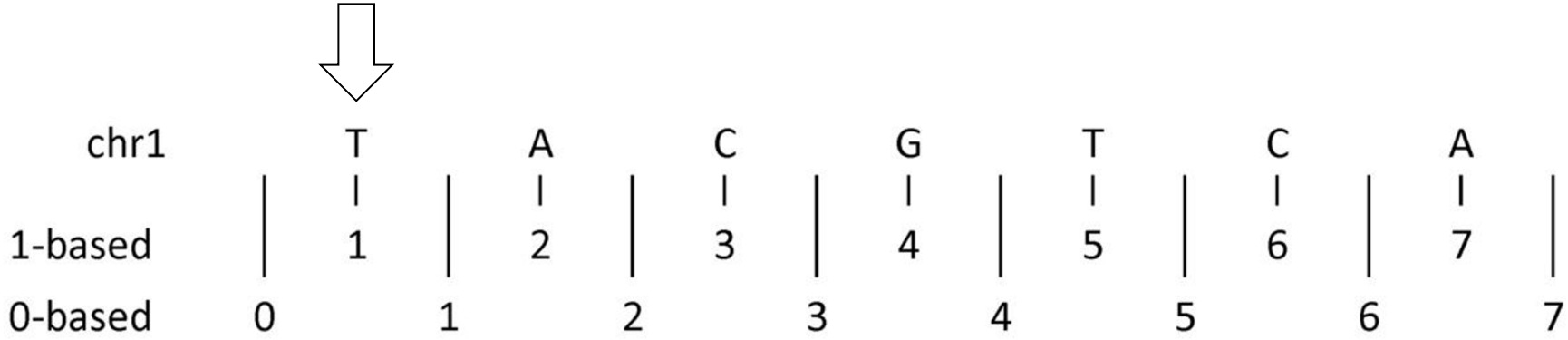
The 9 additional optional BED fields are:

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). When there is no thick part, thickStart and thickEnd are usually set to the chromStart position.
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*.

BED files use 0 based, half open intervals. Say what?



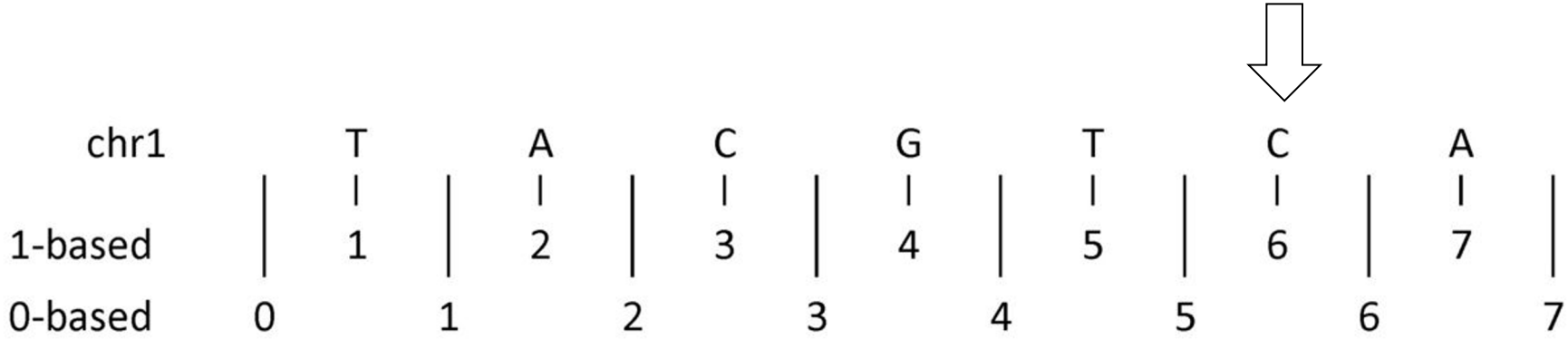
What interval defines the 1st nucleotide of chromosome 1?

1-based: chr1, start = 1, end = 1

0-based: chr1, start = 0, end = 1

https://www.biostars.org/p/84686/

BED files use 0 based, half open intervals. Say what?



What interval defines the 1st nucleotide of chromosome 1?

1-based: chr1, start = 1, end = 1

0-based: chr1, start = 0, end = 1

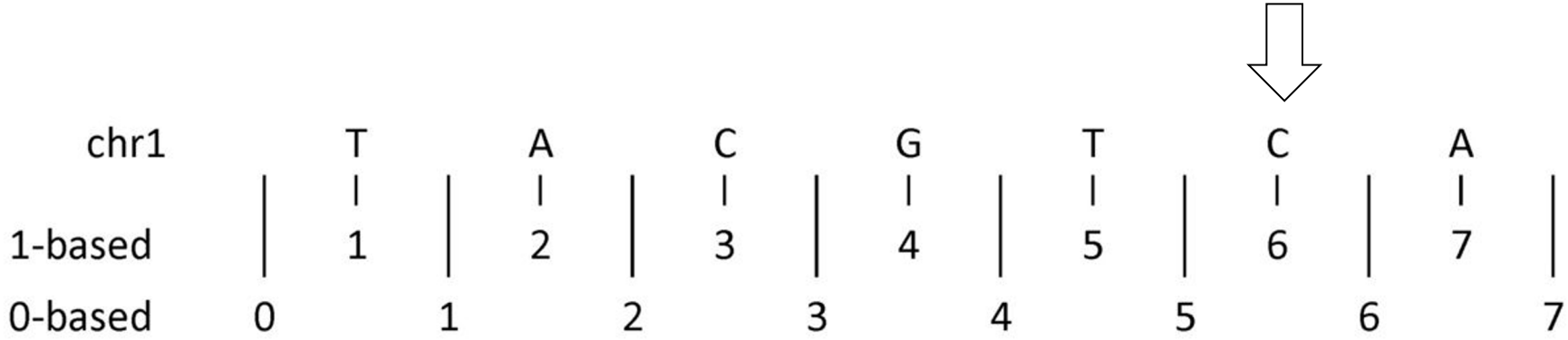
What interval defines the 6th nucleotide of chromosome 1?

