Genome arithmetic

what, why, how, and how not

Aaron Quinlan
BIOCH 5080
Mar. 20, 2013
arq5x@virginia.edu
quinlanlab.org

Goals of today's lecture

- Overview of genome features
- What is genome arithmetic?
- Genome arithmetic approaches
- The UCSC "binning" algorithm
- An introduction to BEDTools
- Some real world examples

What should you do to reinforce the lecture material?

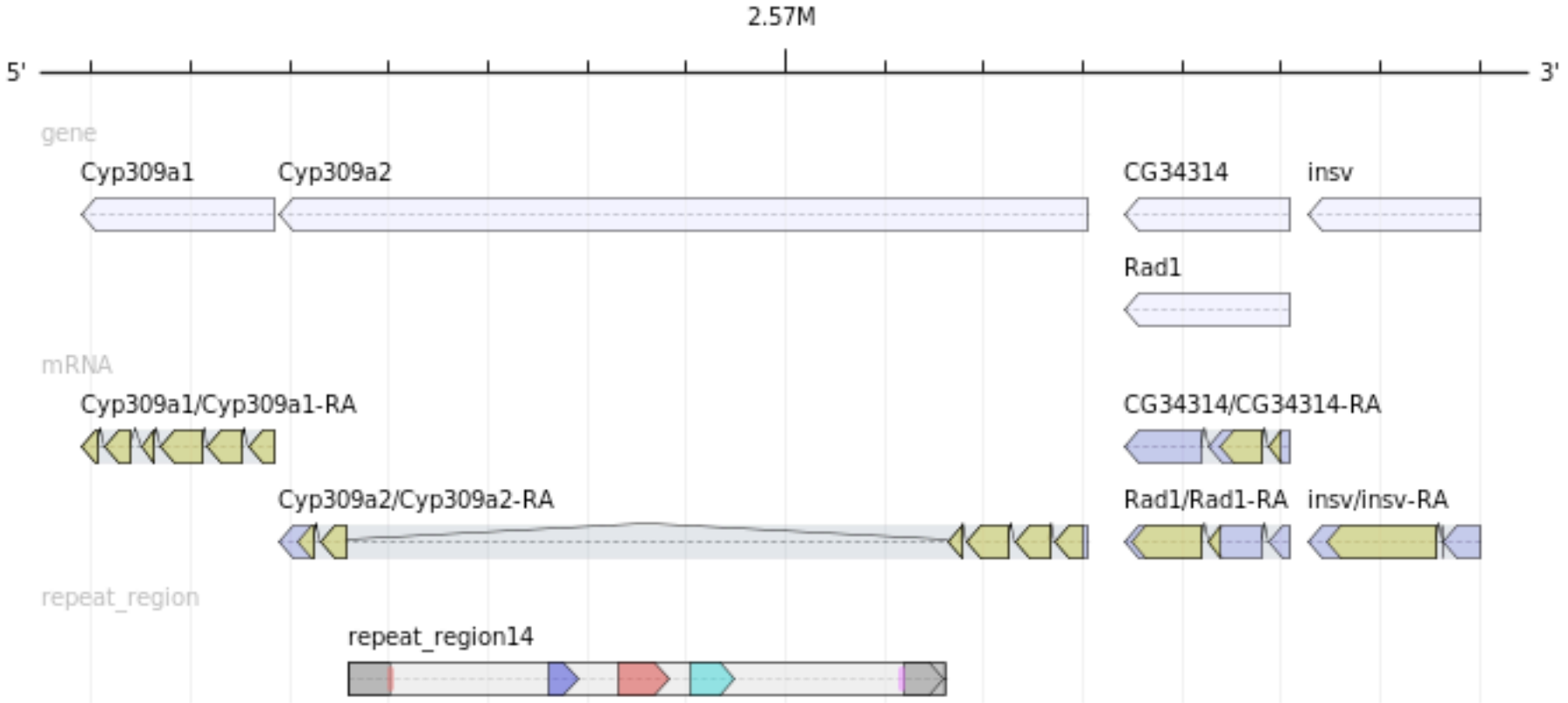
- Recreate examples in lecture
- Read BEDTools docs [1]
- Watch Galaxy screencasts [2]
- Download genome features from UCSC and use BEDTools on the command line
- Review common file formats [3]

[1] http://bedtools.readthedocs.org
[2] http://main.g2.bx.psu.edu/
[3] http://genome.ucsc.edu/FAQ/FAQformat

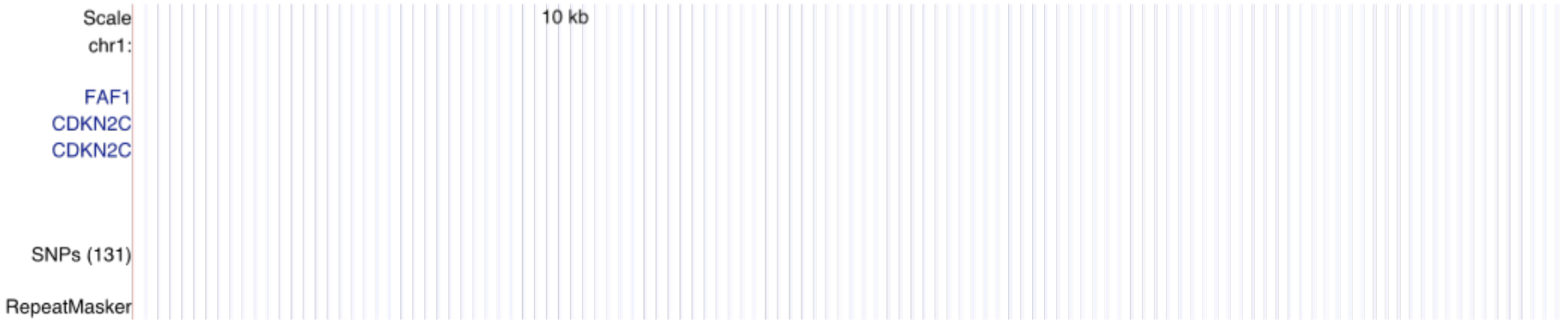
What is a genome "feature"?

- Genes: exons, introns, UTRs, promoters
- Conservation
- Genetic variation
- Transposons
- Origins of replication
- TF binding sites
- CpG islands
- Segmental duplications
- Sequence alignments
- Chromatin annotations
- Gene expression data
- …
- **Your own observations: put them in context**

Genome "features"

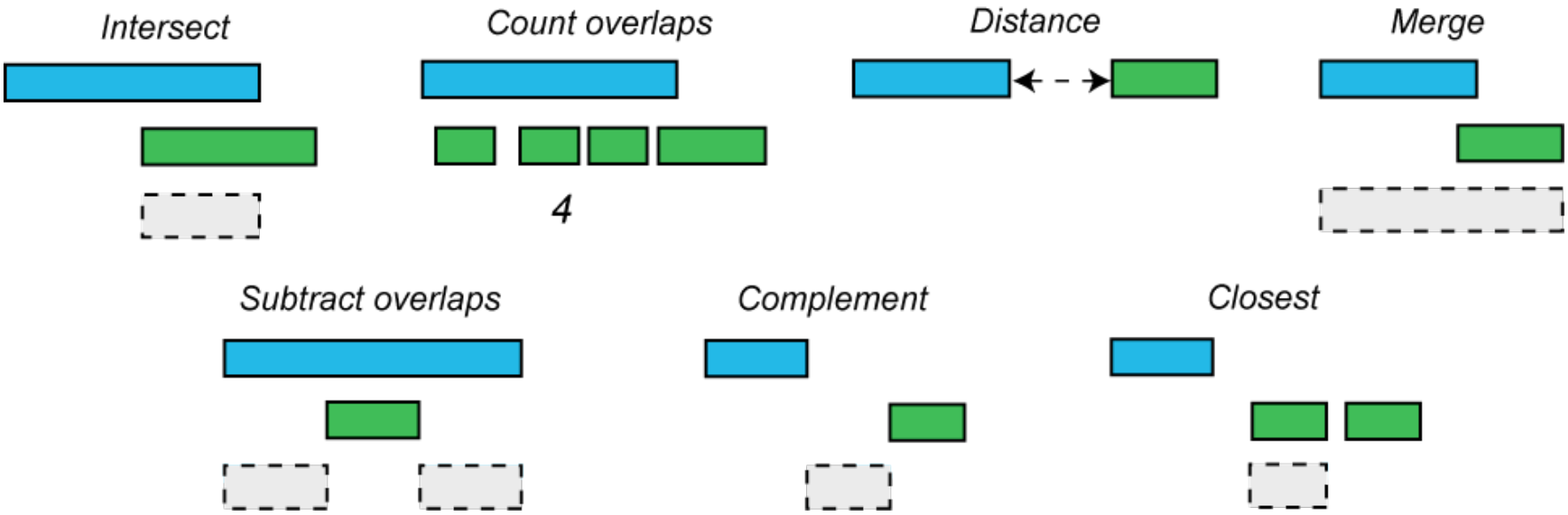


Overlaps, closest, merge…



What is genome arithmetic?

"Set theory on the genome"



Answerable questions

- Closest gene to a ChIP-seq peak.
- Is my latest discovery novel?
- Is there strand bias in my data?
- How many genes does this mutation affect?
- Where did I fail to collect sequence coverage?
- Is my favorite feature significantly correlated with some other feature?

An annoying wealth of formats

- BED
- BEDGRAPH
- WIG
- GFF
- VCF
- SAM/BAM
- …

No standards: Some formats use 1-based coordinates, while others use 0-based

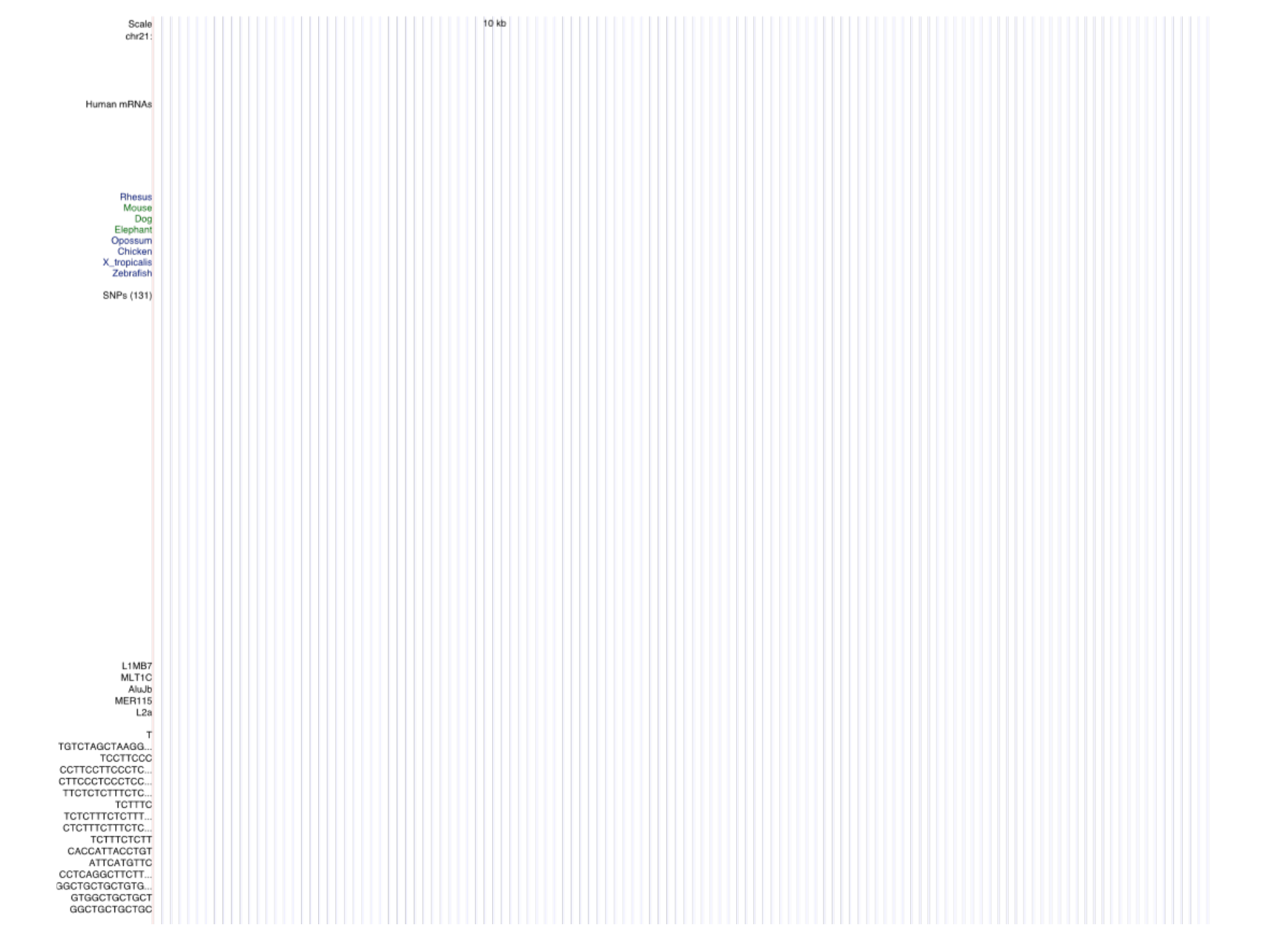
An annoying wealth of formats

- BED
- BEDGRAPH
- WIG
- GFF
- VCF
- SAM/BAM
- …

Standard attributes

- chrom
- start position
- end position
- name / label
- score / value
- strand
- other

