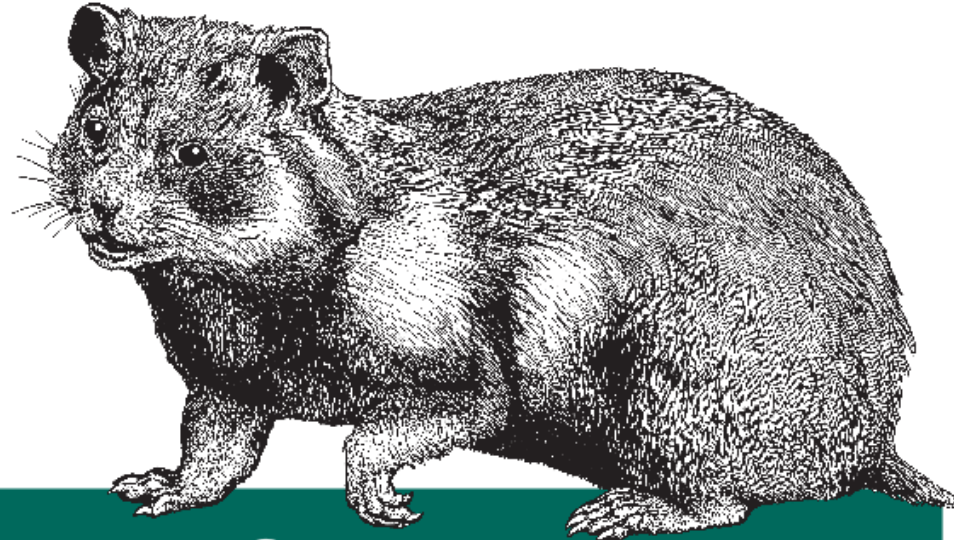


Work with range data

Bioinformatics Data Skills

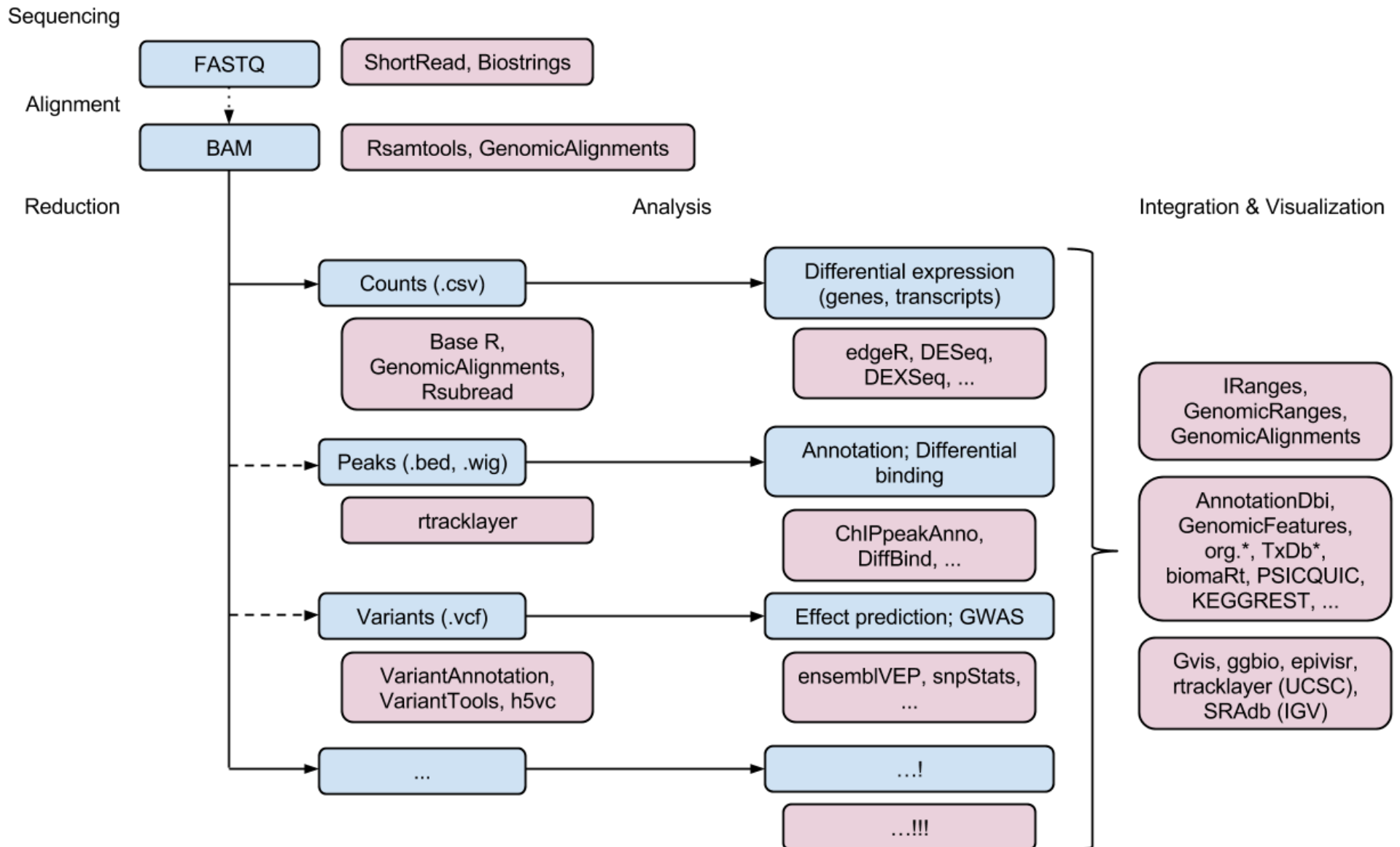


Bioinformatics Data Skills

REPRODUCIBLE AND ROBUST RESEARCH WITH OPEN SOURCE TOOLS

<https://github.com/vsbuffalo/bds-files>

Bioconductor



http://bioconductor.org/packages/release/BiocViews.html#___Software

Today's topic

- **GenomicRanges**: package dealing with genomic intervals
 - built on more general package **IRanges**
 - provide a rich collection of functions for genomic interval operations
- **GenomicFeatures**: package for transcript centric genomic annotations

Operation type	Functions
Arithmetic	shift, resize, restrict, flank