1-based: chr1, start = ?, end = ?

0-based: chr1, start = ?, end = ?

https://www.biostars.org/p/84686/

BED files use 0 based, half open intervals. Say what?



What interval defines the 1st nucleotide of chromosome 1?

1-based: chr1, start = 1, end = 1

0-based: chr1, start = 0, end = 1

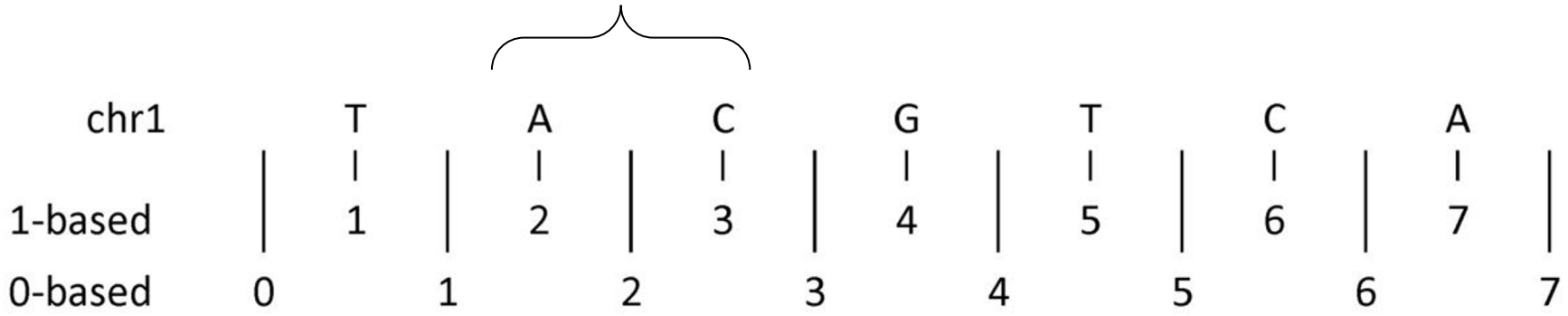
What interval defines the 6th nucleotide of chromosome 1?

1-based: chr1, start = 6, end = 6

0-based: chr1, start = 5, end = 6

https://www.biostars.org/p/84686/

BED files use 0 based, half open intervals. Say what?

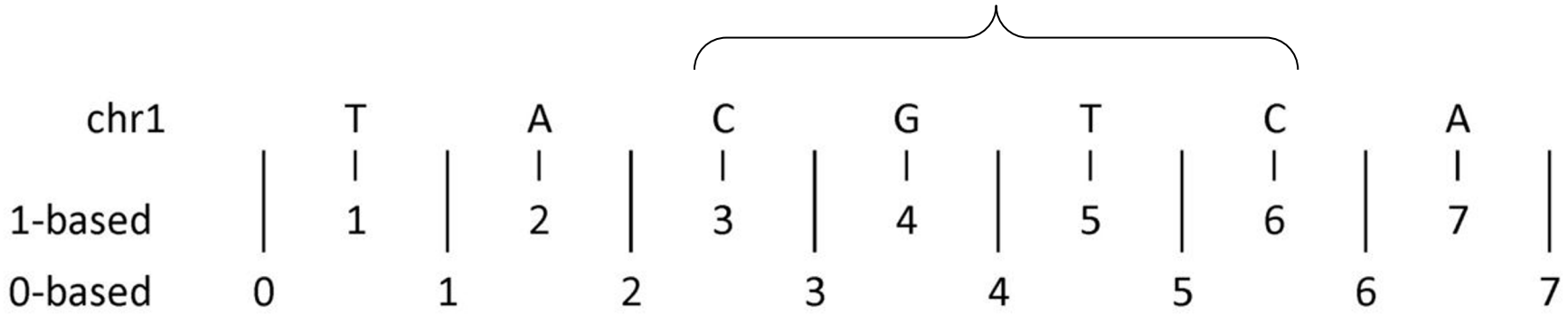


What interval defines the 2nd and 3rd nucleotides of chromosome 1?

- 1-based? chr1, start = 2, end = 3
- 0-based? chr1, start = 1, end = 3

https://www.biostars.org/p/84686/

BED files use 0 based, half open intervals. Say what?