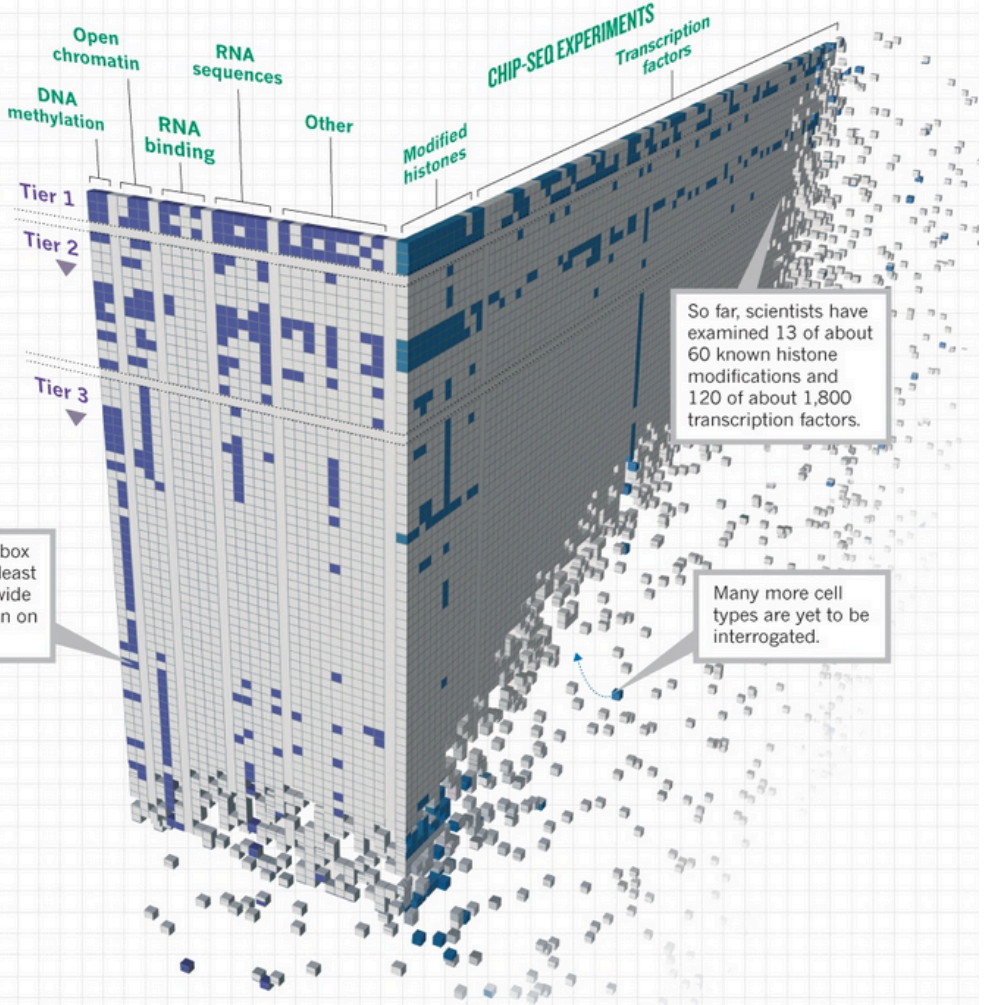
No standards: Some formats use 1-based coordinates, while others use 0-based


Scale
chr21:
10 kb
Human mRNAs
Rhesus
Mouse
Dog
Elephant
Opossum
Chicken
X_tropicalis
Zebrafish
SNPs (131)
L1MB7
MLT1C
AluJb
MER115
L2a
T
TGTCTAGCTAAGG...
TCCTTCCC
CCTTCCTTCCCTC...
CTTCCCTCCCTCC...
TTCTCTCTTTCTC...
TCTTTC
TCTCTTTCTCTTT...
CTCTTTCTTTCTC...
TCTTTCTCTT
CACCATTACCTGT
ATTCATGTTC
CCTCAGGCTTCTT...
GGCTGCTGCTGTG...
GTGGCTGCTGCT
GGCTGCTGCTGC

Why?

MAKING A GENOME MANUAL

Scientists in the Encyclopedia of DNA Elements Consortium have applied 24 experiment types (across) to more than 150 cell lines (down) to assign functions to as many DNA regions as possible — but the project is still far from complete.



EXPERIMENTAL TARGETS

DNA methylation: regions layered with chemical methyl groups, which regulate gene expression.

Open chromatin: areas in which the DNA and proteins that make up chromatin are accessible to regulatory proteins.

RNA binding: positions where regulatory proteins attach to RNA.

RNA sequences: regions that are transcribed into RNA.

ChIP-seq: technique that reveals where proteins bind to DNA.

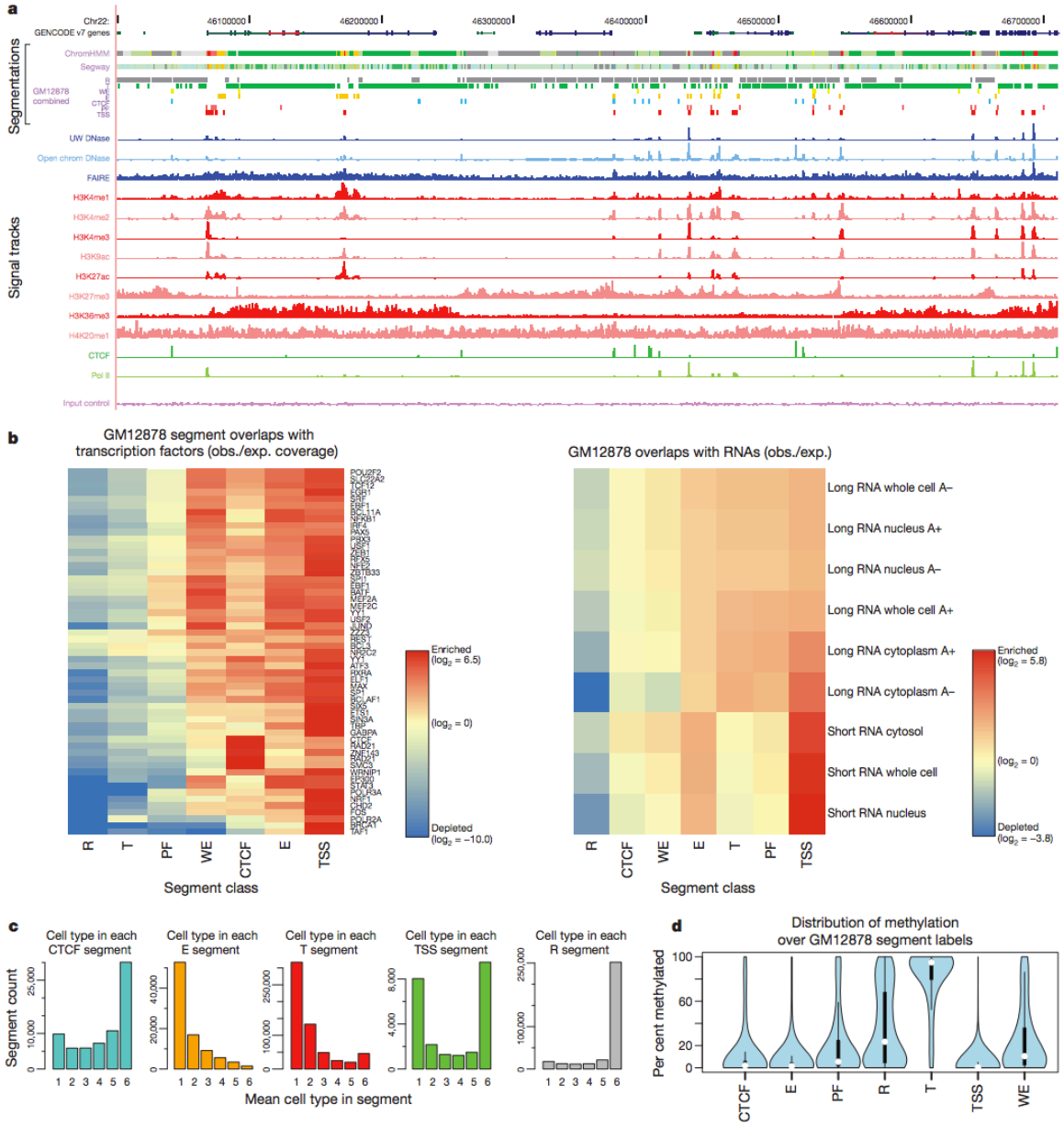
Modified histones: histone proteins, which package DNA into chromosomes, modified by chemical marks.

Transcription factors: proteins that bind to DNA and regulate transcription.

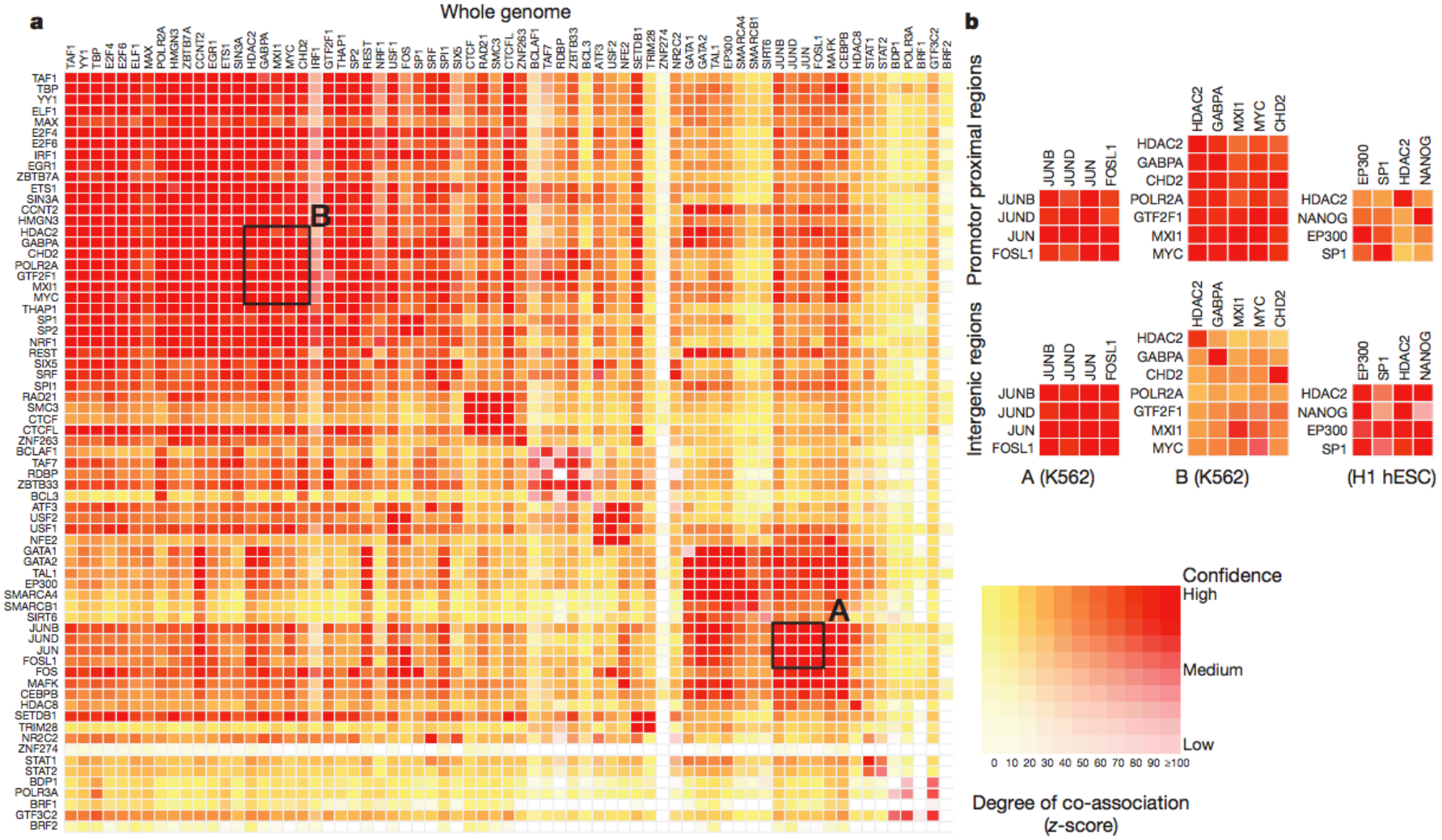
CELL LINES

Tiers 1 and 2: widely used cell lines that were given priority.

Tier 3: all other cell types.



http://www.nature.com/nature/journal/v489/n7414/pdf/nature11247.pdf



http://www.nature.com/nature/journal/v489/n7414/pdf/nature11247.pdf

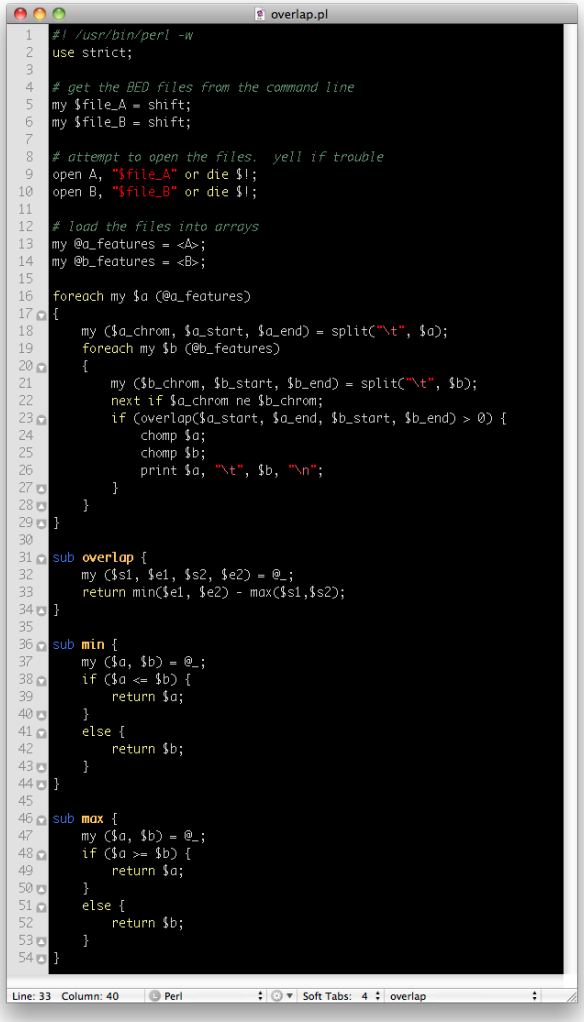
How?

Searching for overlaps

- Brute force: loop over all N features
  - Can be tragically slow when N is large
- Build a "tree"
  - R-tree
  - NCList
  - UCSC "binning" algorithm
- Extend the sort & "merge" algorithm
  - Widely used to "join" database table

Brute force: slooooow

```
foreach fA in file A
{
   foreach fB in file B
   {
      does fA overlap fB?*
   }
}
```



* When A and B contain many features, the answer will be NO the vast majority of the time. This means most of the processing is spent NOT finding features that overlap.

