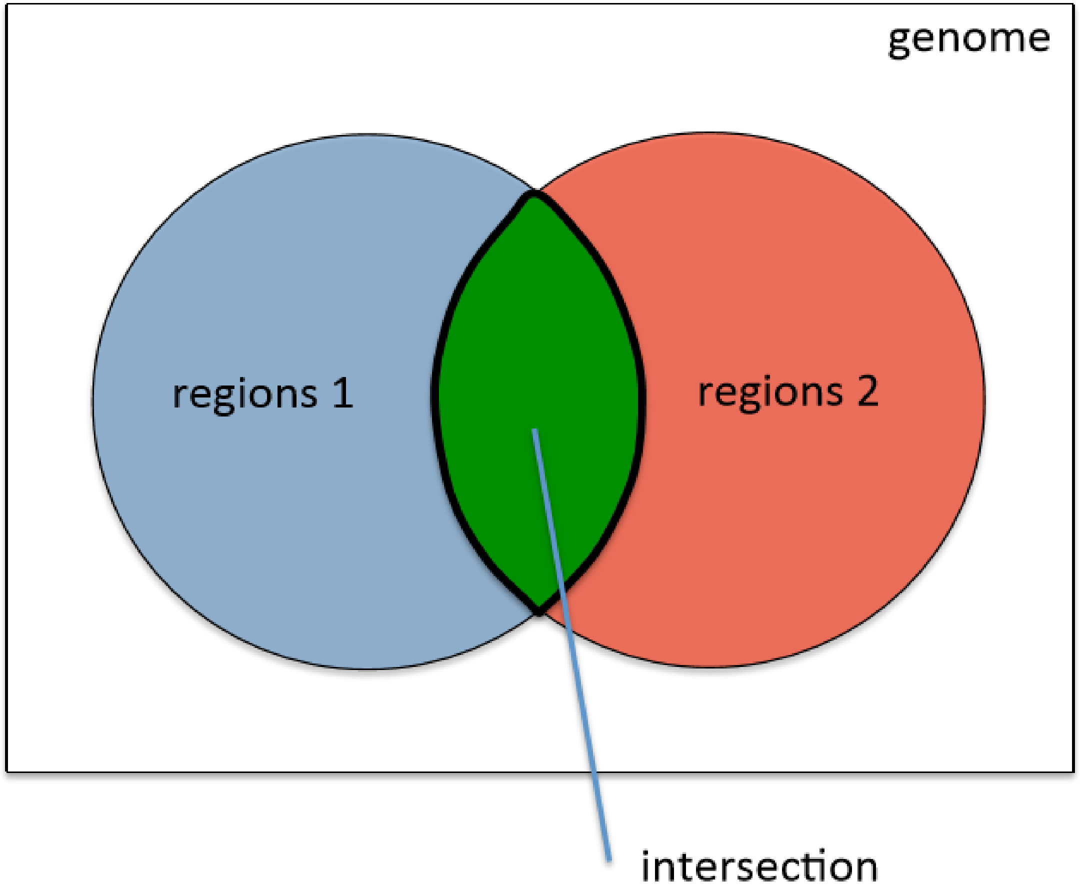
bedtools

coverage, multicov, genomecov, shuffle

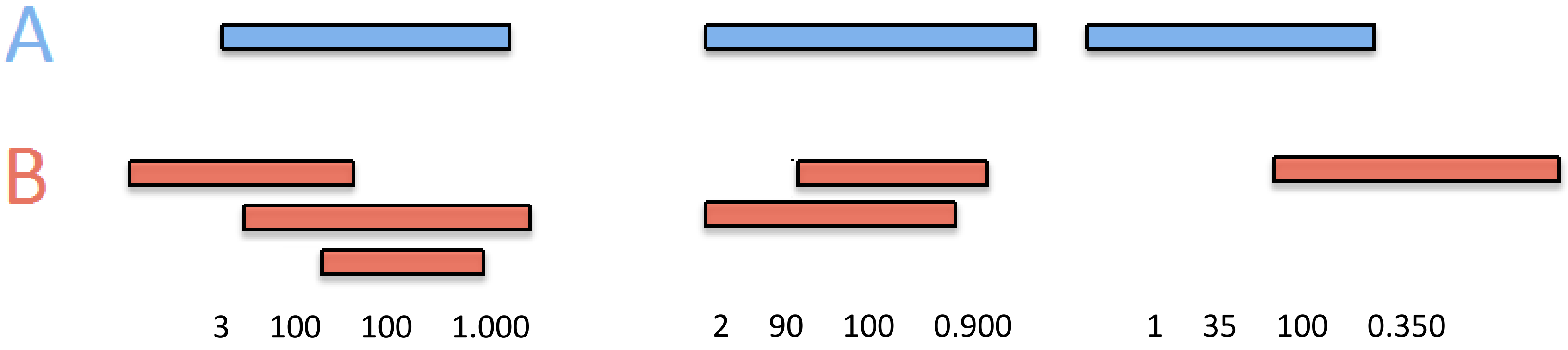
Review - bedtools intersect



What if we need more details about the intersection?