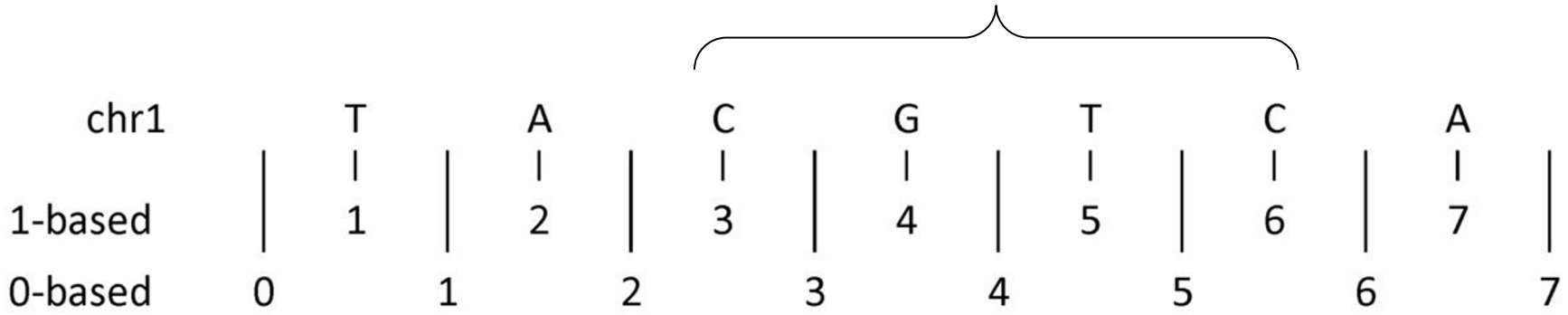


What interval defines the 3rd and 6th nucleotides of chromosome 1?

- 1-based? chr1, start = ?, end = ?
- 0-based? chr1, start = ?, end = ?

https://www.biostars.org/p/84686/

BED files use 0 based, half open intervals. Say what?

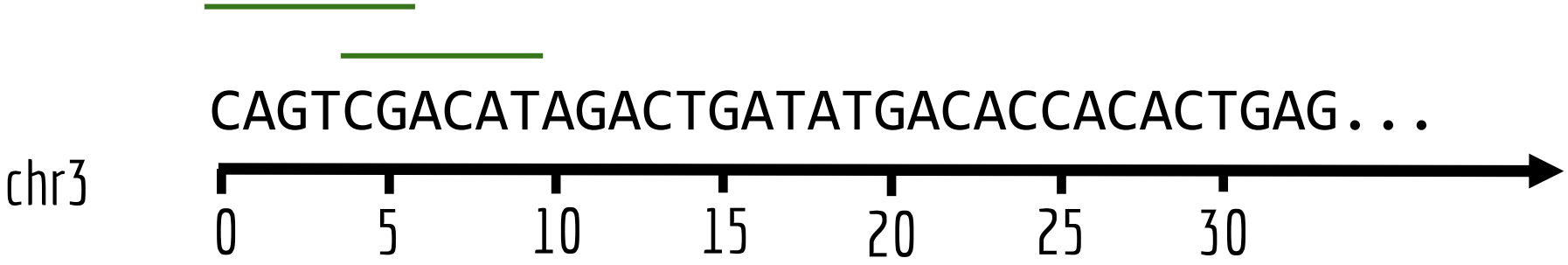


What interval defines the 3rd and 6th nucleotides of chromosome 1?

- 1-based? chr1, start = 3, end = 6
- 0-based? chr1, start = 2, end = 6
- **0-based is convenient when measuring length of an interval**

https://www.biostars.org/p/84686/

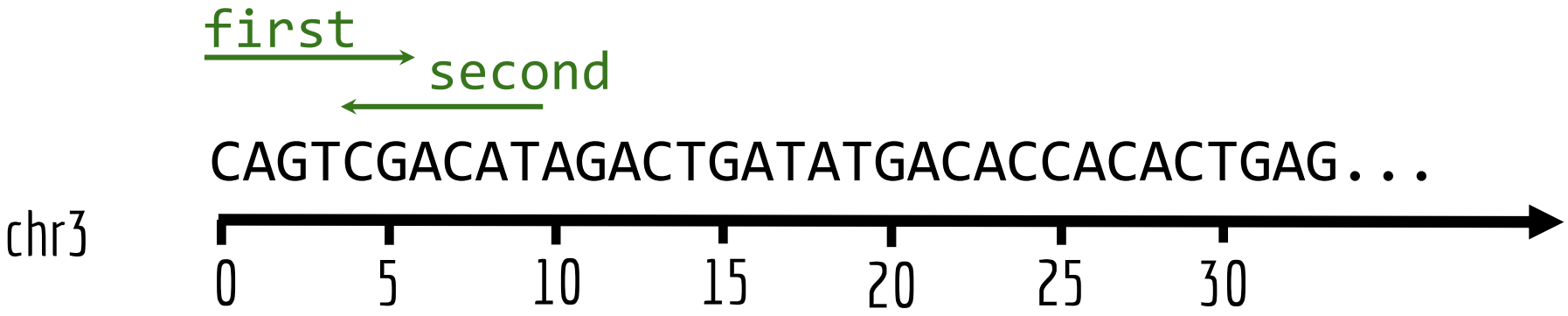
BED **annotations** also support "names" or "labels" (4th column)




```
chr3    0    7    exon_gene1
chr3    4    10   exon_gene2
```

Annotations

Start and end coordinates are agnostic to direction of the feature




```
chr3    0     7    +_strand_gene
chr3    4     10   -_strand_gene
```

GFF format

GFF format Index ▷

GFF (General Feature Format) lines are based on the Sanger GFF2 specification. GFF lines have nine required fields that *must* be tab-separated. If the fields are separated by spaces instead of tabs, the track will not display correctly. For more information on GFF format, refer to Sanger's GFF page.