Narrow the search space w/ "binning"

the UCSC genome browser approach

Human chrom 1: ca. 250 Mb

Narrow the search space w/ "binning"

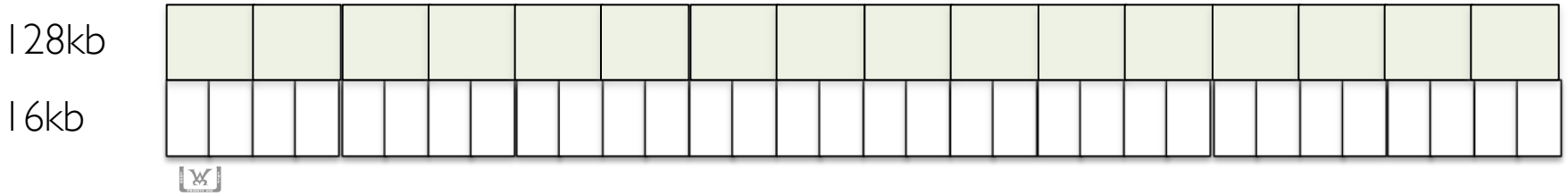
the UCSC genome browser approach

16kb

Human chrom 1: ca. 250 Mb

Narrow the search space w/ "binning"

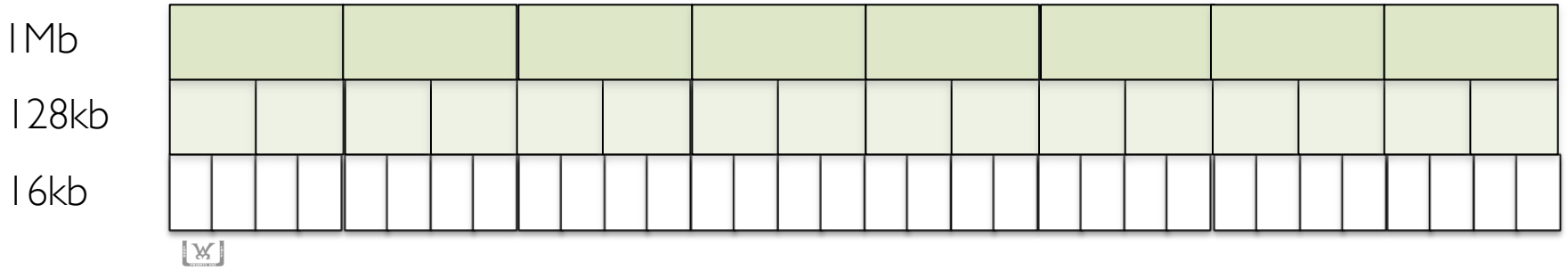
the UCSC genome browser approach



Human chrom 1: ca. 250 Mb

Narrow the search space w/ "binning"

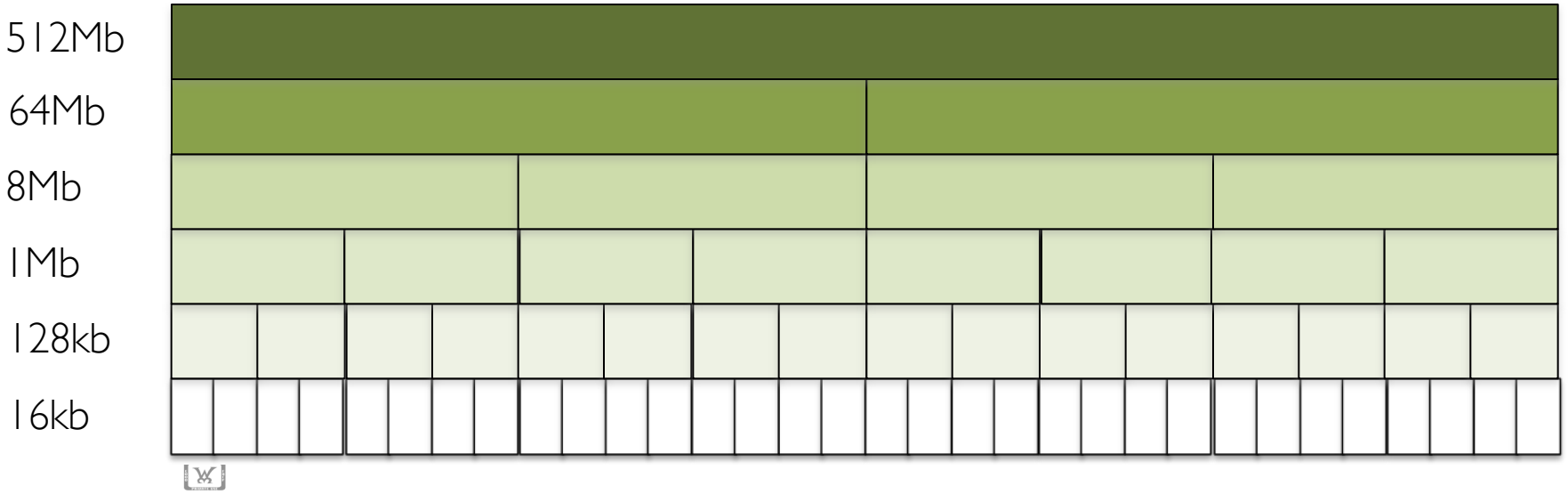
the UCSC genome browser approach



Human chrom 1: ca. 250 Mb

Narrow the search space w/ "binning"

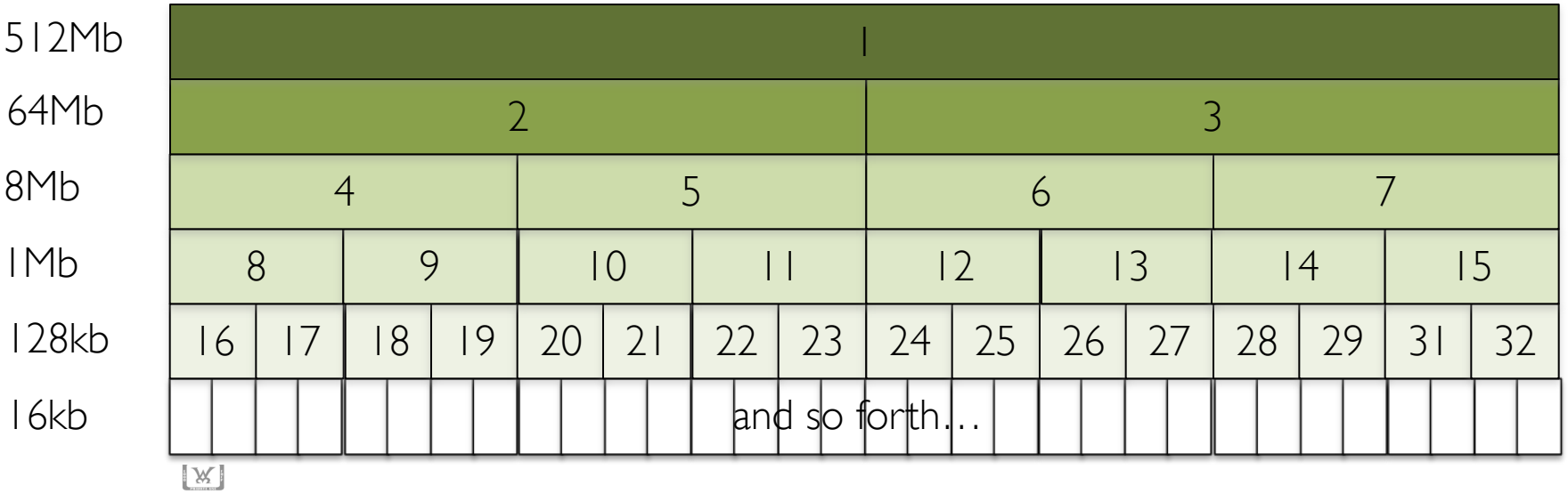
the UCSC genome browser approach



Human chrom 1: ca. 250 Mb

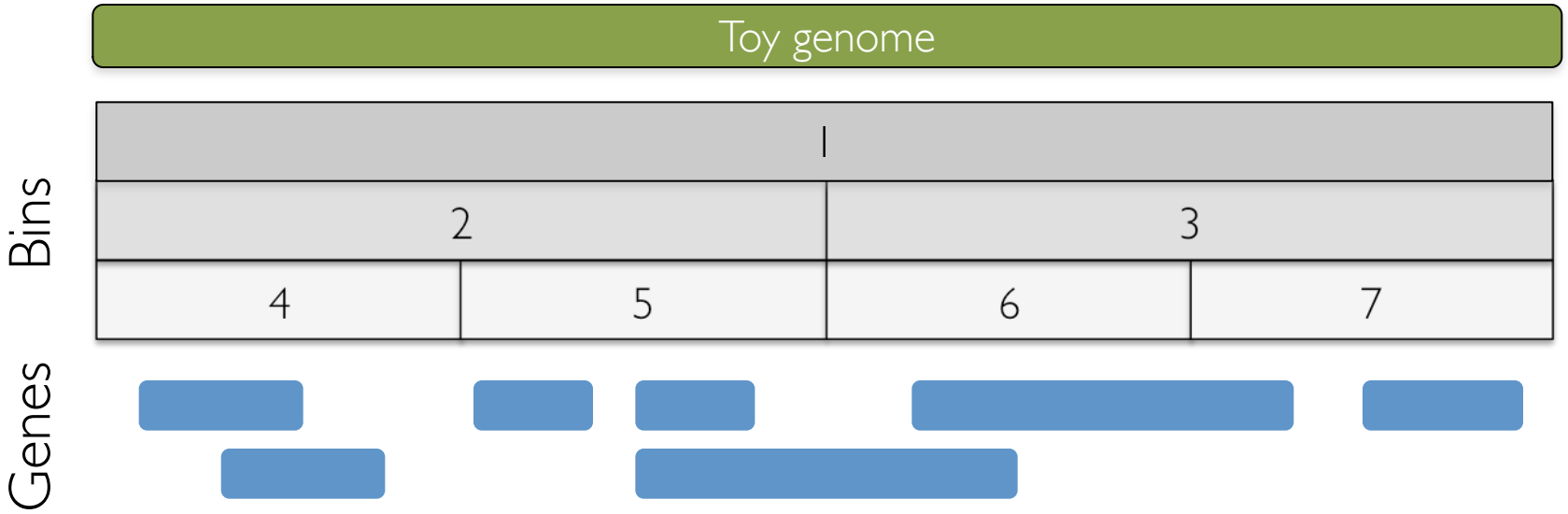
Narrow the search space w/ "binning"

