Bedtools Lab

Bedtools: example analyses

- Closest gene to a ChIP-seq peak.
- 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?
- What is the density of variants in ”windows” along the genome?

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

Basic structure of a bed file

- **Minimum 3 columns:**
  - **Chr**
  - **Start (0-based)**
  - **End**
- **Up to 9 additional columns**
- **Tab separated**

```
chr7    127471196  127472363
chr7    127472363  127473530
chr7    127473530  127474697
chr7    127474697  127475864
chr7    127475864  127477031
chr7    127477031  127478198
chr7    127478198  127479365
chr7    127479365  127480532
chr7    127480532  127481699
```

Description of the files in sandbox.bio

cpg.bed --> Genome coordinates + annotations for CpG islands or genomic intervals enriched for C and G nucleotides

exons.bed --> genome coordinates + transcript + strand information for exons in the human

gwas.bed --> genome coordinates + ID for disease-associated single nucleotide polymorphisms

hesc.chromHmm.bed --> genome coordinates + function in the genome (e.g., promoter, enhancer, etc.)

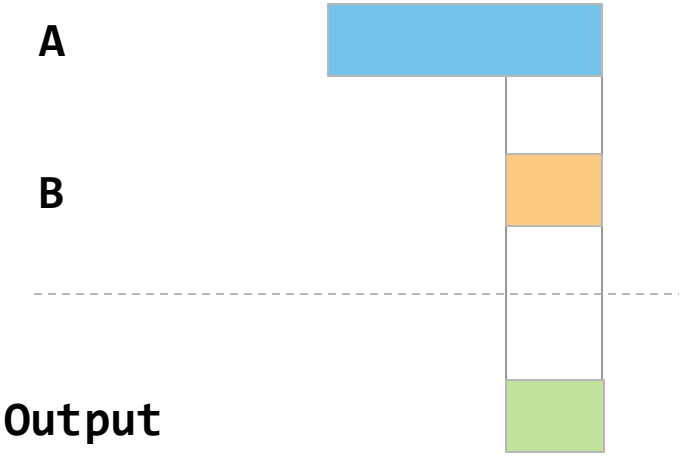
bedtools intersect

```
## Run this command:
$ bedtools

## Run bedtools with a subcommand (e.g., intersect)
$ bedtools intersect

## Note what inputs are required (scroll to the top of the output)
Usage:   bedtools intersect [OPTIONS] -a <bed/gff/vcf/bam> -b <bed/gff/vcf/bam>
```

bedtools intersect

```
$ cat intersect_A.bed
chr1  10  20
chr1  30  40
```

```
$ cat intersect_B.bed
chr1  15  20
```



Where is the intersection taking place?

```
$ bedtools intersect -a
intersect_A.bed -b intersect_B.bed
```

Stepping through bedtools intersect output

```
$ cat intersect_A.bed
chr1  10  20
chr1  30  40
```

```
$ cat intersect_B.bed
chr1  15  20
```

```
$ bedtools intersect -a intersect_A.bed -b intersect_B.bed
chr1   15     20
```

These are the portions of A intervals that overlap B intervals.