Note that there is also a GFF3 specification that is not currently supported by the Browser. All GFF tracks must be formatted according to Sanger's GFF2 specification.

If you would like to obtain browser data in GFF (GTF) format, please refer to Genes in gtf or gff format on the Wiki.

Here is a brief description of the GFF fields:

1. **seqname** - The name of the sequence. Must be a chromosome or scaffold.
2. **source** - The program that generated this feature.
3. **feature** - The name of this type of feature. Some examples of standard feature types are "CDS", "start_codon", "stop_codon", and "exon".
4. **start** - The starting position of the feature in the sequence. The first base is numbered 1.
5. **end** - The ending position of the feature (inclusive).
6. **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). If there is no score value, enter ".".
7. **strand** - Valid entries include '+', '-', or '.' (for don't know/don't care).
8. **frame** - If the feature is a coding exon, *frame* should be a number between 0-2 that represents the reading frame of the first base. If the feature is not a coding exon, the value should be '.'.
9. **group** - All lines with the same group are linked together into a single item.

Example:
Here's an example of a GFF-based track. This example can be pasted into the browser without editing. NOTE: Paste operations on some operating systems will replace tabs with spaces, which will result in an error when the GFF track is uploaded. You can circumvent this problem by pasting the URL of the above example (http://genome.ucsc.edu/goldenPath/help/regulatory.txt) instead of the text itself into the custom annotation track text box. If you encounter an error when loading a GFF track, check that the data lines contain tabs rather than spaces.

```
browser position chr22:10000000-10025000
browser hide all
track name=regulatory description="TeleGene(tm) Regulatory Regions"
visibility=2
chr22  TeleGene enhancer  10000000  10001000  500 +  .  touch1
chr22  TeleGene promoter  10010000  10010100  900 +  .  touch1
chr22  TeleGene promoter  10020000  10025000  800 -  .  touch2
```

Click here to display this track in the Genome Browser.

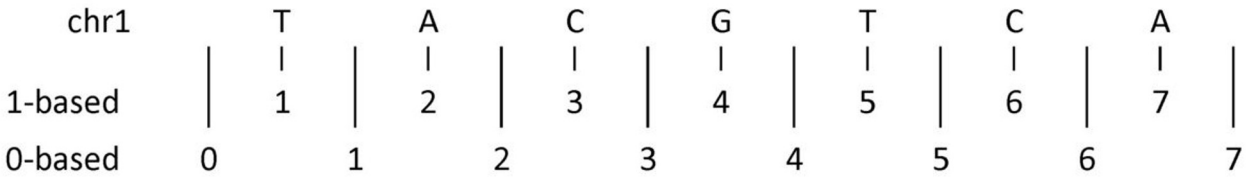
```
chr22  TeleGene enhancer  10000000  10001000  500 +  .  touch1
chr22  TeleGene promoter  10010000  10010100  900 +  .  touch1
chr22  TeleGene promoter  10020000  10025000  800 -  .  touch2
```

Note that the start and end coordinates are in different columns versus BED format

Formats use different coordinate systems. Because science.

BED: 0-based, half-open
GFF: 1-based, closed
SAM: 1-based, closed
BAM: 0-based, half-open.
VCF: 1-based, closed