Set	intersect, union, setdiff, gaps
Summary	coverage, reduce, disjoint
Comparison	findOverlaps, nearest, order

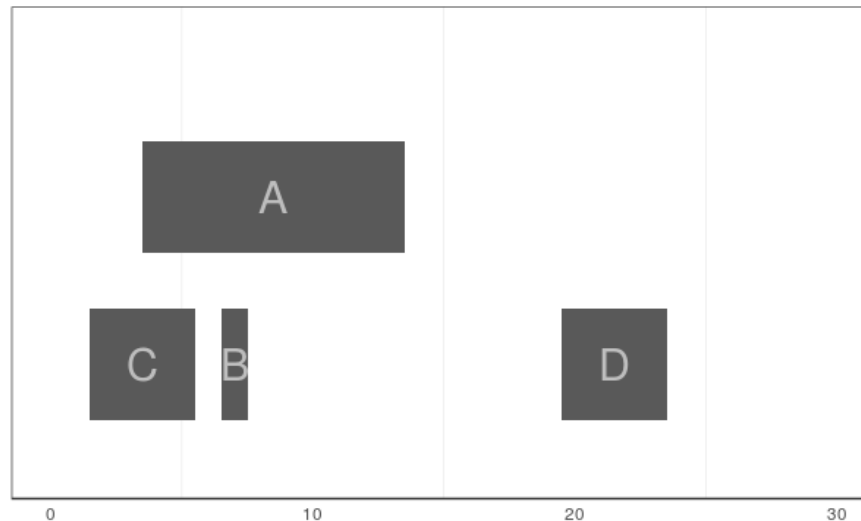
Storing Generic Ranges with IRanges

- IRanges(start=NULL, end=NULL, width=NULL, names=NULL)

```
``{r}
#The IRanges package is a dependency of the GenomicRanges package we installed earlier
library(IRanges)
#The ranges we create with the IRanges package are called IRanges objects
rng <- IRanges(start=c(4, 7, 2, 20), end=c(13, 7, 5, 23))
names(rng) <- LETTERS[1:4]
rng
``
```

IRanges object with 4 ranges and 0 metadata columns:

	start	end	width
	<integer>	<integer>	<integer>
A	4	13	10
B	7	7	1
C	2	5	4
D	20	23	4