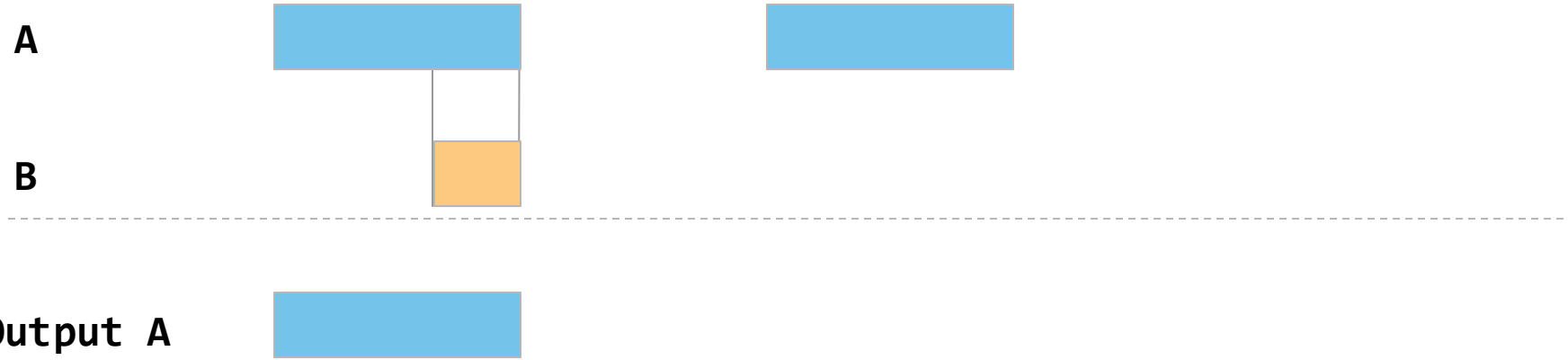
Note: These do not represent the original A intervals!!!

How can we report the original interval with bedtools?

bedtools intersect -wa (write A)

`-wa` option will return the original A interval(s) that are being overlapped

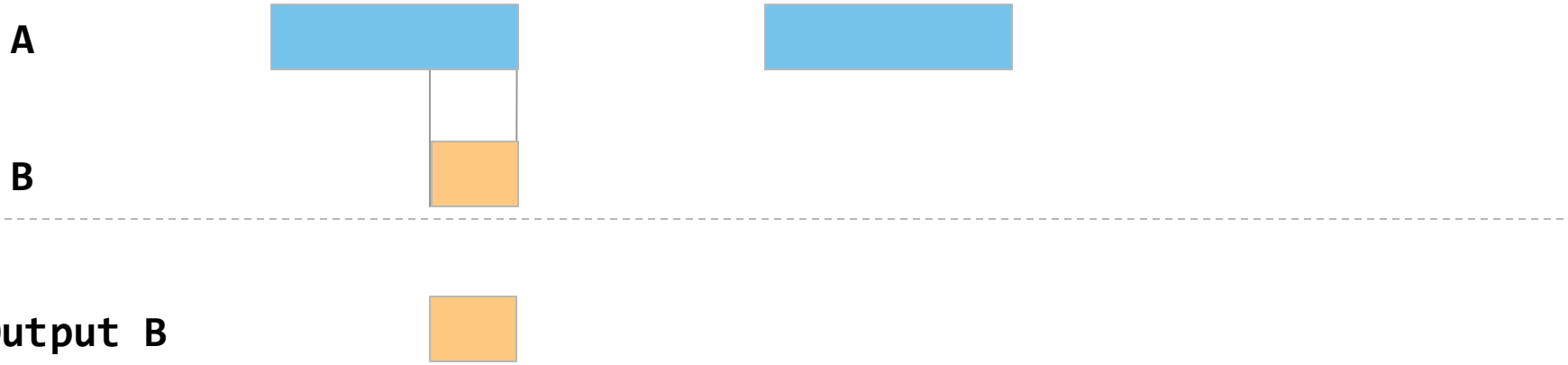
```
$ bedtools intersect -a intersect_A.bed -b intersect_B.bed -wa
```



bedtools intersect -wb (write B)

`-wb` option will return the original B interval(s) that are being overlapped

```
$ bedtools intersect -a intersect_A.bed -b intersect_B.bed -wb
```



Mini review

Explore sandbox.bio tutorial 5
Exercise 1: Identify the overlapping coordinates between CpG Islands (cpg.bed) and exons (exons.bed)

Exercise 2: Report the original exon coordinates that have CpG Islands (cpg.bed). Use **cut** to get only the exon columns.

Exercise 3: How can you report the original intervals from both A and B?