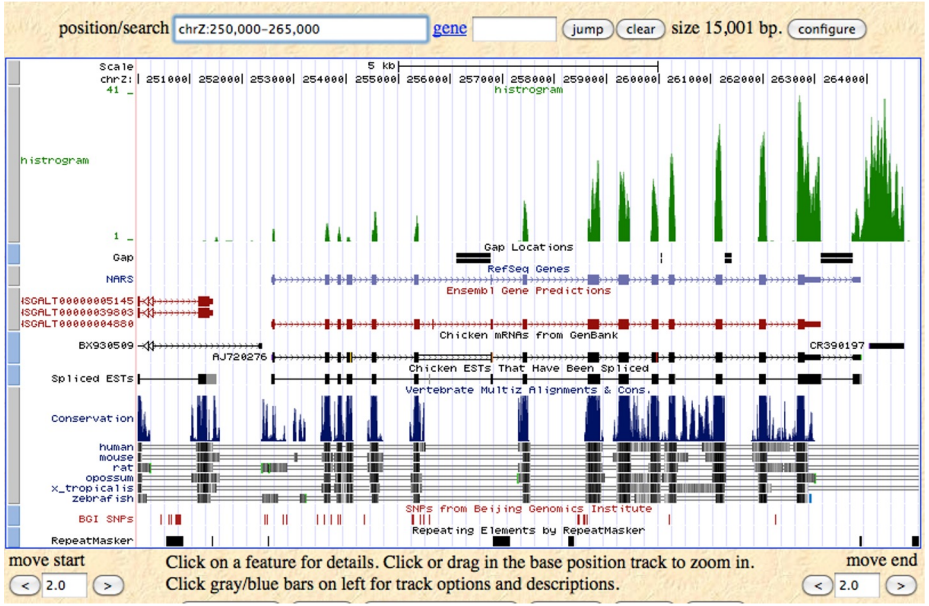
…



- **inconsistent chromosome labels. (chr1 vs 1 vs c1)**
- **different sorting criteria. (lexicographic vs natural)**
- mixed UNIX/Windows newlines. (transfer files across programs/systems)
- file violates spec with vigor.
- **file is gzipp'ed, not bgzipp'ed.**
- annotations use diff. genome builds.
- tool only works for one format.
- tool is hard-coded for specific build.
- **tool requires act of gods to compile.**

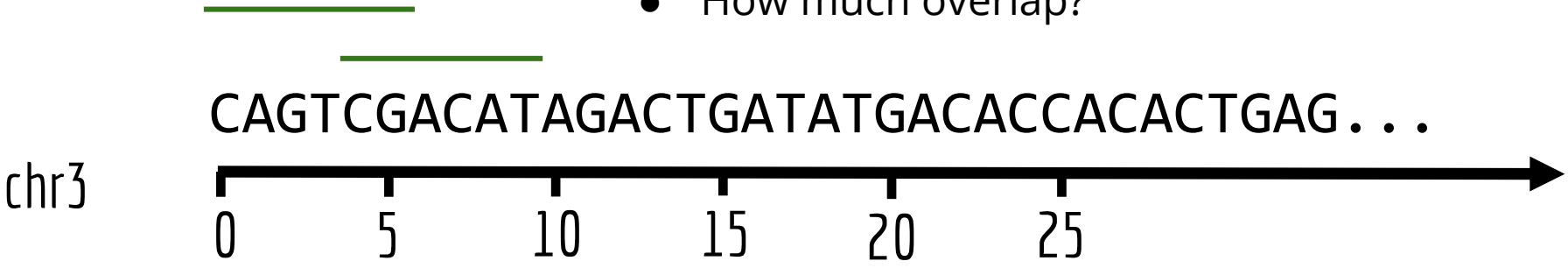
Performing “omic” analyses requires an understanding of genome coordinates

- Genes: exons, introns, UTRs, promoters (BED, GFF, GTF)
- Genetic variation (VCF)
- Transcription factor binding sites (BED, BEDGRAPH)
- CpG islands (BED)
- Chromatin annotations (BED)
- Gene expression data (WIG, BIGWIG, BEDGRAPH)
- **Genome arithmetic**: the method of comparing, contrast and gain insight among multiple genome interval files



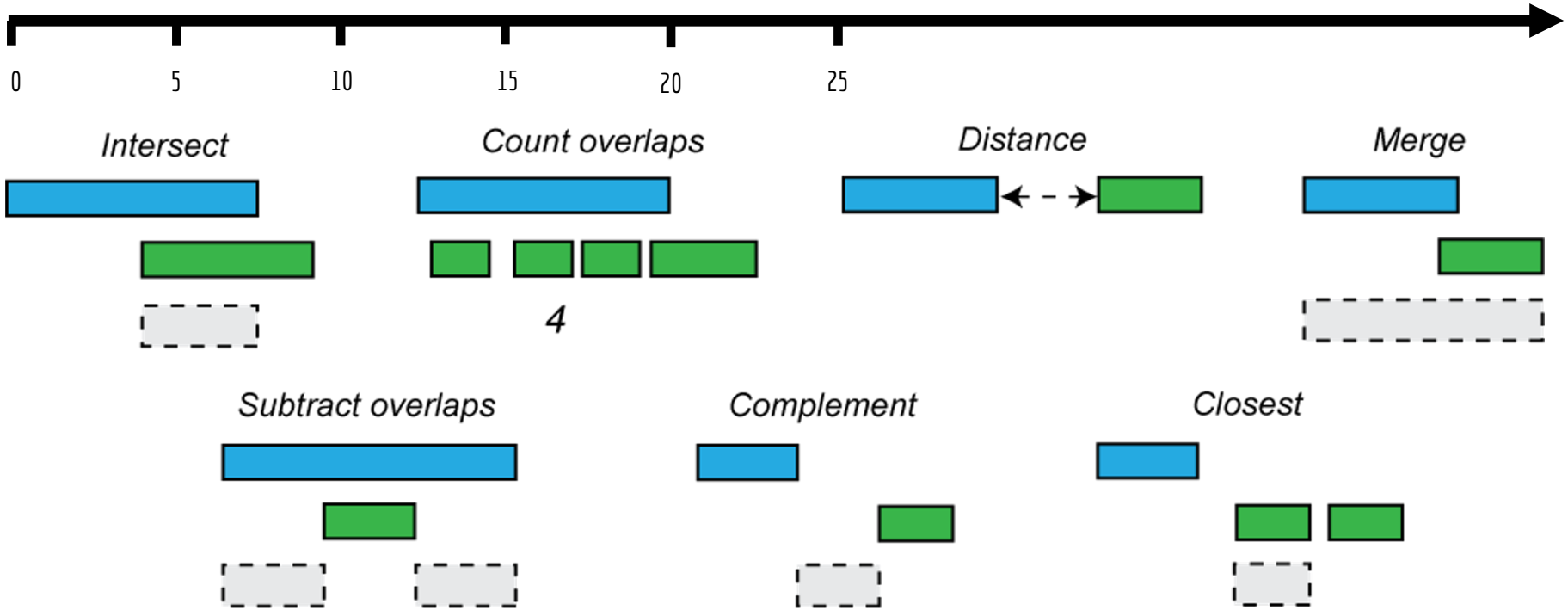
Genome arithmetic depends upon the genome coordinate system