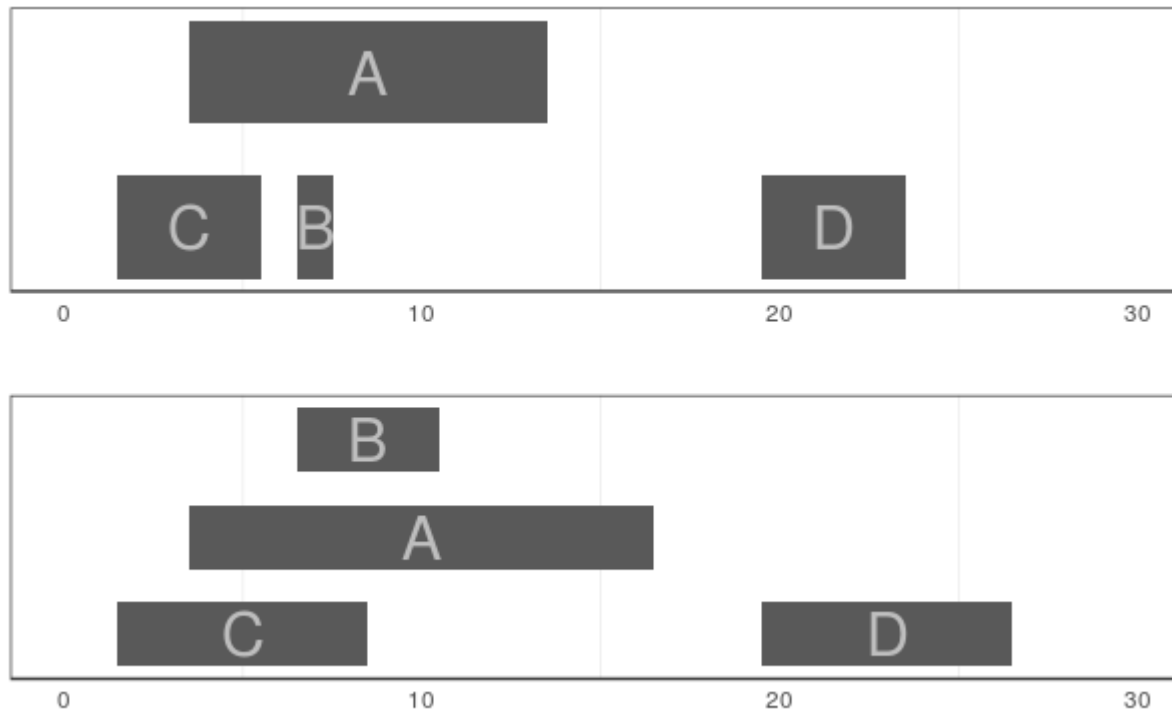


Basic Range Operations

- `start(rng)`
- `end(rng)`
- `width(rng)`
- `range(rng)`
- `rng[1:3]`
- `rng["C"]`
- `rng[start(rng) > 5]`
- `m <- IRanges(start = 3, width = 5)`
- `n <- IRanges(start = 6, width = 8)`
- `c(m, n)`

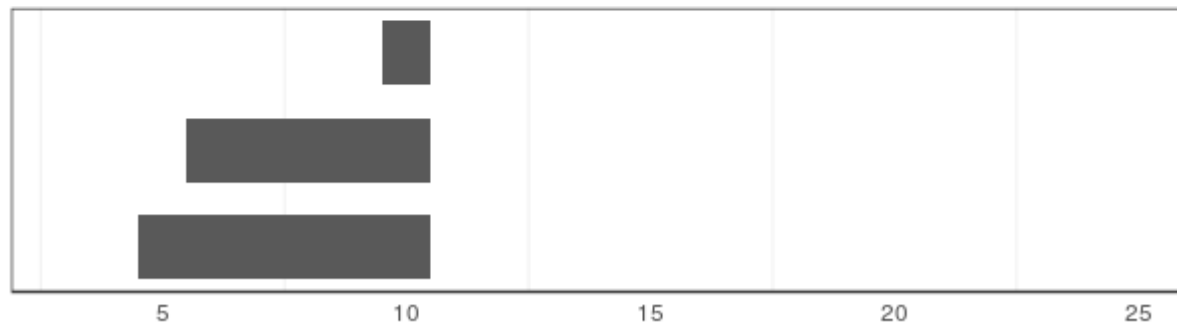
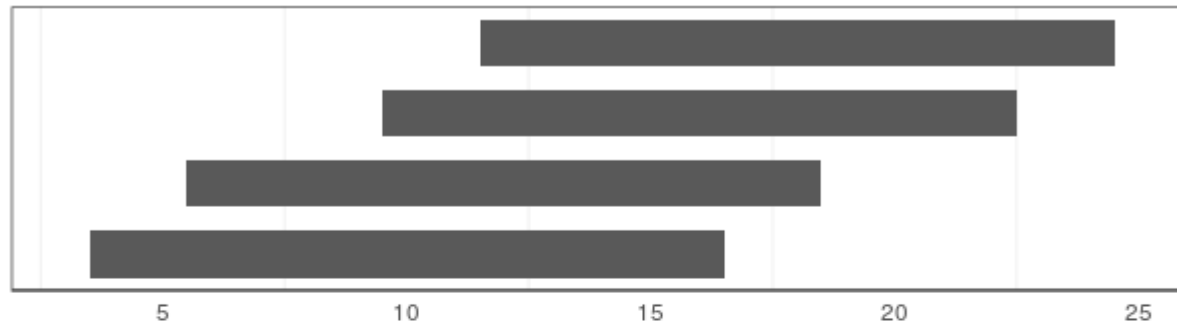
Basic Range Operations