Mini review

Exercise 1: Identify the overlapping coordinates between CpG Islands (cpg.bed) and exons (exons.bed)

```
bedtools intersect -a cpg.bed -b exons.bed
```

Exercise 2: Report the original exon coordinates that have CpG Islands (cpg.bed). Use **cut** to get the exon columns.

```
bedtools intersect -a exons.bed -b cpg.bed -wa | cut -f 1-3 | head
```

```
bedtools intersect -a cpg.bed –b exons.bed -wb | cut -f 5-7 | head
```

Exercise 3: How can you report the original intervals from both A and B?

```
bedtools intersect -a exons.bed -b cpg.bed -wa -wb | head
```

What if an A interval has multiple overlapping B intervals?

```
$ cat intersect_A.bed
chr1  10  20
chr1  30  40
```

```
$ cat intersect_B.bed
chr1  15  20
chr1  17  23
```

Let’s add this file in to our intersect_B.bed file

What if an A interval has multiple overlapping B intervals?

$ cat intersect_A.bed

```
chr1  10  20
chr1  30  40
```

$ cat intersect_B.bed

```
chr1  15  20
chr1  17  23
```

$ bedtools intersect -a intersect_A.bed -b intersect_B.bed

```
chr1   15     20
chr1   17     20
```

Why are these two rows being outputted?

Did the original A interval overlap with any interval in B?

bedtools intersect -u (uniq intervals)

```
$ cat intersect_A.bed
chr1  10  20
chr1  30  40
```

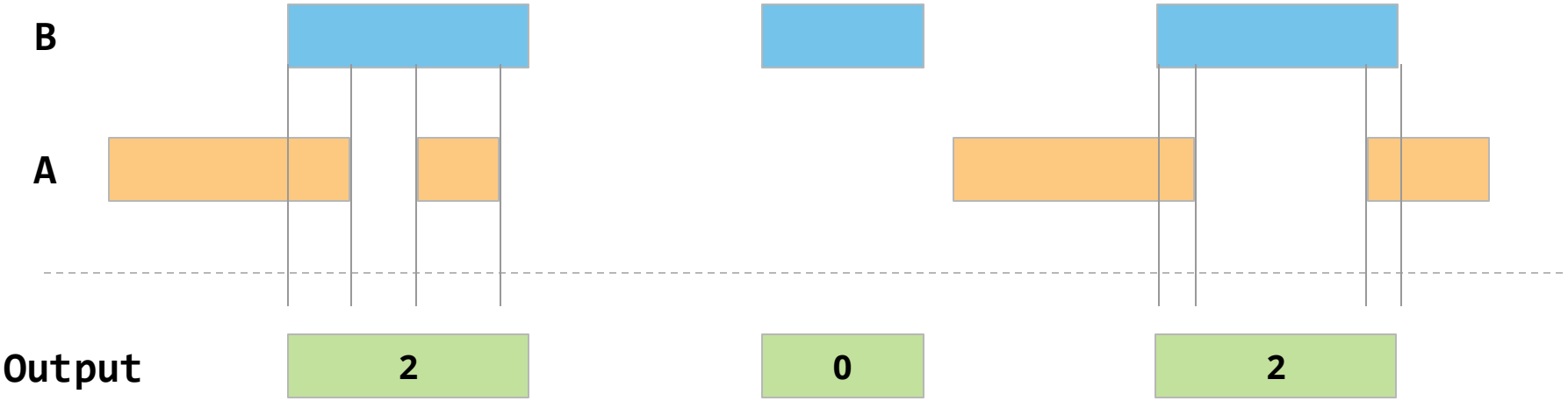
```
$ cat intersect_B.bed
chr1  15  20
chr1  17  23
```

```
$ bedtools intersect -a intersect_A.bed -b intersect_B.bed
chr1   15     20
chr1   17     20
```

```
$ bedtools intersect -a intersect_A.bed -b intersect_B.bed -u
chr1   10     20
```

Notice that **-u** returns the original intervals from the A.bed file!!