the UCSC genome browser approach

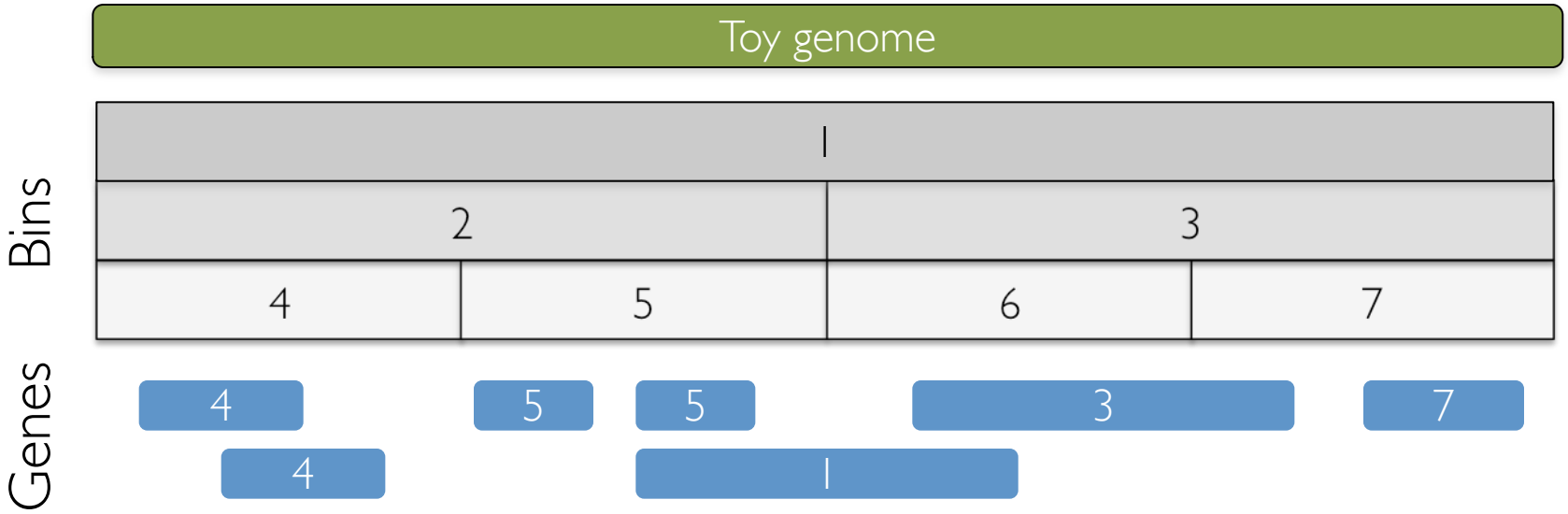


Human chrom 1: ca. 250 Mb

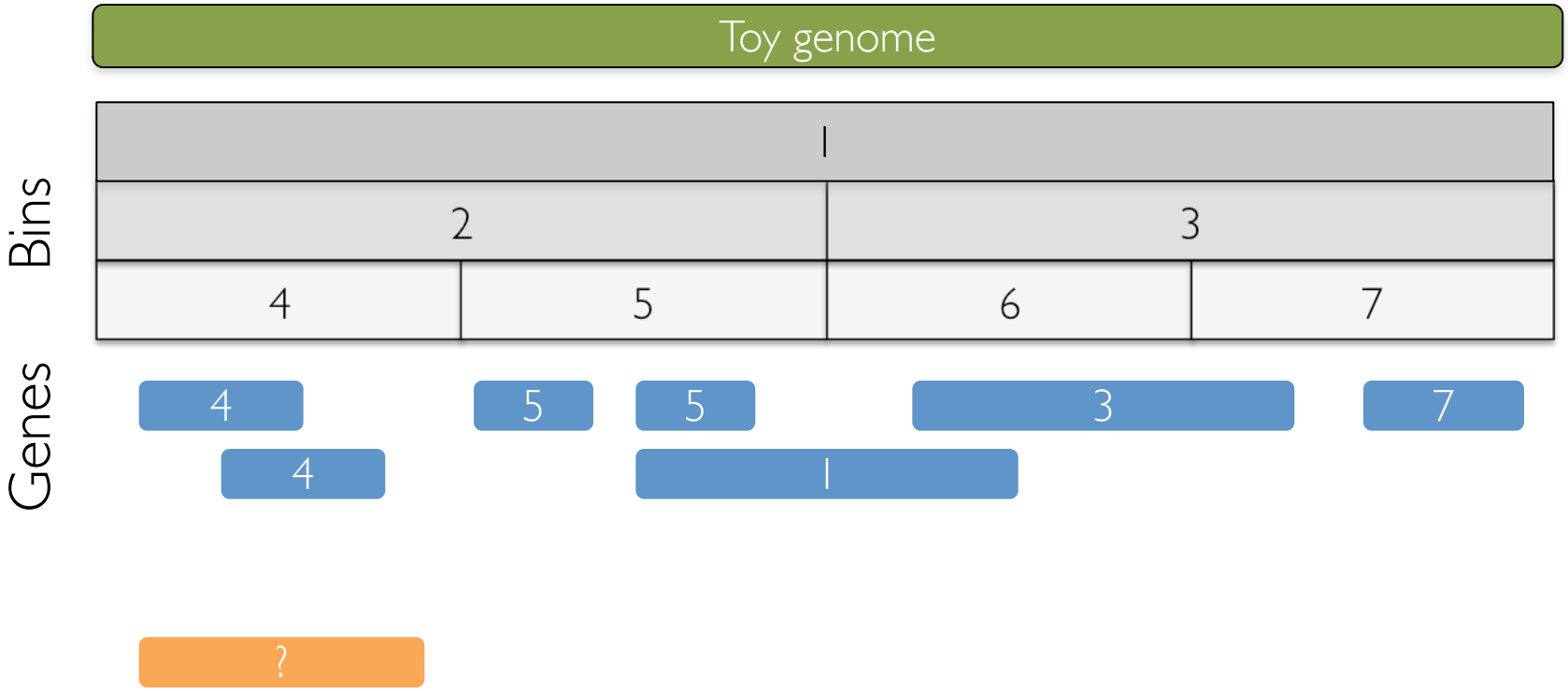
A toy example



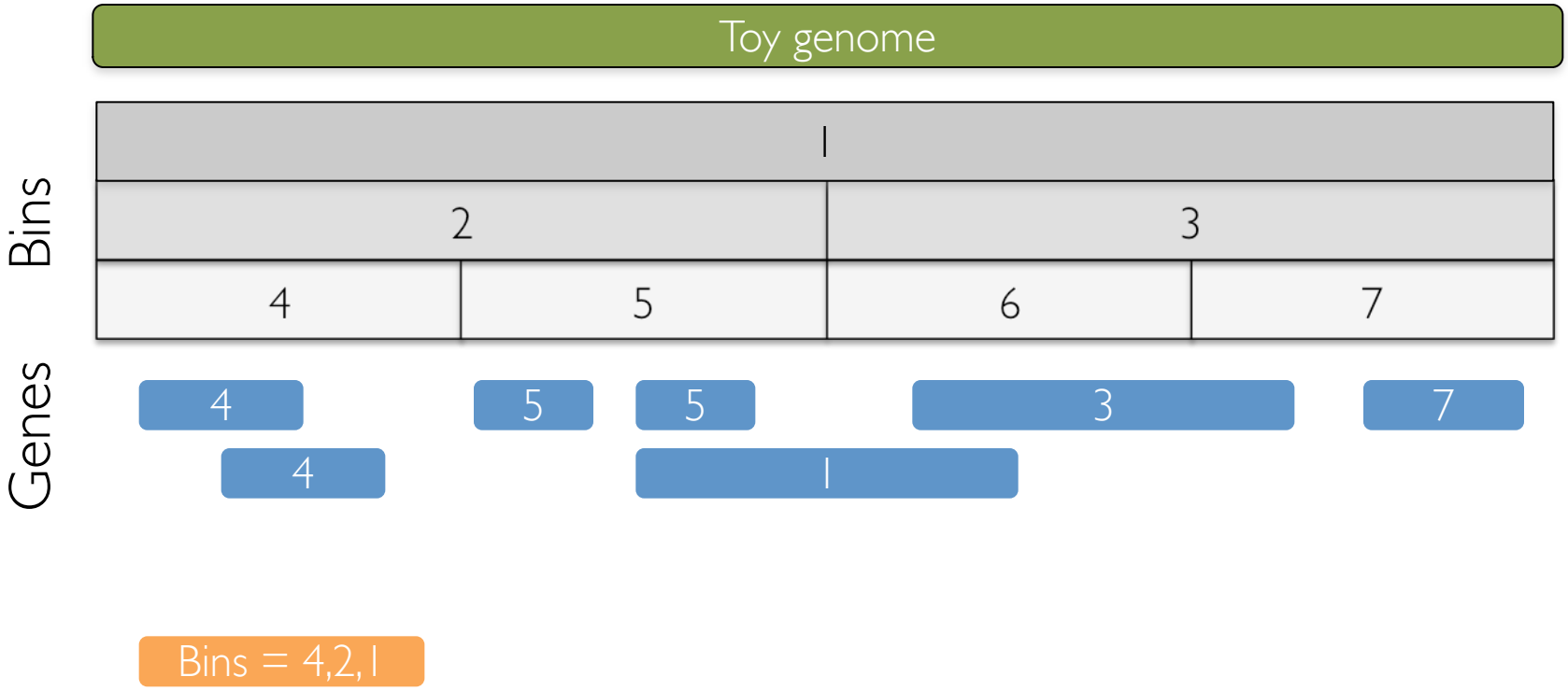
A toy example



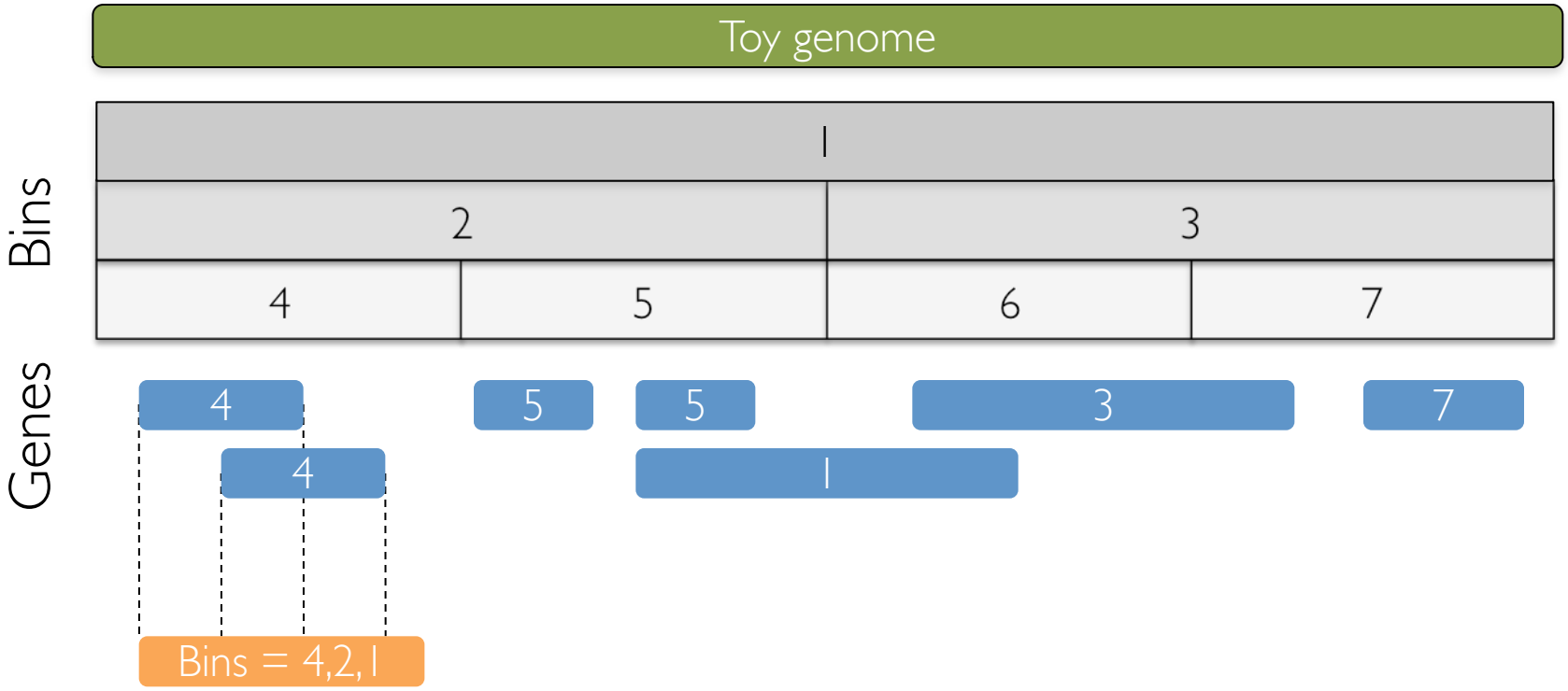
A toy example



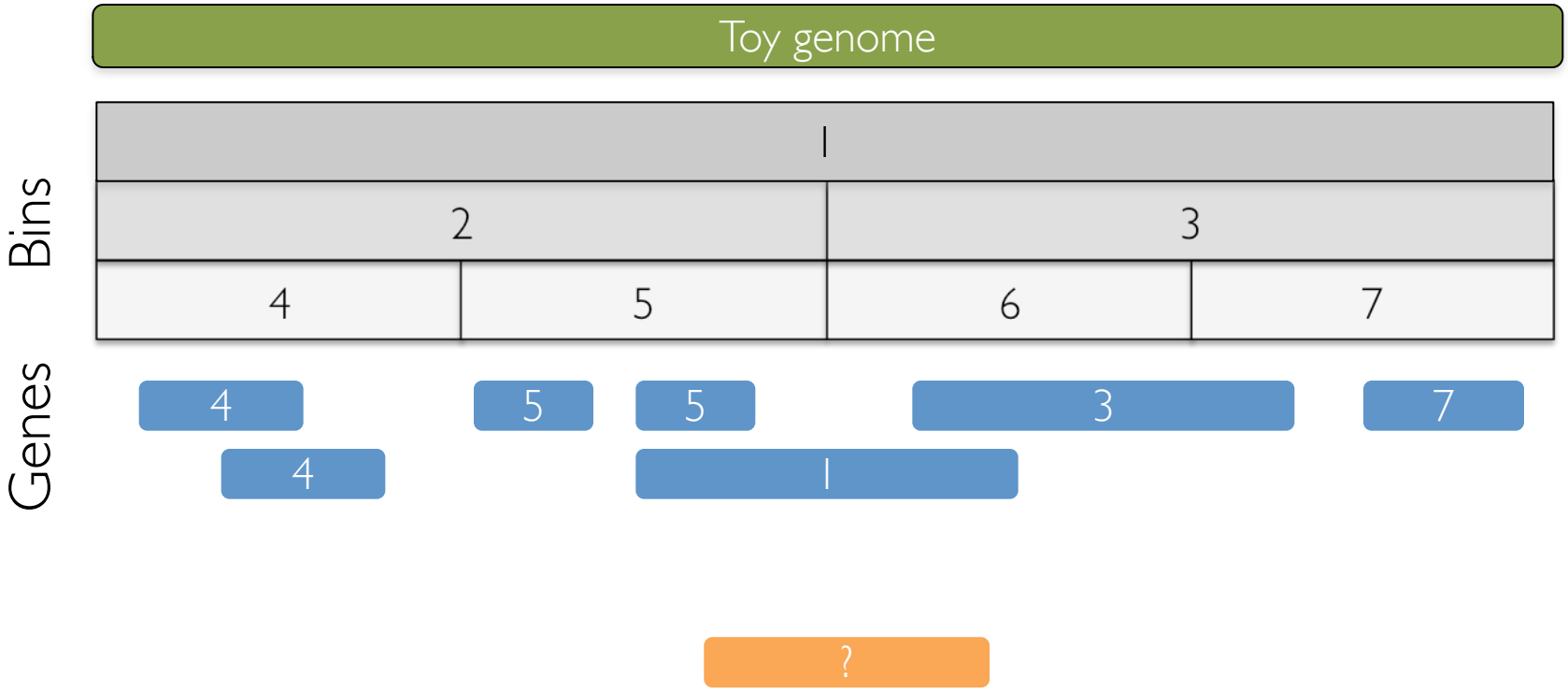
A toy example



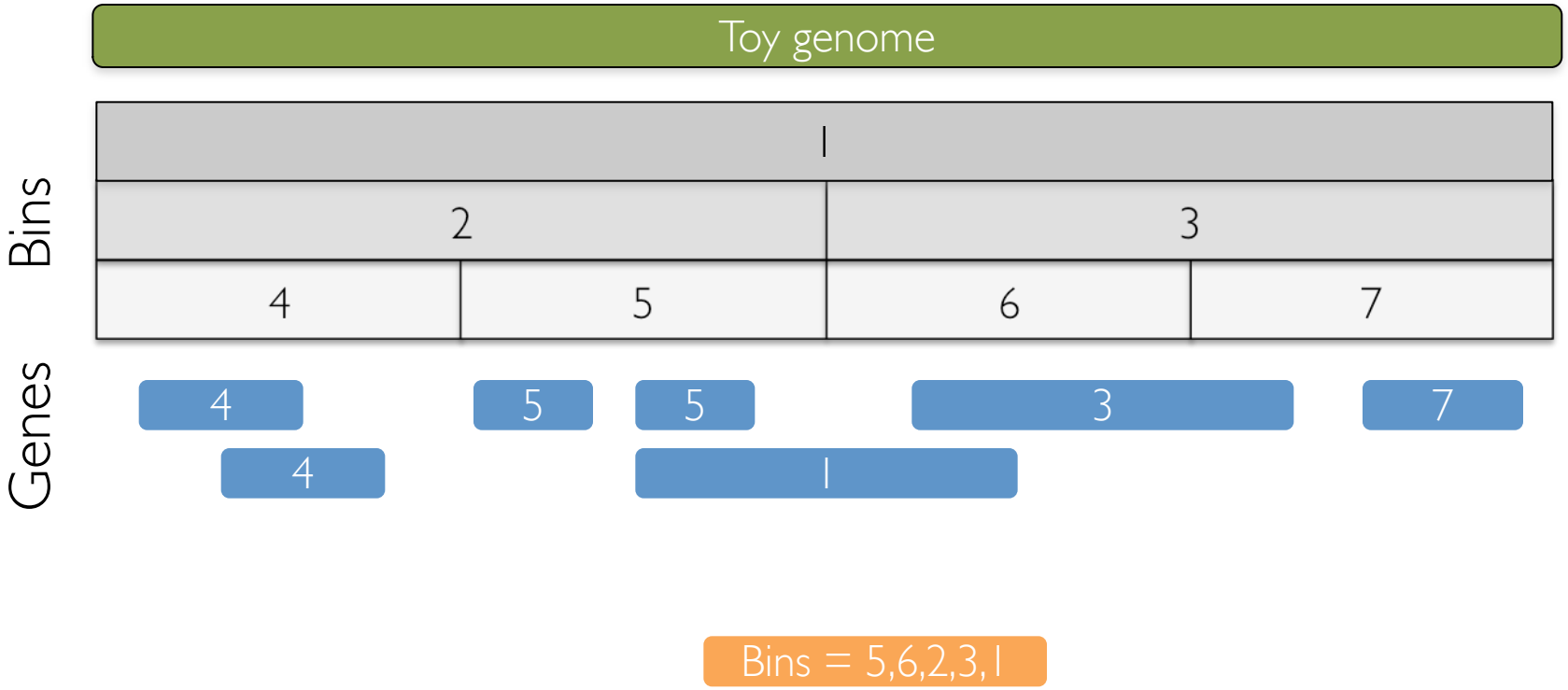