- Is there overlap?
- How much overlap?

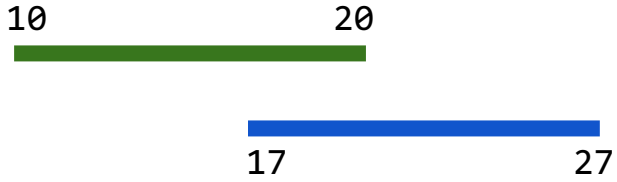


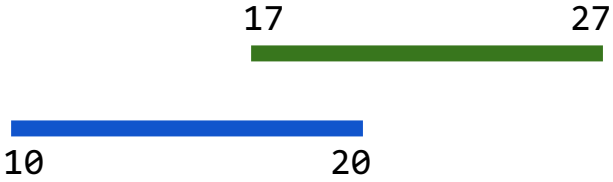

```
chr3    0     7  enhancer
chr3    4     10 TF_binding_site
```

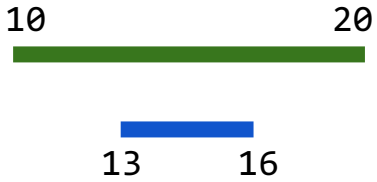
Genome arithmetic operations

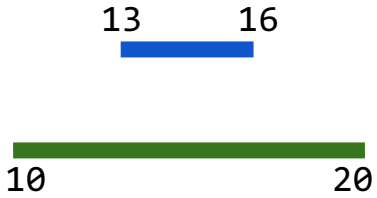


Do two intervals intersect (overlap)?









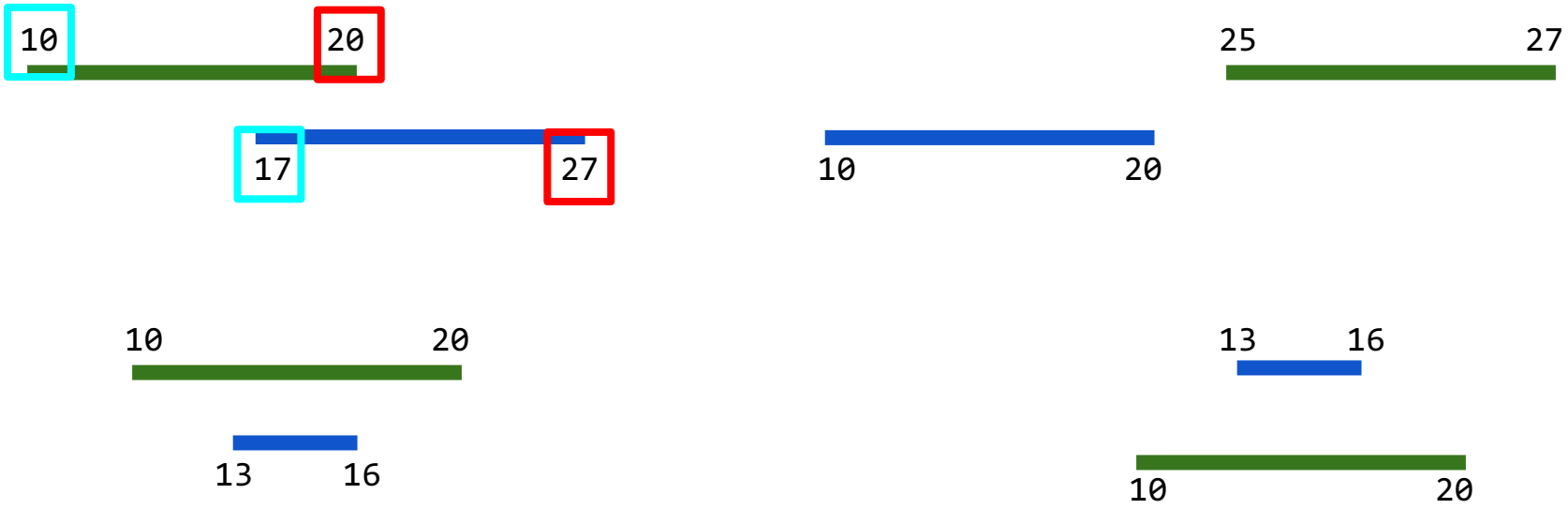

```
if ((a.start <= b.start and a.end >= b.start) or
    (b.start <= a.start and b.end >= a.start) or
    (a.start <= b.start and a.end >= b.end)   or
    (b.start <= a.start and b.end >= a.end))
{
  INTERSECTION!!!
}
else NADA!!!
```

Do two intervals intersect (overlap)? A simpler way.


```
I = min(a.end, b.end) - max(a.start, b.start)

  =min(20, 27) - max(10, 17)
       = 20 - 17 = 3
```

Do two intervals intersect (overlap)? A simpler way.