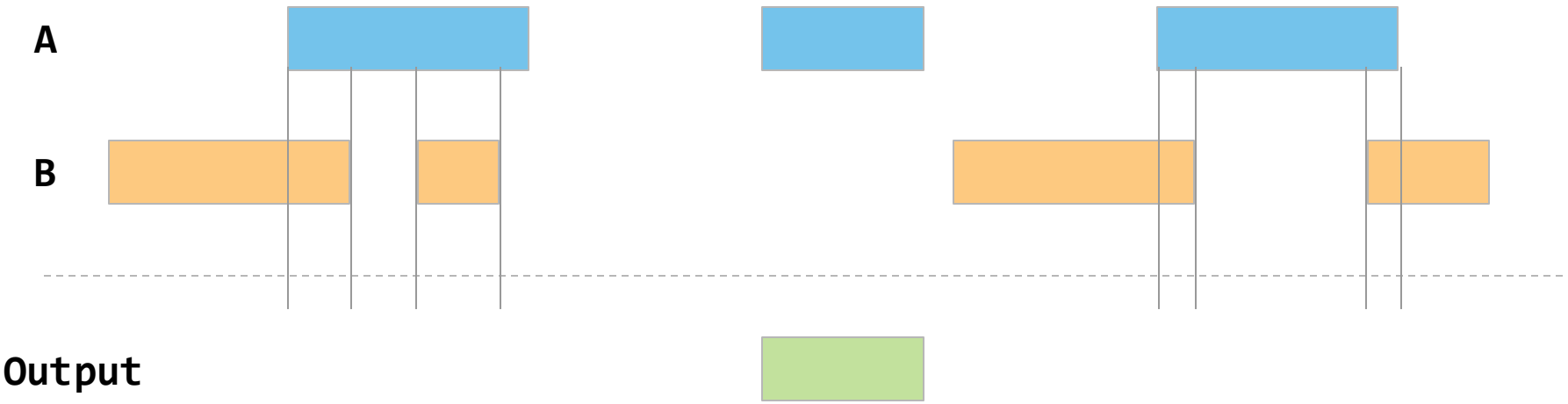
bedtools intersect -c (count overlapping features)



How many B intervals overlap with the original A intervals?

```
$ bedtools intersect -a intersect_A.bed -b intersect_B.bed -c
```

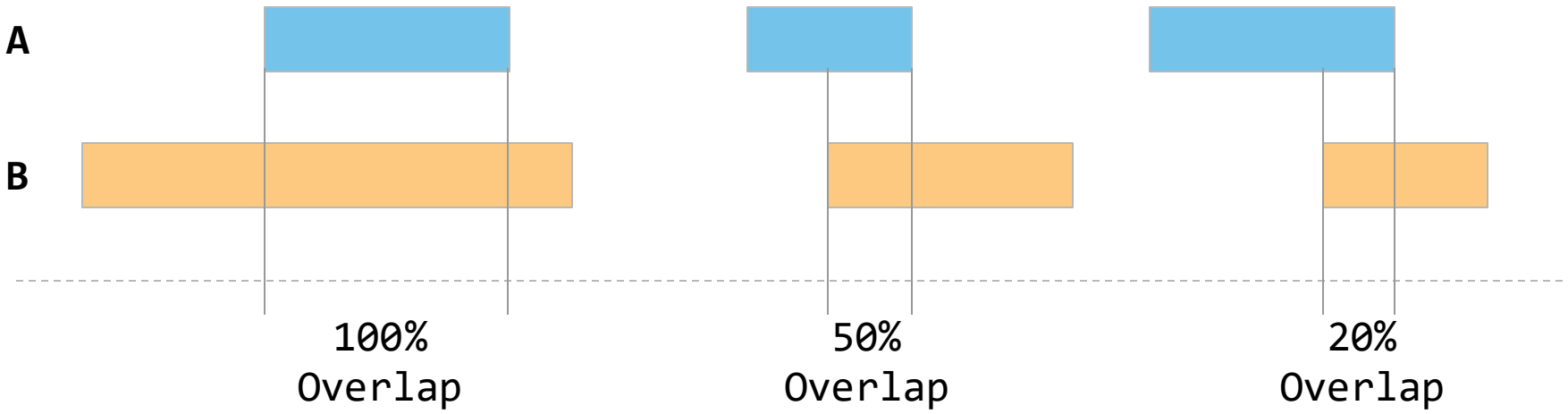
bedtools intersect -v (no overlap)