A toy example



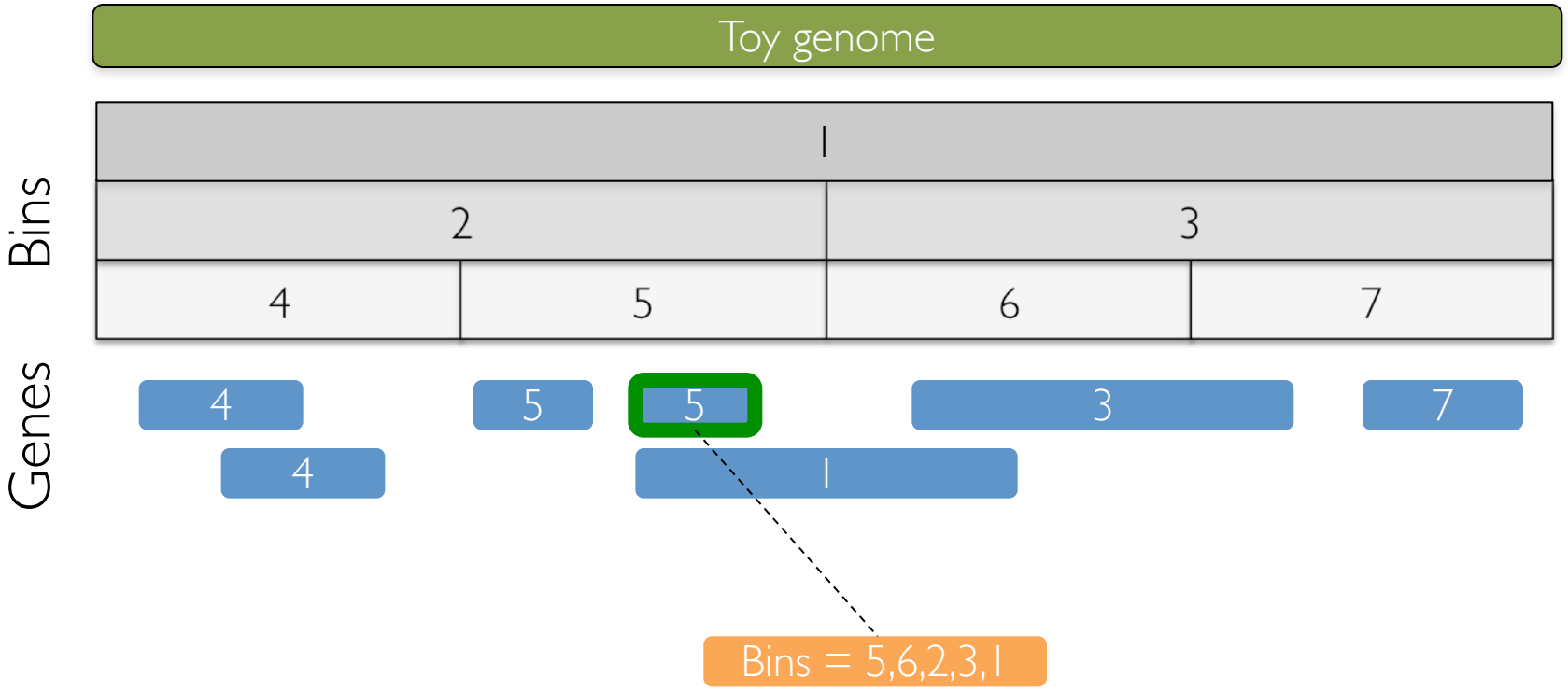
A toy example



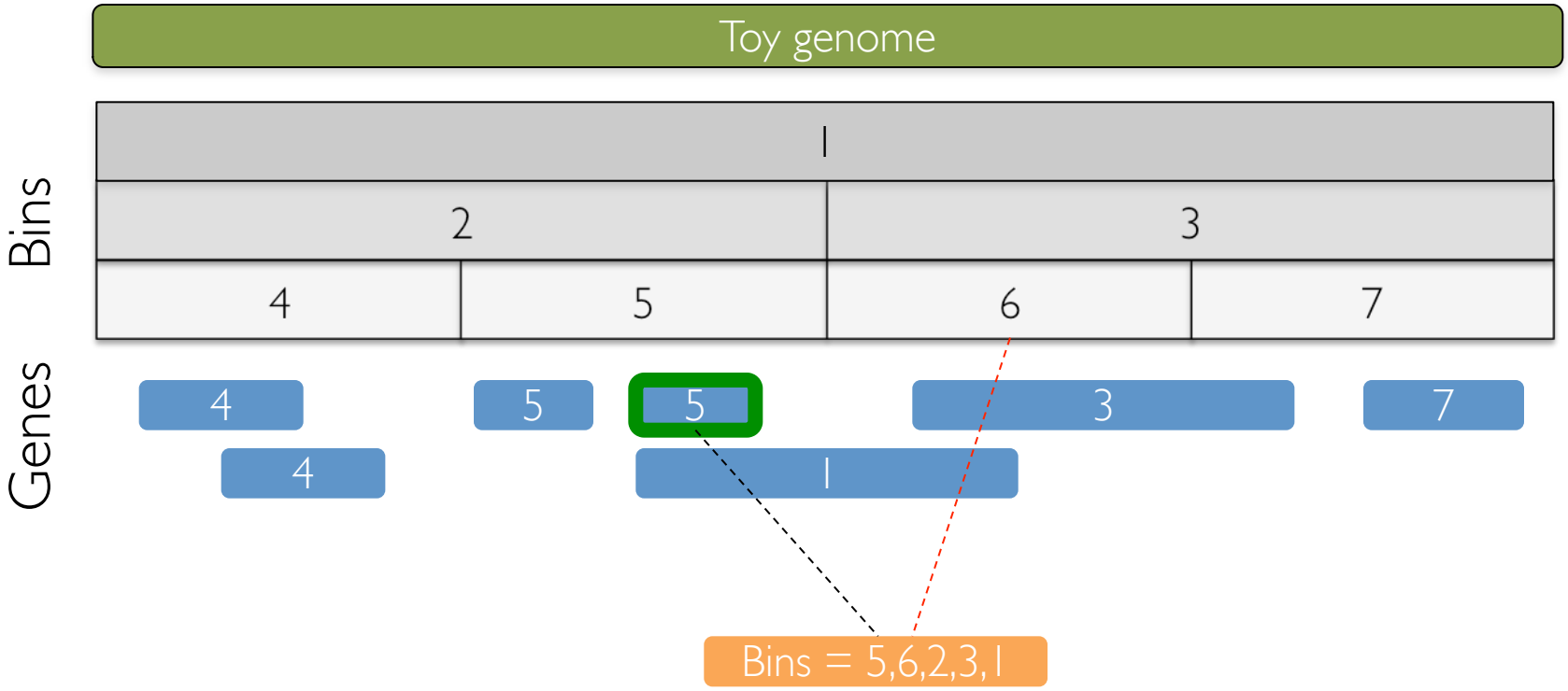
A toy example



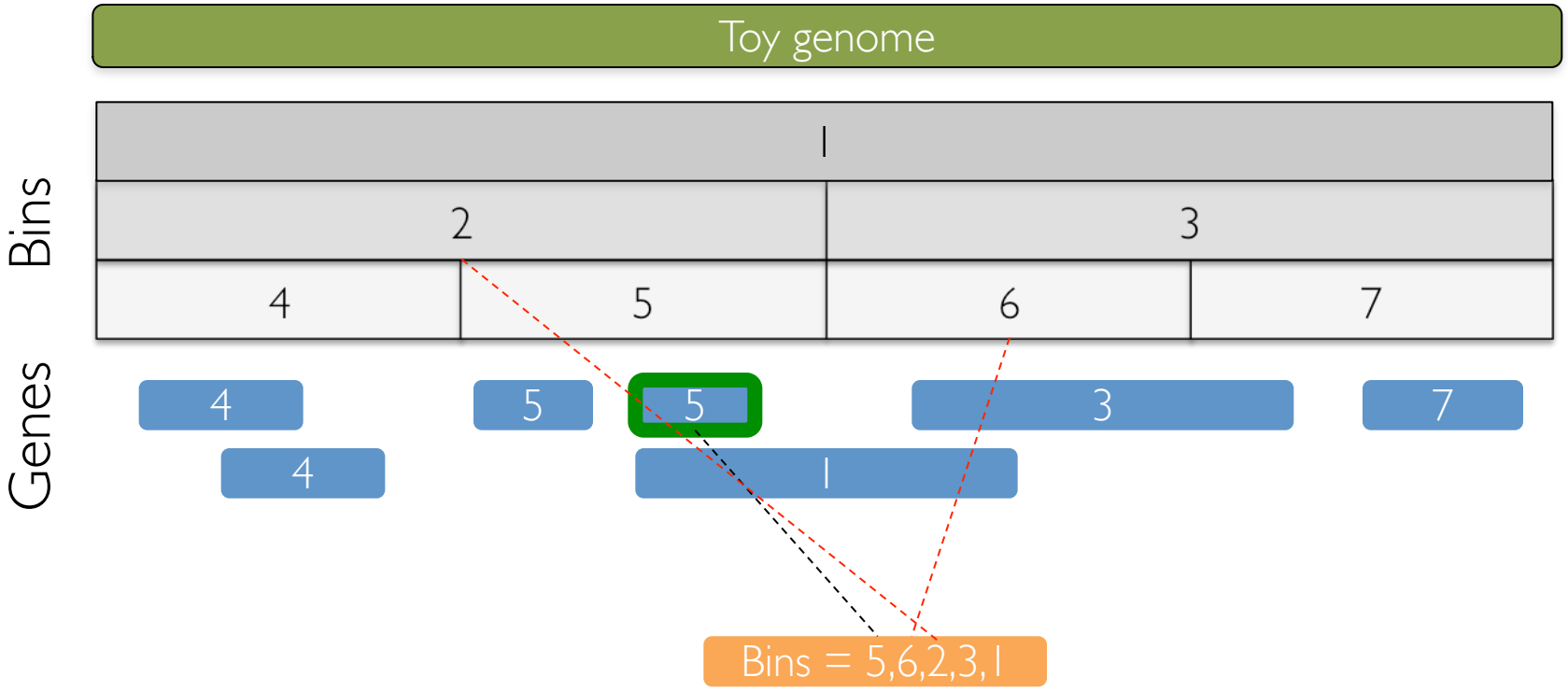
A toy example



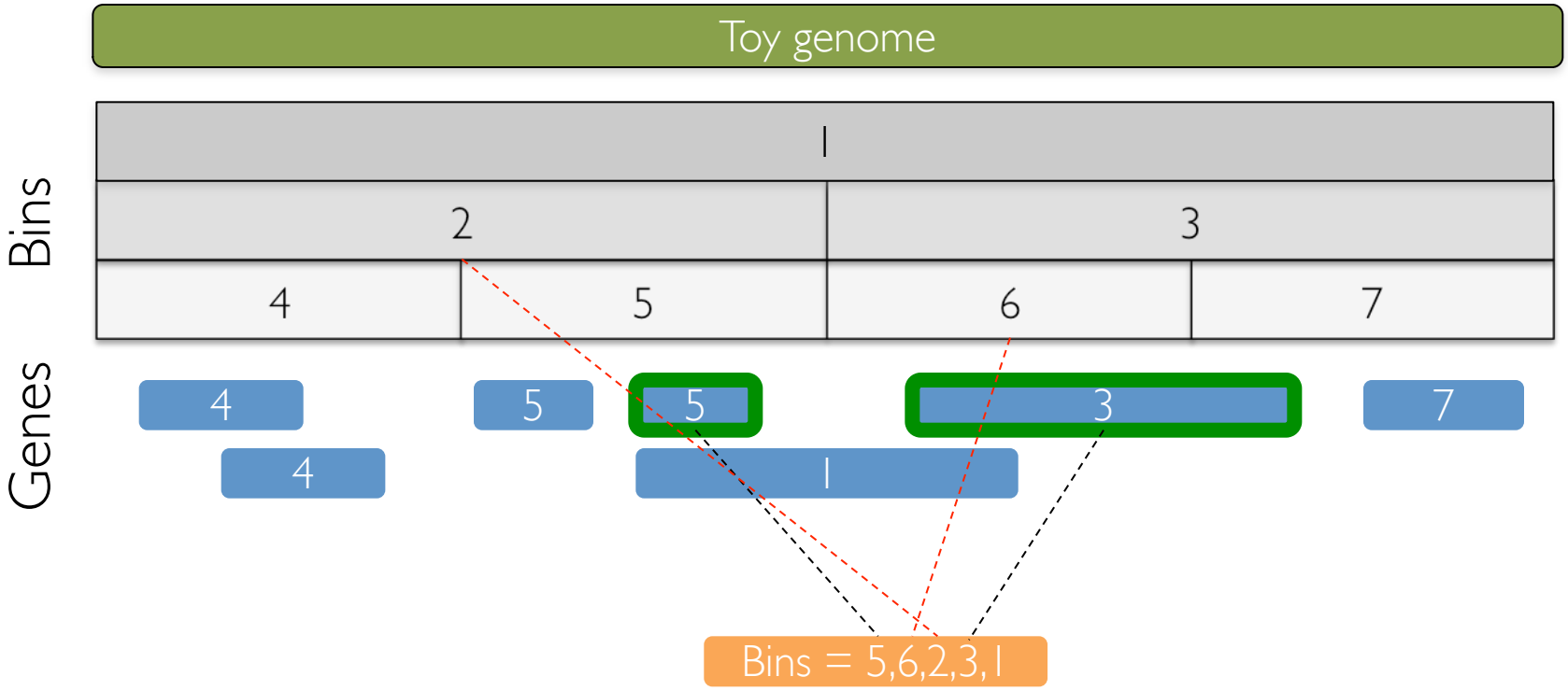
A toy example



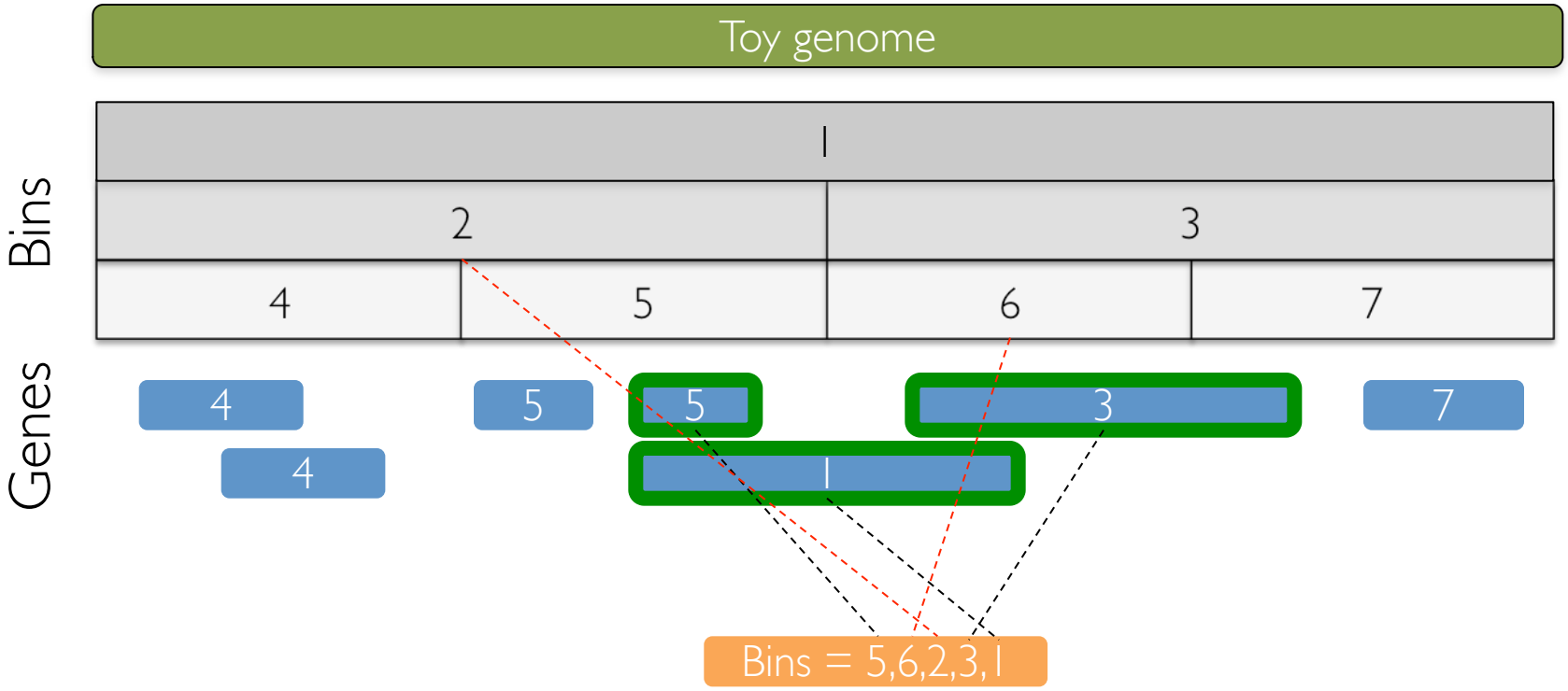
A toy example



A toy example



A toy example

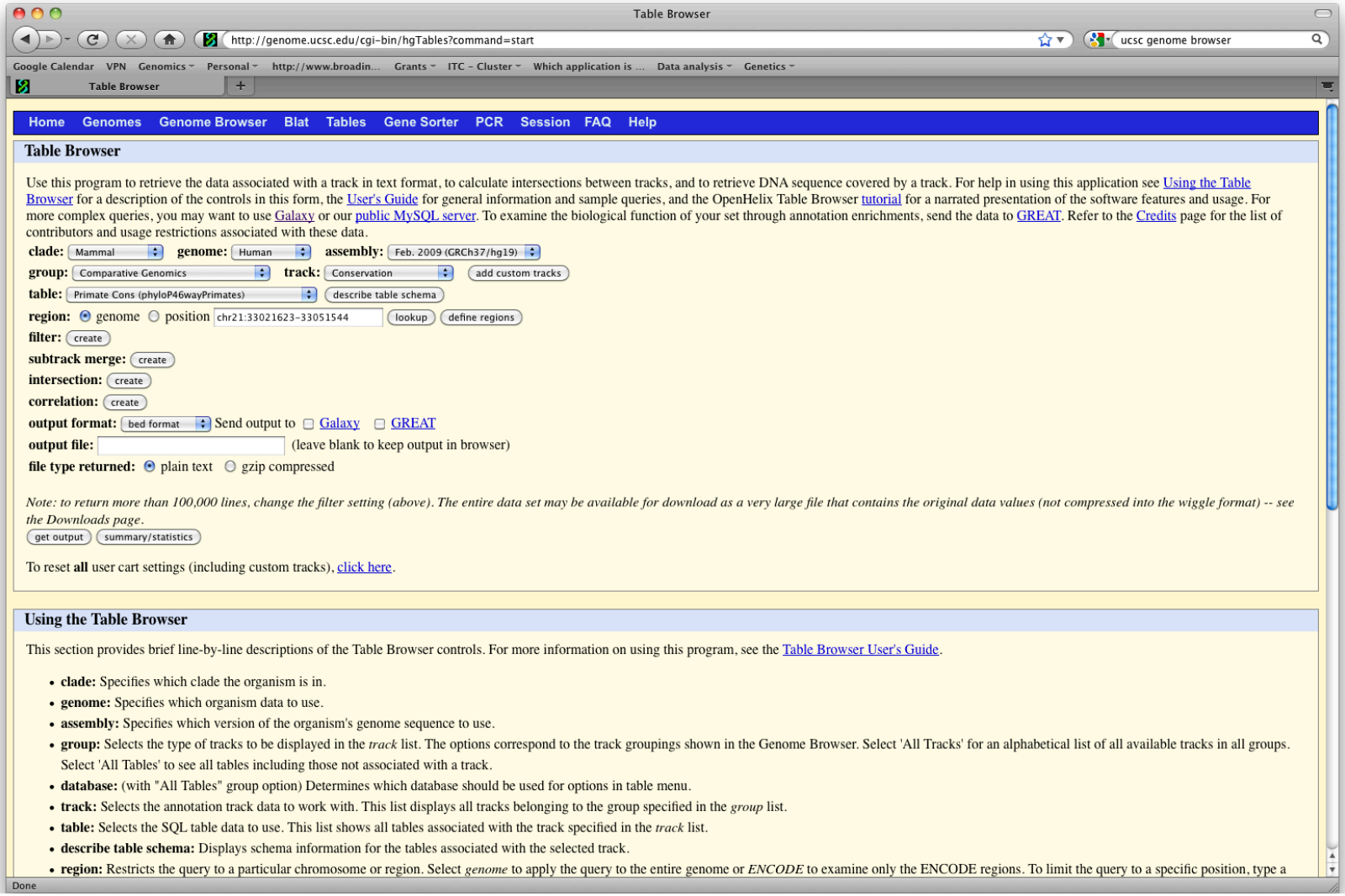