I = min(a.end, b.end) - max(a.start, b.start)

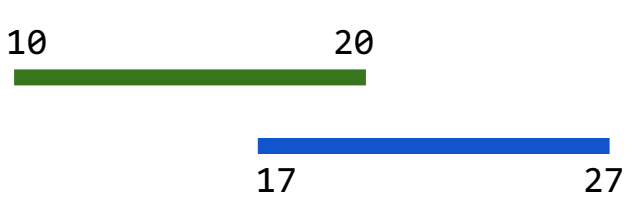
=min(20, 27) - max(10, 17)

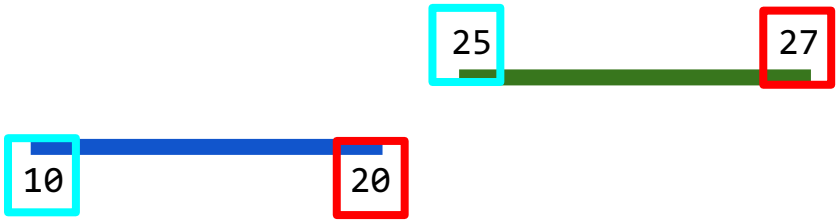
= 20 - 17 = 3

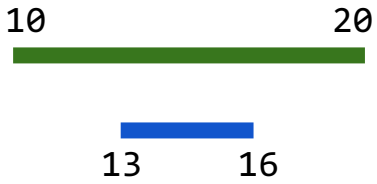
If I > 0 --> there is intersection

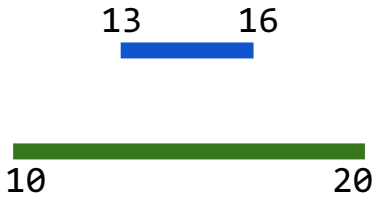
If I ≤ 0 --> there is no intersection

Do two intervals intersect (overlap)? A simpler way.









I = min(a.end, b.end) - max(a.start, b.start)

=min(20, 27) - max(10, 25)

= 20 - 25 = -5

If I > 0 --> there is intersection

If I ≤ 0 --> there is no intersection

Bedtools: a swiss army knife for genome analysis

BEDTools: a flexible suite of utilities for comparing genomic features

Aaron R. Quinlan ✉; Ira M. Hall ✉

Bioinformatics (2010) 26 (6): 841-842.
DOI: https://doi.org/10.1093/bioinformatics/btq033
Published: 28 January 2010 **Article history** ▾

Abstract

Motivation: Testing for correlations between different sets of genomic features is a fundamental task in genomics research. However, searching for overlaps between features with existing web-based methods is complicated by the massive datasets that are routinely produced with current sequencing technologies. Fast and flexible tools are therefore required to ask complex questions of these data in an efficient manner.

Results: This article introduces a new software suite for the comparison, manipulation and annotation of genomic features in Browser Extensible Data (BED) and General Feature Format (GFF) format. BEDTools also supports the comparison of sequence alignments in BAM format to both BED and GFF features. The tools are extremely efficient and allow the user to compare large datasets (e.g. next-generation sequencing data) with both public and custom genome annotation tracks. BEDTools can be combined with one another as well as with standard UNIX commands, thus facilitating routine genomics tasks as well as pipelines that can quickly answer intricate questions of large genomic datasets.

Papers:

https://doi.org/10.1093/bioinformatics/btq033
DOI: 10.1002/0471250953.bi1112s47

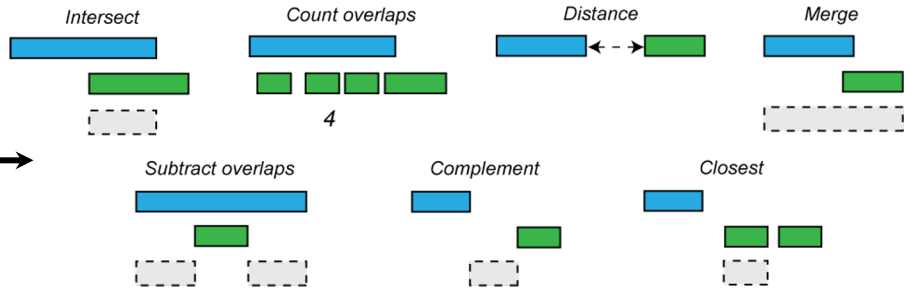
Documentation:

http://bedtools.readthedocs.io/en/latest/

Code:

https://github.com/arq5x/bedtools2

Supports most interval formats & handles diff. coordinate systems



Sort chromosomes lexicographically.