Which intervals in A do not overlap any intervals in B?

```
$ bedtools intersect -a intersect_A.bed -b intersect_B.bed -v
```

One final bedtools intersect -f (fraction)



`-f` requires ≥x% of interval A overlapped by interval B (0 < x ≤ 1)

```
$ bedtools intersect -a intersect_A.bed -b intersect_B.bed -f 0.50
```

The directionality of the intersection is specified by the **-a** and **-b** files!

Everything is in reference to the **-a** file!

Sorting bed files

- **Some bedtools works with unsorted bed files**
- **But working with sorted bed files will greatly improve speed!**

```
## Sorting a bed file:
$ sort -k1,1 -k2,2n exons.bed > sorted_exons.bed

## After sorting a bed file, may use the -sorted option:
$ bedtools intersect -a sorted_A.bed -b sorted_B.bed -sorted
```

Explore **exercise 6-8** on sandbox.bio

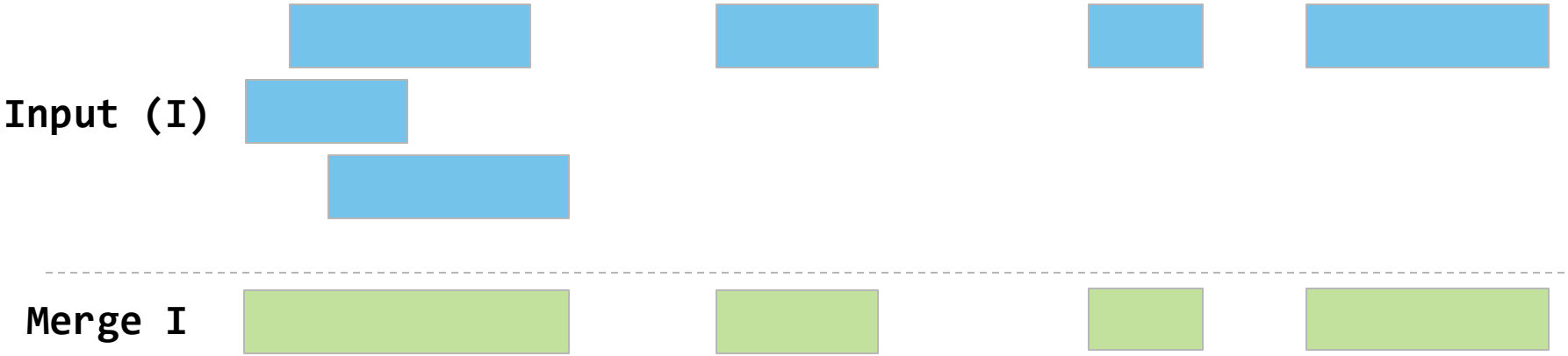
What do each of the bedtools intersect options do?

Command	**Purpose**	**Return original A interval?**	**Return original B interval?**
`bedtools intersect`	Report overlapping regions of A and B intervals	No	No
`bedtools intersect -wa`	Identify A itnervals overlapping B	Yes	No
`bedtools intersect -wb`	Identify B intervals overlapping A	Yes	Yes
`bedtools intersect -u`	Unique A intervals overlapping ≥1 B interval	Yes	No
`bedtools intersect -c`	Counts overlapping intervals from B with A	Yes	No
`bedtools intersect -v`	A intervals without B overlap	Yes	No
`Bedtools intersect -f`	Filters intervals to overlap ≥ -f value	No (without -wa)	No (without -wb)

intersect is the workhorse of the bedtools suite of tools...