Available tools for GA

UCSC Genome Browser

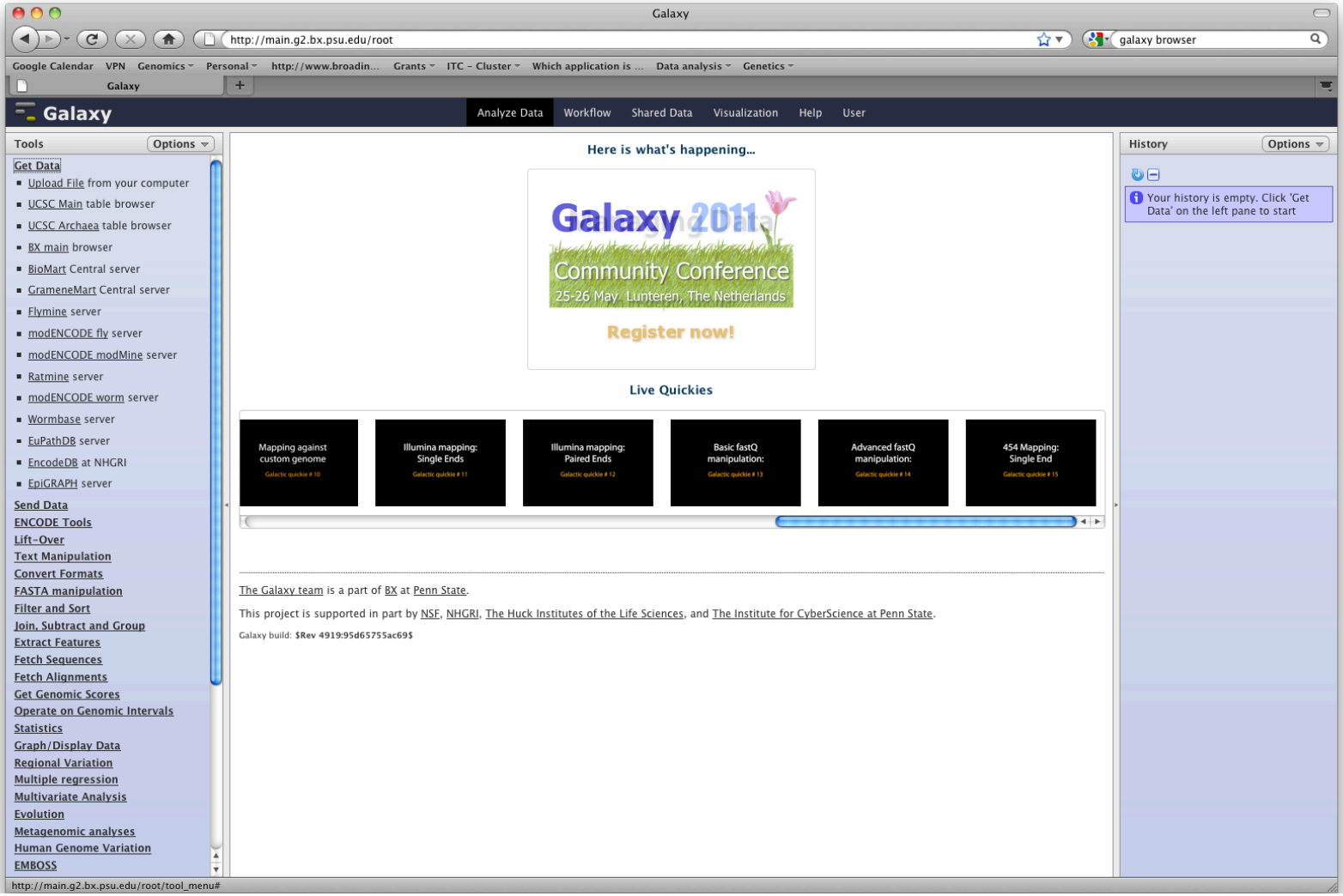
UCSC Genome Browser

http://genome.ucsc.edu/cgi-bin/hgTables?command=start



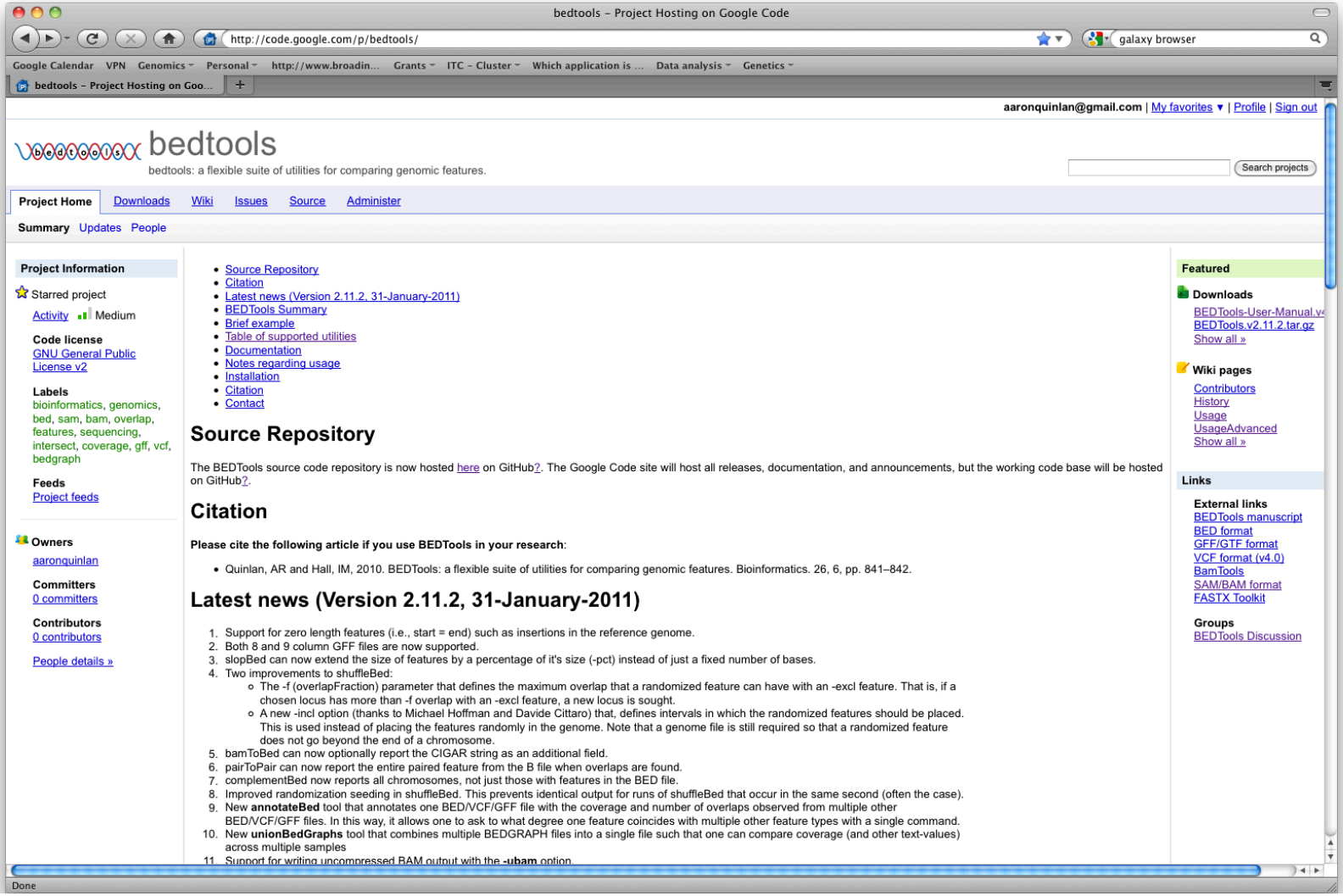
Galaxy

http://main.g2.bx.psu.edu/root

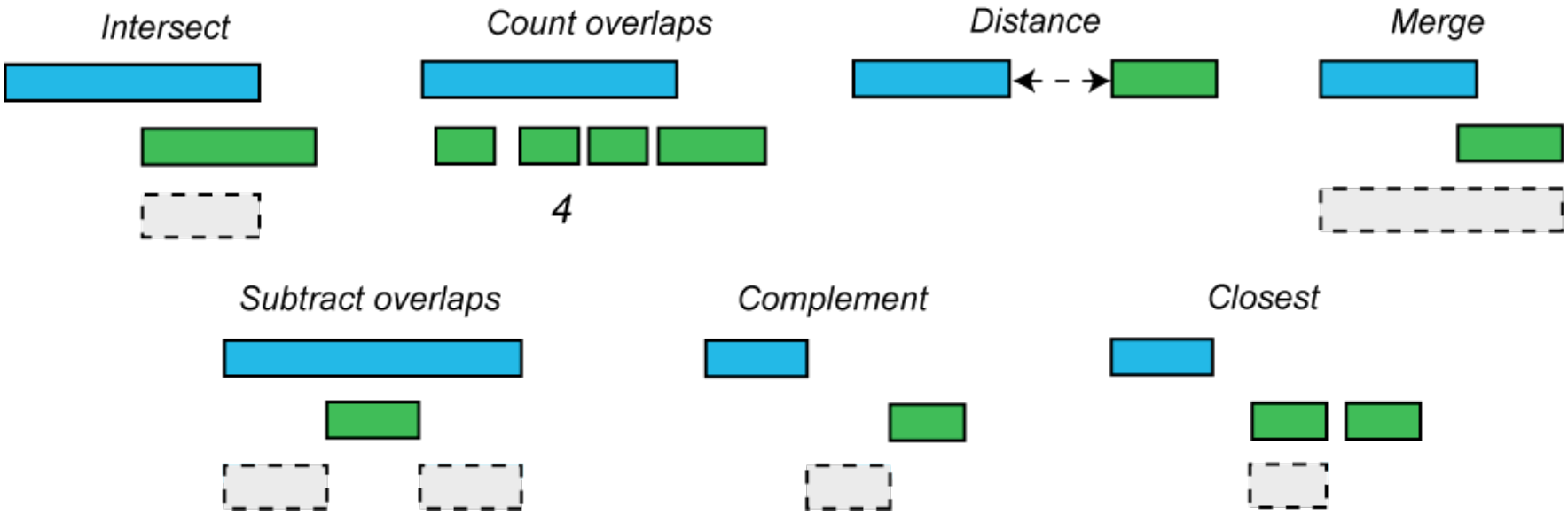


BEDTools

http://code.google.com/p/bedtools/



BEDTools supports common GA operations and common formats