- With IRanges objects, addition (subtraction) will grow (shrink) a range symmetrically by the value added (subtracted) to it
- `end(rng) <- end(rng) + 3`



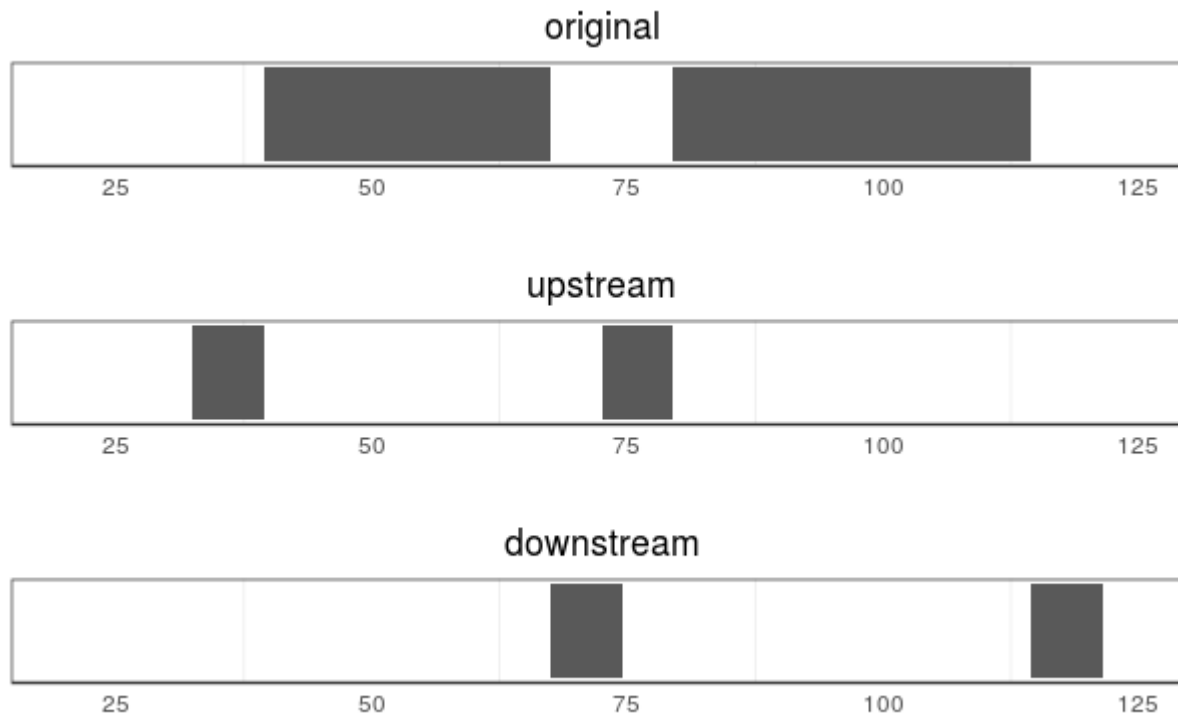
Basic Range Operations

- `restrict()` cuts a set of ranges such that they fall inside of a certain bound
- `y <- IRanges(start = c(4, 6, 10, 12), width = 13)`
- `y.r <- restrict(y, start = 5, end = 10)`