What about other useful subcommands?

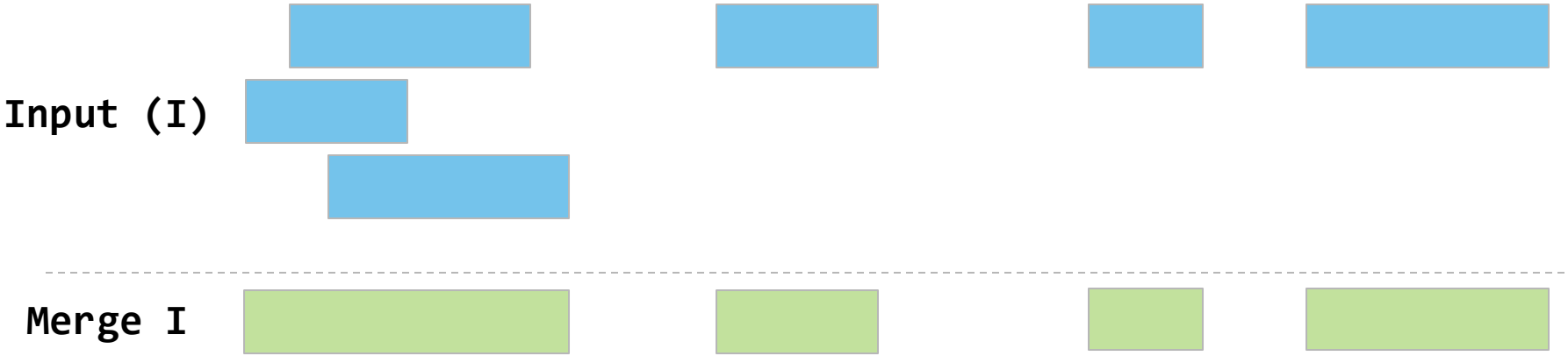
bedtools merge



Why didn’t the below command work?

```
$ bedtools merge exons.bed
```

bedtools merge



All fixed!

```
$ bedtools merge -i exons.bed
```

```
Usage:   bedtools merge [OPTIONS] -i <bed/gff/vcf>
```

bedtools merge

```
$ cat merge_file.bed
chr1  100  200
chr1  180  250
chr1  250  500
chr1  501  1000
```

Why was the 3rd row merged?

But not 4th row?

```
$ bedtools merge -i merge_file.bed
chr1  100  500
chr1  501  1000
```

bedtools merge

```
$ cat merge_file.bed
chr1  100  200
chr1  180  250
chr1  250  500
chr1  501  1000
```

Why was the 3rd row merged?

But not 4th row?

0-based coordinates!

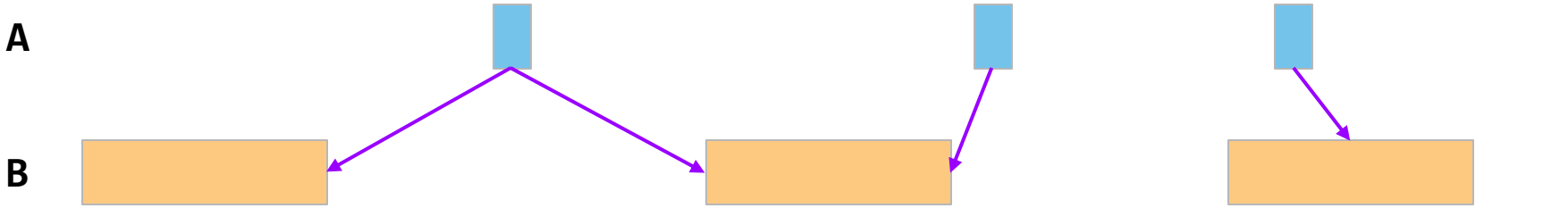
bedtools merge exercises 9-11

NOTE:

- Merge requires bed file pre-sorted
- How to check if a file is sorted:

```
## Sorting a bed file:
$ sort -k1,1 -k2,2n exons.bed > sorted_exons.bed

## Check if a file was already sorted:
$ diff exons.bed sorted_exons.bed

## If nothing shows up, two files are identical!
```

And finally... bedtools closest



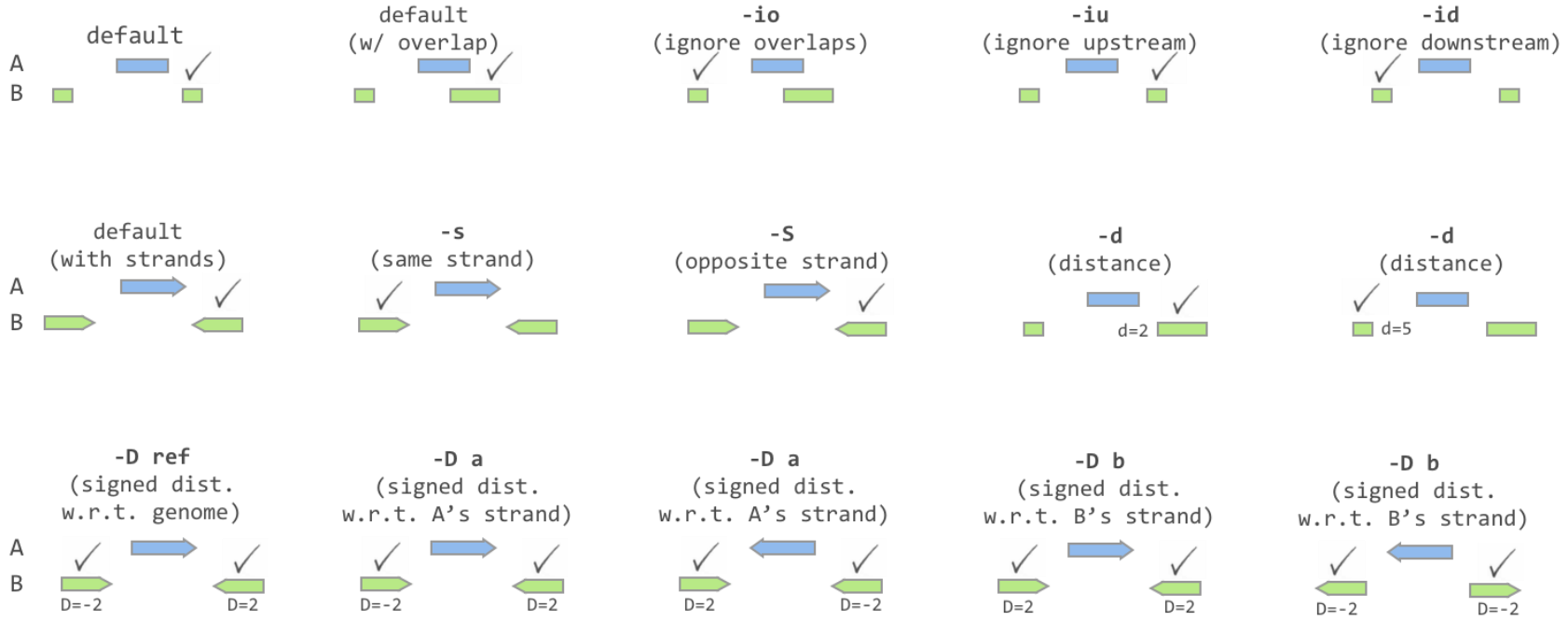

```
$ bedtools closest -a gwas.bed -b exons.bed | head -n 5
```