BEDTools + UNIX

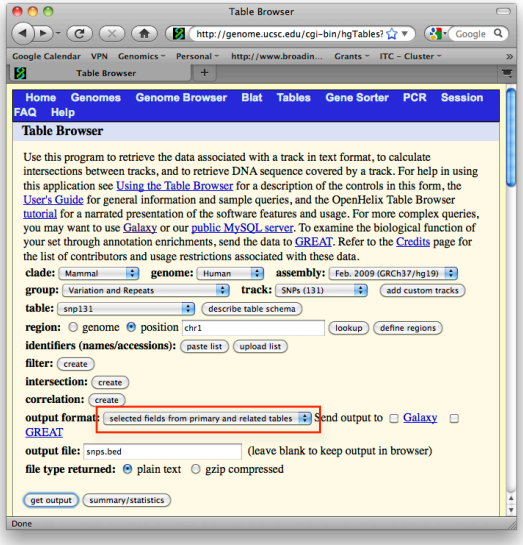
Real world usage

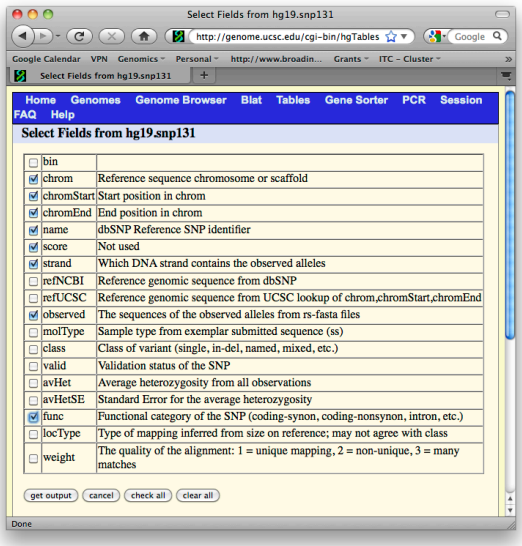
Find SNPs that have the potential to alter gene expression regulation by

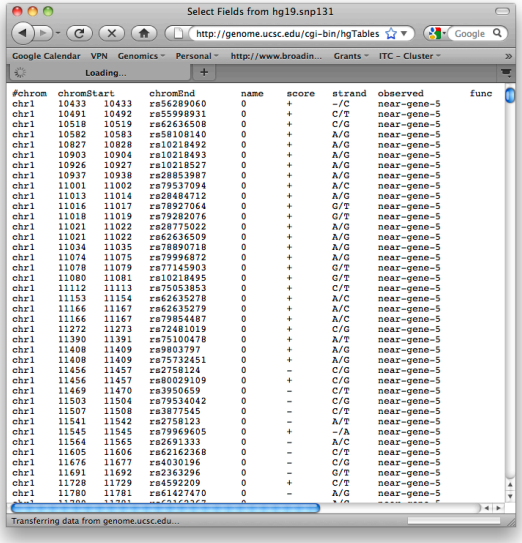
Step 1: Get annotations

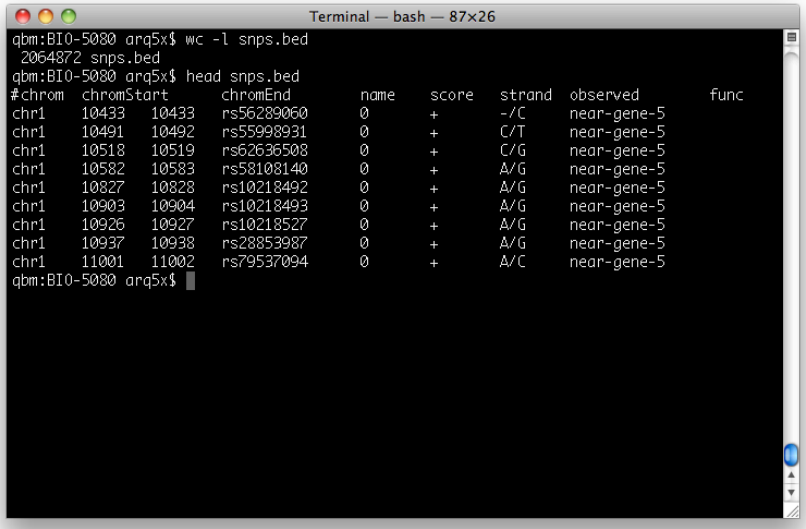
- Single-nucleotide polymorphisms (SNPs)
- CpG islands
- Genes