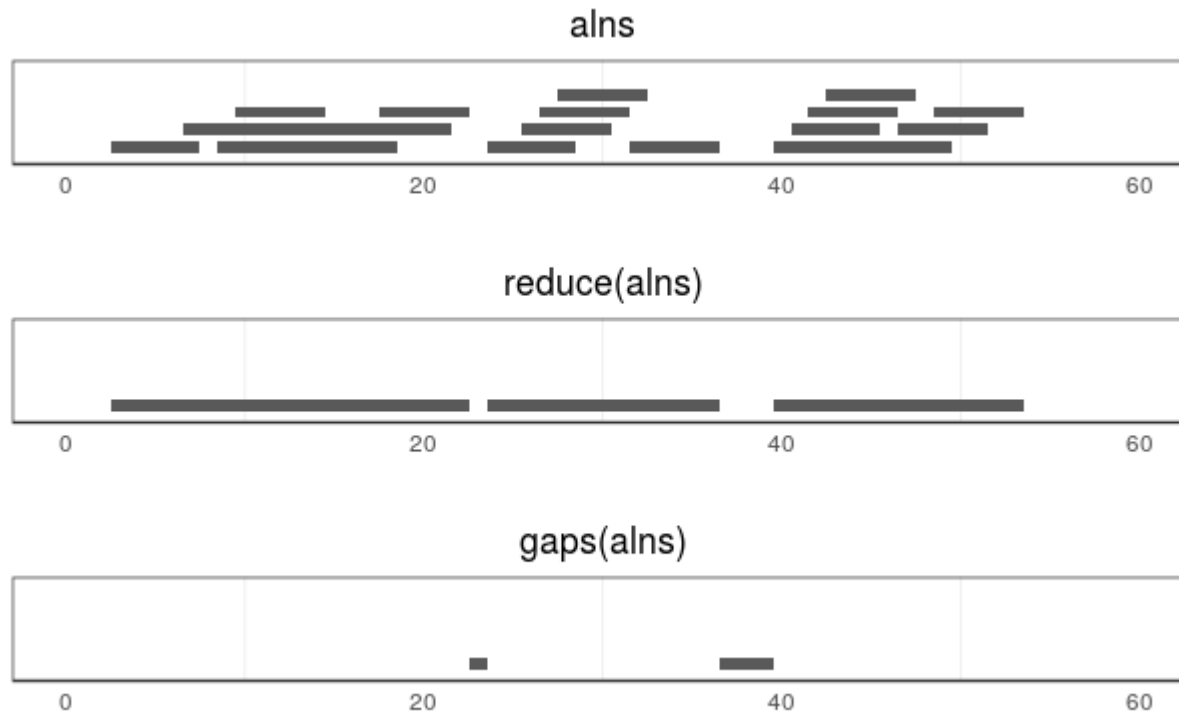
Basic Range Operations

- `flank()` returns the regions that flank (are on the side of) each range in an `IRanges` object.
- `flank()` is useful in creating ranges upstream and downstream of protein coding genes that could contain promoter sequences.
- `x <- IRanges(start=c(40, 80), end=c(67, 114))`
- `flank(x, width = 7)`
- `flank(x, width = 7, start = FALSE)`