bedtools coverage

```
bedtools coverage -a <file A> -b <file B>
```



Default output for each region in A

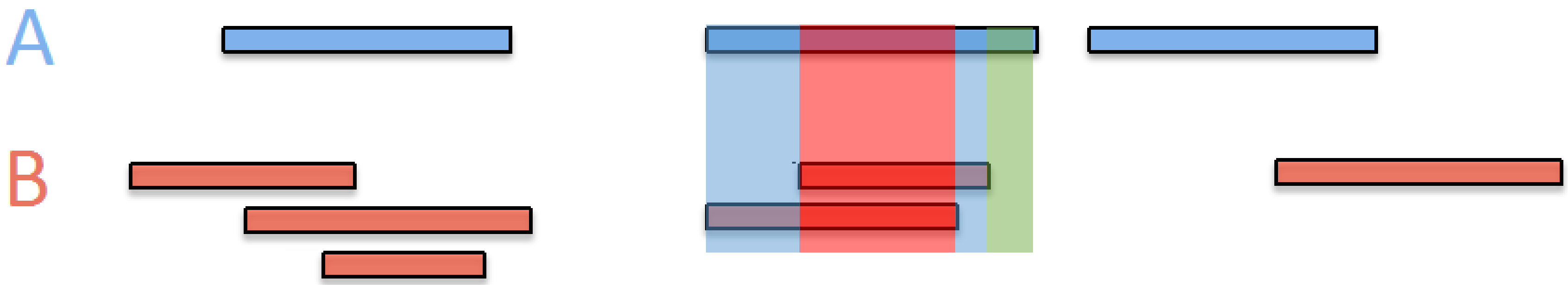
1) Number of overlapping features in B (*depth*)
2) Number of basepairs in the feature that have coverage in B
3) Total length of feature in A
4) Fraction of bases in the feature that have coverage in B (#2 / #3)

* Command line options –f and –r are the same as for *intersect*

bedtools coverage -hist

```
bedtools coverage -a <file A> -b <file B> -hist
```

- Histogram output (-hist): for each region in A, output a histogram of the percentage of basepairs at each depth



```
chr1    200    300    0    15    100    0.150
chr1    200    300    1    40    100    0.400
chr1    200    300    2    45    100    0.450
```

bedtools coverage -d

```
bedtools coverage –a <file A> –b <file B> -d
```

- For each *basepair* in each region in A, report the depth of intersection with B
- Example output:

```
chr1    0    100    1    0
chr1    0    100    2    1
chr1    0    100    3    1
chr1    0    100    4    2
…
```

Exercises

Consider the five regions listed in *short_list.bed* and the ChIP-seq peaks in *K562_CTCF_CTCF_ENCFF002CEL_chr15.bed*.

- Which of the five regions in *short_list.bed* overlaps with the *least* number of ChIP-seq peaks?
- What percentage of the first region in *short_list.bed* overlaps with more than one ChIP-seq peak? What percentage of the second region overlaps with more than one ChIP-seq peak?
- At what basepair does the first region in *short_list.bed* transition from overlapping two ChIP-seq peaks to overlapping only one?

bedtools multicov

```
bedtools multicov –bams <list of BAM files> –bed <BED file>
```

- Like *intersect –c* but with multiple BAM file inputs
- For each region in the BED file, lists the number of overlapping regions in each BAM file *separately*
- Example output:

```
a.BED                     bedtools multicov –bams bam1.bam bam2.bam –bed a.BED

Chr1     0    100         Chr1     0    100   <bam1 overlaps> <bam2 overlaps>
Chr1   100    200    →    Chr1   100    200   <bam1 overlaps> <bam2 overlaps>
Chr1   200    300         Chr1   200    300   <bam1 overlaps> <bam2 overlaps>
```

* Command line options –f and –r are the same as for *intersect*

bedtools genomecov

```
bedtools genomecov –i <input file> -g <genome file> [-max m]
```