Then sort numerically by start coordinate

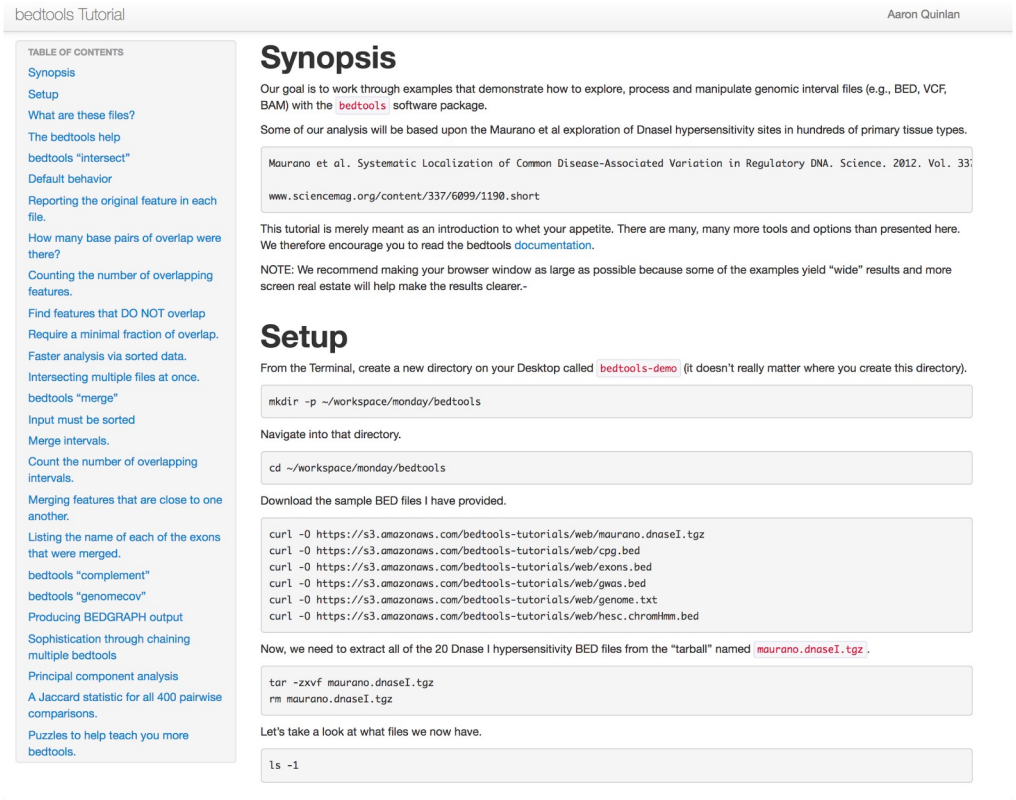
```
sort -k1,1 -k2,2n interval_1.bed > interval_1.sorted.bed
```

Bedtools: example analyses

- Closest gene to a ChIP-seq peak.
- How many genes does this mutation affect?
- Examine large-scale genetic abnormalities across mouse strains
- Where did I fail to collect sequence coverage?
- Is my favorite feature significantly correlated with some other feature?
- What is the density of variants in "windows" along the genome?

http://bedtools.readthedocs.io/en/latest/

Let's work through the bedtools tutorial.



bedtools Tutorial

Aaron Quinlan

TABLE OF CONTENTS

Synopsis

Our goal is to work through examples that demonstrate how to explore, process and manipulate genomic interval files (e.g., BED, VCF, BAM) with the `bedtools` software package.

Some of our analysis will be based upon the Maurano et al exploration of DnaseI hypersensitivity sites in hundreds of primary tissue types.

```
Maurano et al. Systematic Localization of Common Disease-Associated Variation in Regulatory DNA. Science. 2012. Vol. 33

www.sciencemag.org/content/337/6099/1190.short
```

This tutorial is merely meant as an introduction to whet your appetite. There are many, many more tools and options than presented here. We therefore encourage you to read the bedtools documentation.

NOTE: We recommend making your browser window as large as possible because some of the examples yield "wide" results and more screen real estate will help make the results clearer.-

Setup

From the Terminal, create a new directory on your Desktop called `bedtools-demo` (it doesn't really matter where you create this directory).

```
mkdir -p ~/workspace/monday/bedtools
```

Navigate into that directory.

```
cd ~/workspace/monday/bedtools
```

Download the sample BED files I have provided.

```
curl -O https://s3.amazonaws.com/bedtools-tutorials/web/maurano.dnaseI.tgz
curl -O https://s3.amazonaws.com/bedtools-tutorials/web/cpg.bed
curl -O https://s3.amazonaws.com/bedtools-tutorials/web/exons.bed
curl -O https://s3.amazonaws.com/bedtools-tutorials/web/gwas.bed
curl -O https://s3.amazonaws.com/bedtools-tutorials/web/genome.txt
curl -O https://s3.amazonaws.com/bedtools-tutorials/web/hesc.chromHmm.bed
```

Now, we need to extract all of the 20 Dnase I hypersensitivity BED files from the "tarball" named `maurano.dnaseI.tgz`.

```
tar -zxvf maurano.dnaseI.tgz
rm maurano.dnaseI.tgz
```

Let's take a look at what files we now have.

```
ls -1
```

Connect to malibu.

```
mkdir bedtools-tutorial
cd bedtools-tutorial
```

https://bedtools.readthedocs.io/en/latest/content/bedtools-suite.html