SNPs from dbSNP via UCSC



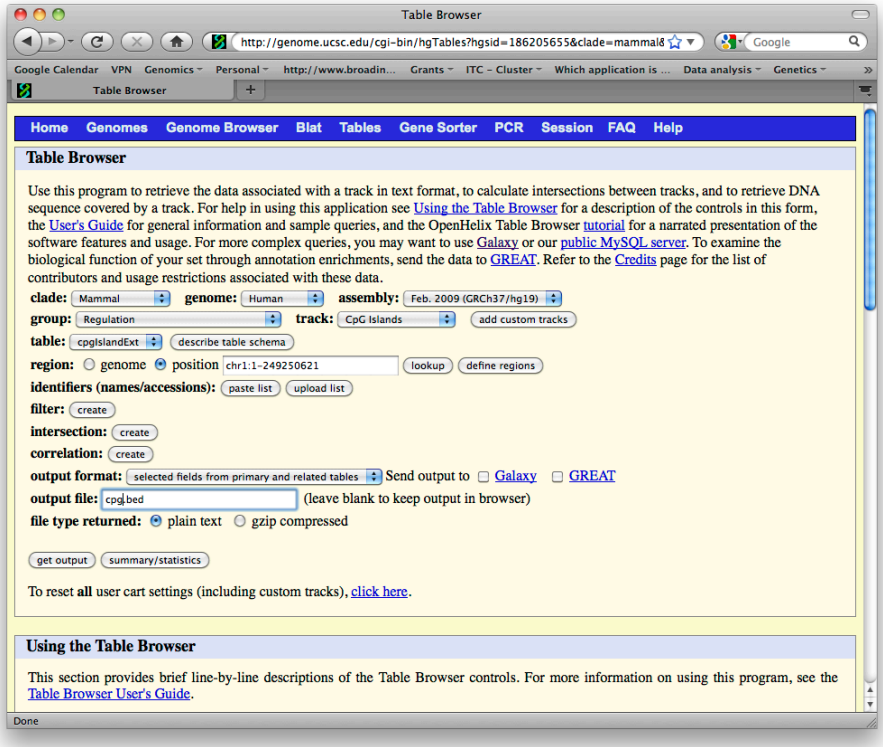


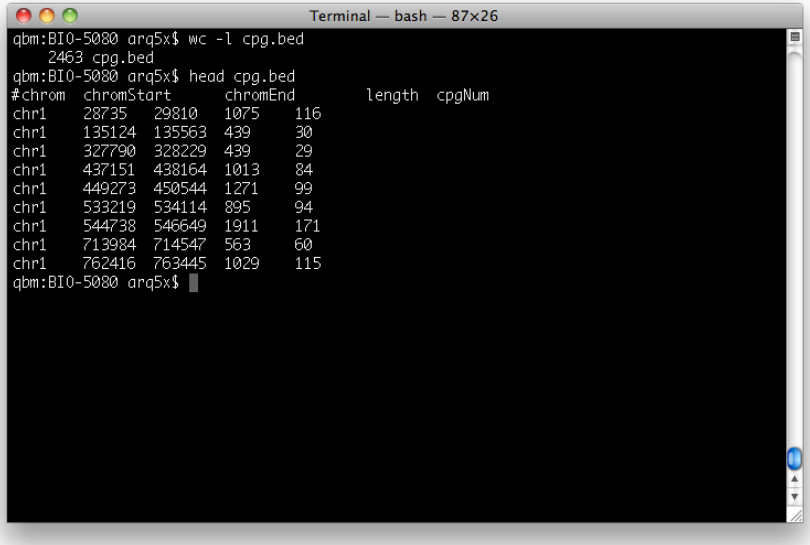




"snps.bed"
N = 2,064,872

CpG islands via UCSC





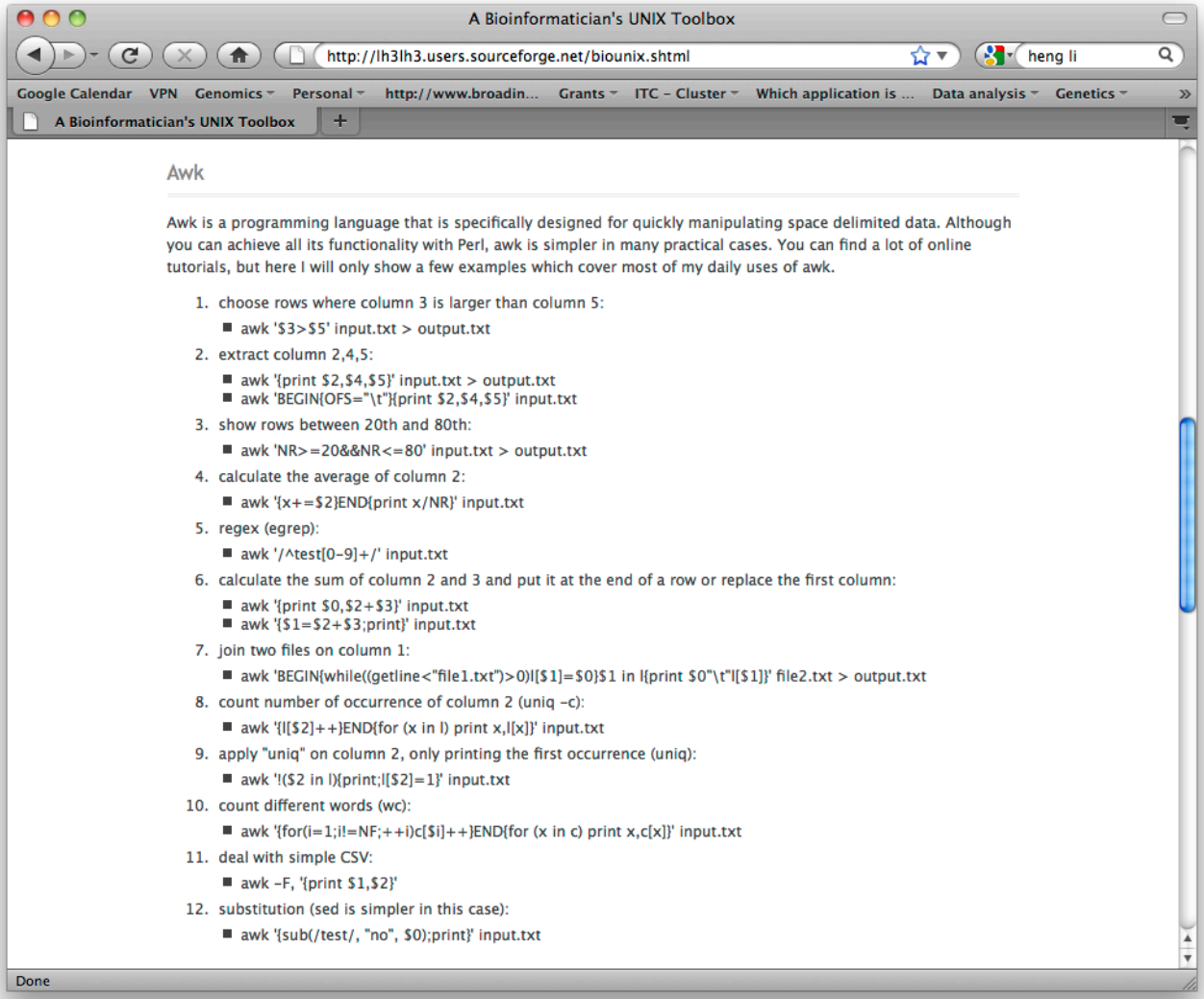
"cpg.bed"
N = 2,463

Problem: No name and no strand in the output. Let's fix it...

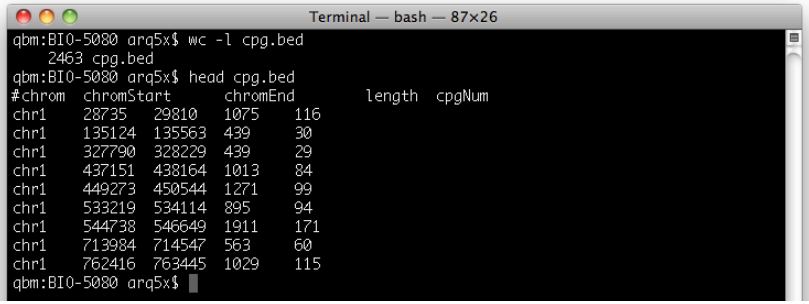
The awesome power of ***awk***

- written in 1977 by Alfred **A**ho, Peter **W**einberger, and Brian **K**ernighan
- Similar constructs as Perl ($1, $2, $3, etc.)
- Typically used for filtering, summarizing, or reorganizing files.
- "One liners", used on files or "streams"
- `awk '{print "Hello World!\n"}'` (Print "Hello World!")
- `awk '$2 >= 30' foo.txt` (Get lines in foo where 2nd col. is g.t.e. 30)
- `awk '{print $3,$2,$7,$1}' foo.txt > foo.reordered.txt`

The awesome power of ***awk***

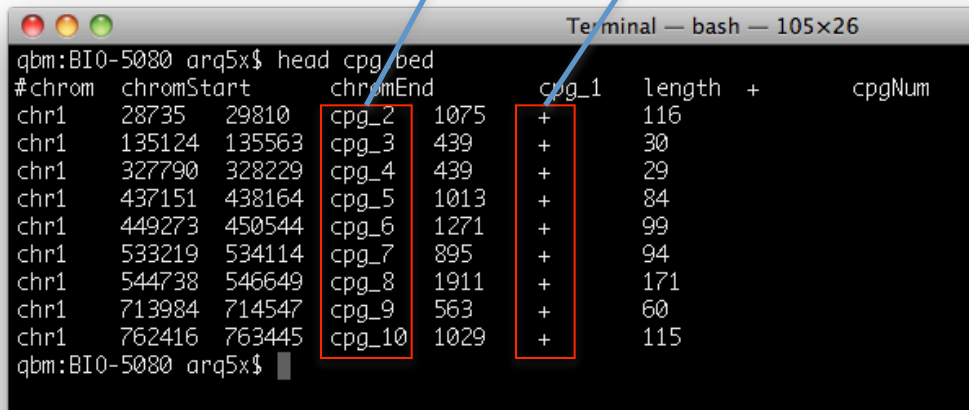


Use awk to fix cpg.bed

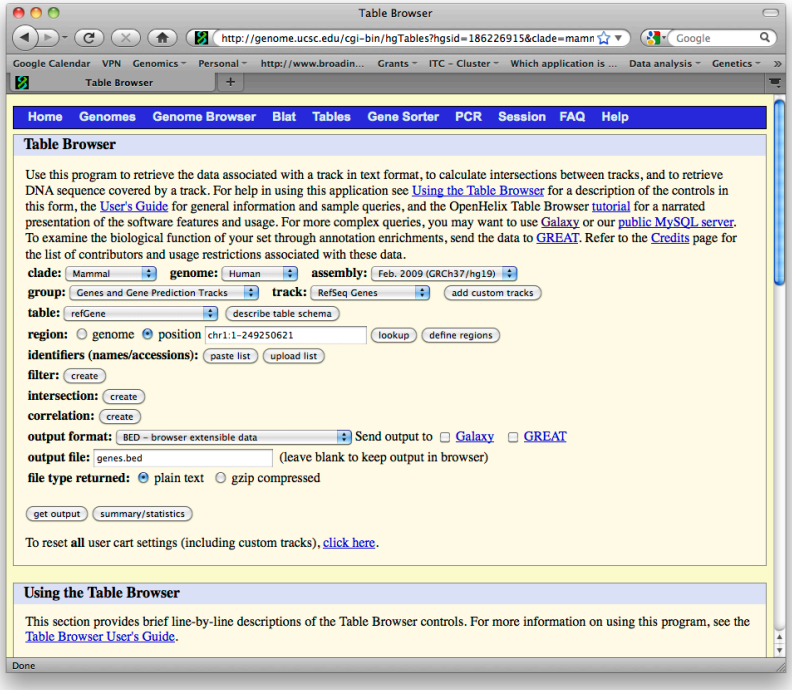


Problem: No name and no strand in the output. Let's fix it...

```
awk '{OFS="\t"; print $1,$2,$3,"cpg_"NR,$4,"+",$5}' cpg.bed > cpg.full.bed
mv cpg.full.bed cpg.bed
```



Genes via UCSC



"genes.bed"
N = 3,665

```
Terminal — bash — 192×20
qbm:BIO-5080 arq5x$ wc -l genes.bed
    3665 genes.bed
qbm:BIO-5080 arq5x$ head genes.bed
chr1    14361   29370   NR_024540       0       -       29370   29370   0       11      468,69,152,159,198,136,137,147,99,154,50,       0,608,1434,2245,2496,2871,3244,3553,3906,10376,14959,
chr1    34610   36081   NR_026818       0       -       36081   36081   0       3       564,205,361,    0,666,1110,
chr1    34610   36081   NR_026820       0       -       36081   36081   0       3       564,205,361,    0,666,1110,
chr1    69090   70008   NM_001005484    0       +       69090   70008   0       1       918,    0,
chr1    323891  328581  NR_028327       0       +       328581  328581  0       4       169,58,2500,15460,396,547,3144,
chr1    323891  328581  NR_028322       0       +       328581  328581  0       3       169,58,4143,    0,396,547,
chr1    323891  328581  NR_028325       0       +       328581  328581  0       3       169,58,4143,    0,396,547,
chr1    367658  368597  NM_001005277    0       +       367658  368597  0       1       939,    0,
chr1    367658  368597  NM_001005224    0       +       367658  368597  0       1       939,    0,
chr1    367658  368597  NM_001005221    0       +       367658  368597  0       1       939,    0,
qbm:BIO-5080 arq5x$
```

Find SNPs that have the potential to alter gene expression regulation by

Step 1: Find SNPs in CpG islands

```
# How many SNPs are there on chrom 1?
wc -l snps.bed
2064872
# Find SNPs that overlap CpG islands using intersectBed
bedtools intersect -a snps.bed -b cpg.bed -wa > snps.cpg.bed
# How many SNPs overlapped a CpG island?
wc -l snps.cpg.bed
14974
```

So, 0.7% of the SNPs were in CpG islands. Is this sensible?

Step 1: Find SNPs in CpG islands

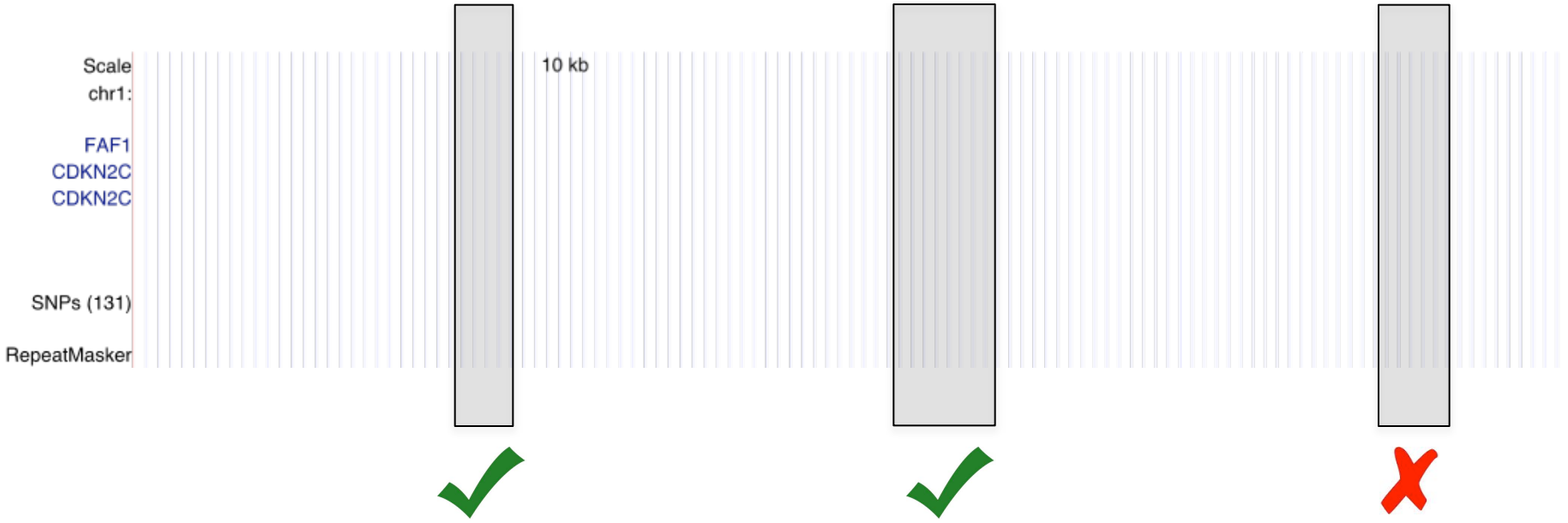
```
# How many SNPs are there on chrom 1?
wc -l snps.bed
2064872
# Find SNPs that overlap CpG islands using intersectBed
bedtools intersect -a snps.bed -b cpg.bed -wa >
snps.cpg.bed
# How many SNPs overlapped a CpG island?
wc -l snps.cpg.bed
14974
```

So, 0.7% of the SNPs were in CpG islands. Is this sensible?

```
# How much of chrom1 do the CpG islands occupy?
awk '{sum+=$5} END{print sum}' cpg.bed
1881629
```

1881629 / 249250621 = 0.75%

Step 2: Find TSS/promoter CpG islands



We want to make sure the CpG is near the start of a gene

Step 2: Find TSS/promoter CpG islands

use "window" from BEDTools

```
bedtools window -l 10000 -r 0 -sw -a genes.bed -b cpg.bed | \
          cut -f 13-19 | \
          sort | \
          uniq > cpg.gene_impact.bed
```

Step 3: Find SNPs in TSS/promoter CpG islands

use intersectBed from BEDTools

```
bedtools intersect -a snps.cpg.bed -b
cpg.gene_impact.bed > snps.cpg.gene_impact.bed
```

Step 3: Find SNPs in TSS/promoter CpG islands

use intersectBed from BEDTools

```
bedtools intersect -a snps.cpg.bed -b
cpg.gene_impact.bed > snps.cpg.gene_impact.bed
```

What types of SNPs are they? E.g., A➔G, G➔T

```
grep -v "\-" snps.cpg.gene_impact.bed | \
     cut -f 7 | \
     sort | \
     uniq -c
```

How do we know if it worked?