- <input file> in BED format must be grouped by chromosome

Input.bed

```
chr1       0      100
chr2       0      100
chr1     100      200
```

sort –k 1,1 Input.bed > Input.sorted.bed

```
chr1       0      100
chr1     100      200
chr2       0      100
```

- <genome file> defines the bounds of each chromosome

human.hg19.genome

```
chr1     249250621
chr2     243199373
chr3     198022430
…
```

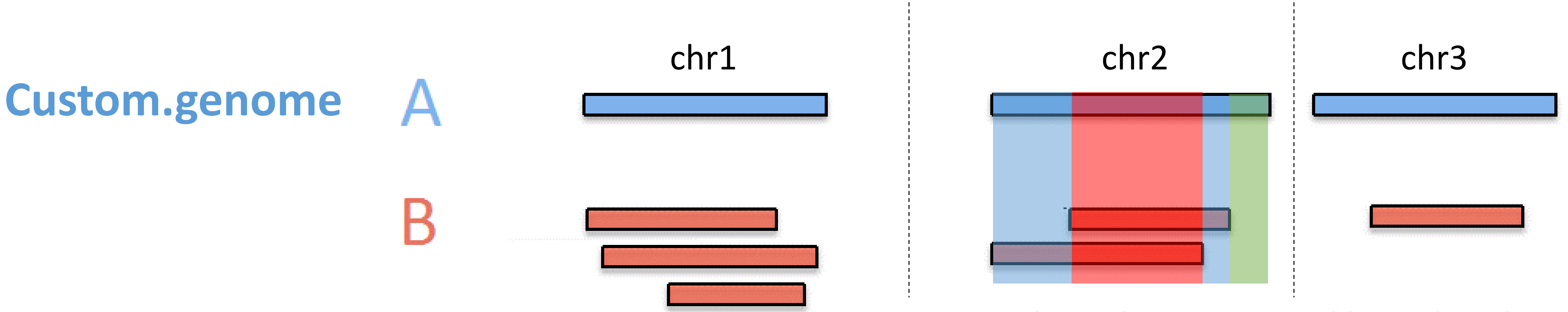
Custom.genome

```
chr1     100
chr2     100
chr3     100
```

bedtools genomecov

```
bedtools genomecov -i <input file> -g <genome file> [-max m]
```

- Like the histogram output for *coverage*, except A is the genome file, the "regions" of A are the entire chromosomes, and B is the input BED file



```
Chr2     0     15     100     0.150
Chr2     1     40     100     0.400
Chr2     2     45     100     0.450
```

-max 1

```
Chr2     0     15     100     0.150
Chr2     1     85     100     0.850
```

bedtools genomecov -d

```
bedtools genomecov –i <input file> -g <genome file> -d
```

- Same idea as *coverage –d*: basepair by basepair output
- Example output:

```
chr1        1       1
chr1        2       1
chr1        3       2
chr1        4       2
…
```

Exercise

Consider the ChIP-seq peaks in *K562_CTCF_CTCF_ENCFF002CEL_chr15.bed* and *K562_CTCF_CTCF_ENCFF002DBD_chr15.bed*

- What percentage of chromosome 15 overlaps at least one ChIP-seq peak for each file?
- How many basepairs of chromosome 15 overlap exactly one ChIP-seq peak for each file? What percentage of chr15 is this for each file?
- Do any of the first 20 basepairs of chr15 overlap with any ChIP-seq peaks in either file?

bedtools shuffle

```
bedtools shuffle –i <input file> -g <genome file>
```

- Randomly shuffle the regions in <input file> to different locations within the genome defined in <genome file>
- By default, any region can be moved *anywhere* (any location on any chromosome) and the regions can overlap with one another
- Options:

  -incl <region file>: new regions may only be placed within the regions defined in <region file>

  -excl <region file>: new regions may *not* be placed within the regions defined in <region file>

  -chrom: shuffled regions retain their original chromosome

  -noOverlapping: shuffled regions may not overlap with each other

Exercise

“We used all 711 VISTA [mouse mm10] enhancers as positive training data, and for negative training data, we created a set of 711 random regions matched to the length and chromosome distribution of the positives to represent the genomic background.”

- Given the 711 VISTA positive regions (vista.bed) and the mouse mm10 assembly genome (mm10.genome), how would you generate the list of negatives described in this methods section excerpt?
- How would you generate the same list of negatives if you wanted to make sure none overlapped with the list of known coding genes in mm10.coding.bed?

Pollard et al. *Integrating Diverse Datasets Improves Developmental Enhancer Prediction*. DOI: 10.1371/journal.pcbi.1003677