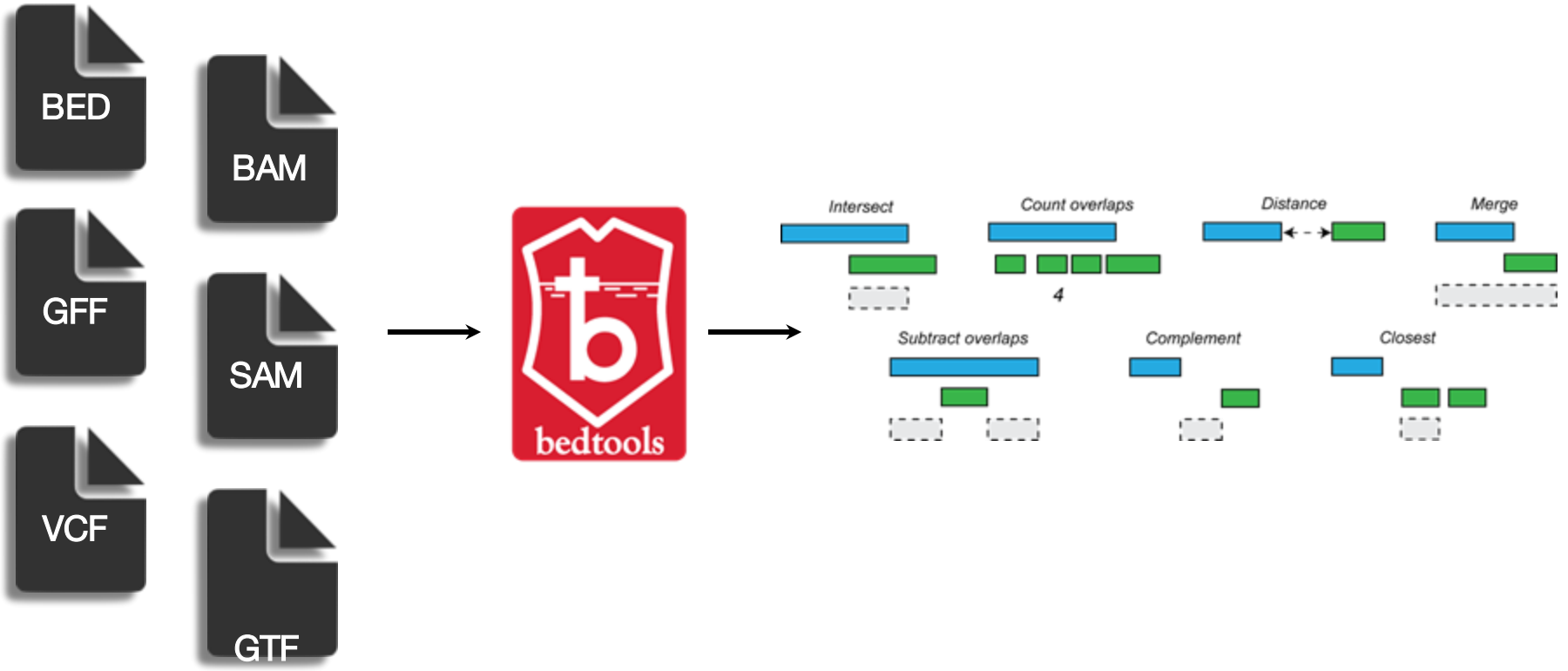
```
chr1    1005805 1005806 rs3934834       chr1    957580  957842  NM_198576_exon_1_0_chr1_957581_f 0       +
chr1    1079197 1079198 rs11260603      chr1    957580  957842  NM_198576_exon_1_0_chr1_957581_f 0       +
chr1    1247493 1247494 rs12103 chr1    957580  957842  NM_198576_exon_1_0_chr1_957581_f0+
chr1    2069171 2069172 rs425277        chr1    957580  957842  NM_198576_exon_1_0_chr1_957581_f 0       +
chr1    2069680 2069681 rs3753242       chr1    957580  957842  NM_198576_exon_1_0_chr1_957581_f 0       +
```

GWAS SNPs

Closest exon(s)

bedtools closest




BED
BAM
GFF
SAM
VCF
GTF
bedtools
Intersect
Count overlaps
4
Distance
Merge
Subtract overlaps
Complement
Closest