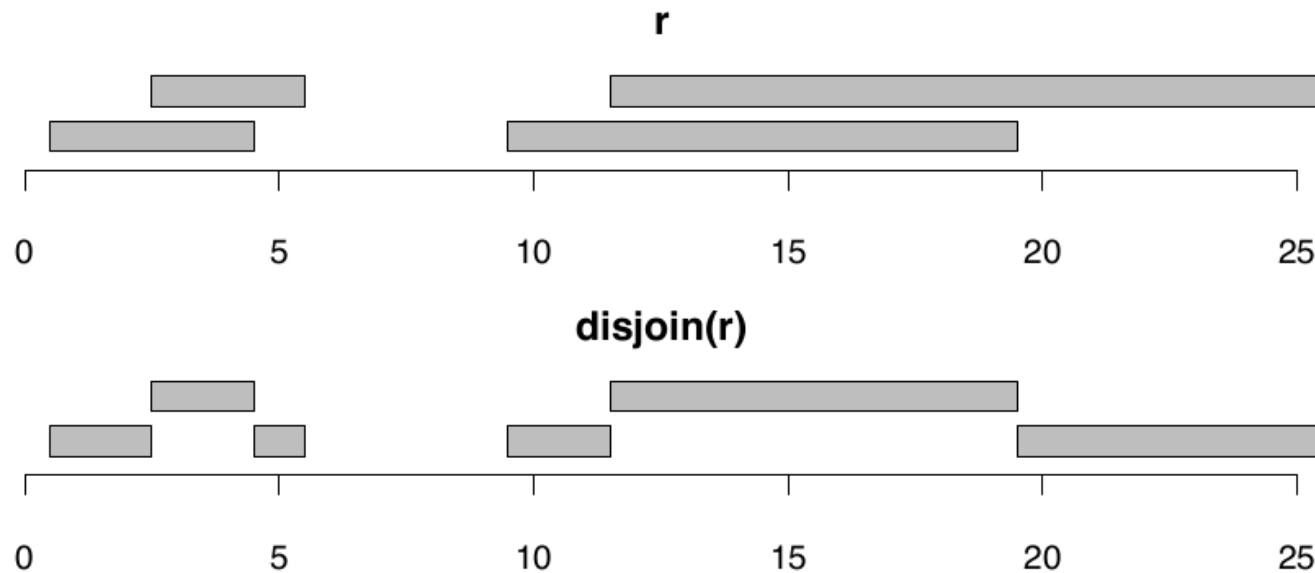
Basic Range Operations

- `reduce()` operation takes a set of possibly overlapping ranges and reduces them to a set of nonoverlapping ranges that cover the same positions
- `gaps()` returns the gaps (uncovered portions) between ranges



Basic Range Operations

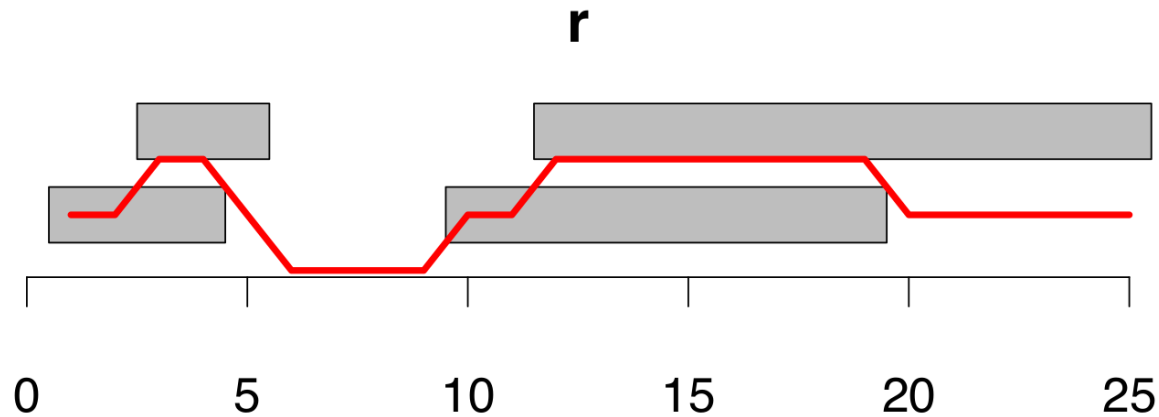
- Return a set of non-overlapping ranges satisfying:
 - (1) the union of results is the same as the union of the inputs.
 - (2) for every range in the result, its overlapping pattern with the input is constant.



Basic Range Operations

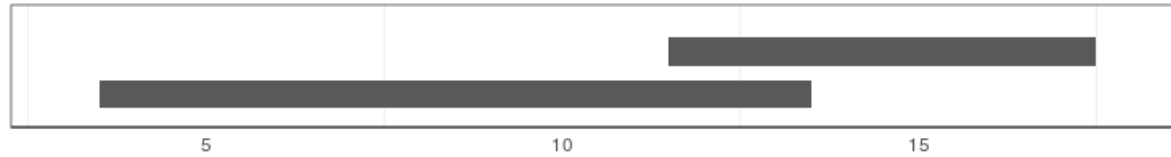
- `coverage()`

Compute the coverage depth by the input ranges of each position.



Basic Range Operations

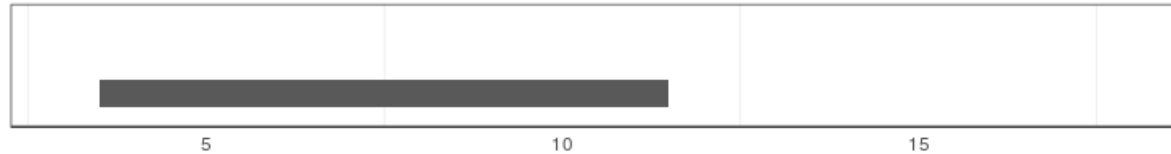
a, b



`intersect(a, b)`



`setdiff(a, b)`



`setdiff(b, a)`



`union(a, b)`

