Introduction to R/Bioconductor

NGS analysis for gene regulation and epigenomics

Physalia 2021

Jacques Serizay

R main classes

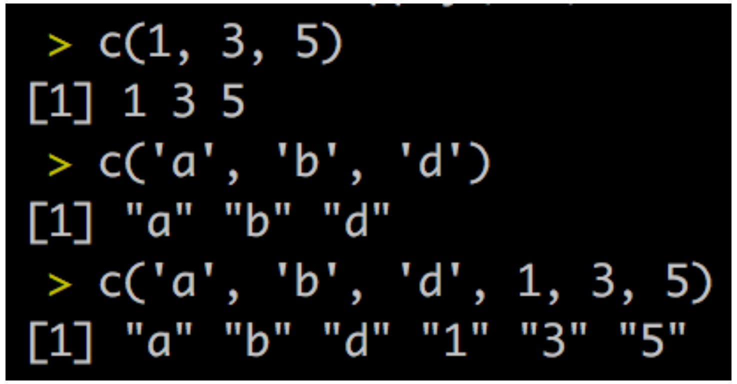
- Vectors

R main classes

- Vectors
  - Defined with c() function

R main classes

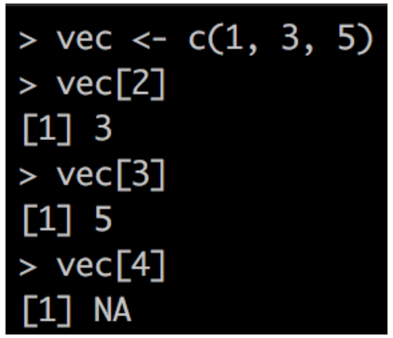
- Vectors
  - Defined with c() function
  - All the elements must be from the same class



```
> c(1, 3, 5)
[1] 1 3 5
> c('a', 'b', 'd')
[1] "a" "b" "d"
> c('a', 'b', 'd', 1, 3, 5)
[1] "a" "b" "d" "1" "3" "5"
```

R main classes

- Vectors
  - Defined with c() function
  - All the elements must be from the same class
  - Can be subset with […]



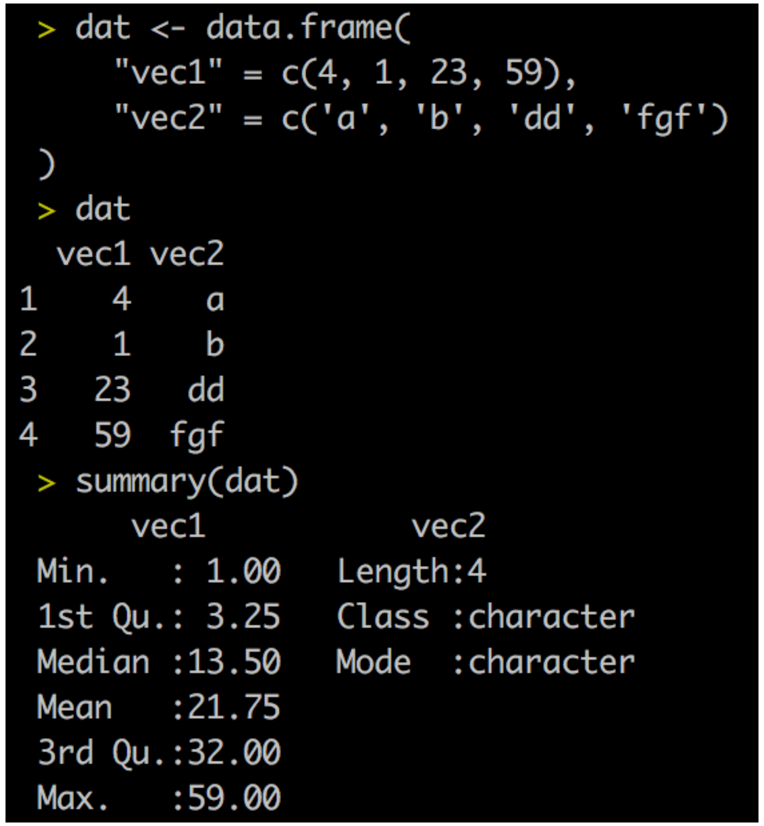
```
> vec <- c(1, 3, 5)
> vec[2]
[1] 3
> vec[3]
[1] 5
> vec[4]
[1] NA
```

R main classes

- data.frame

R main classes

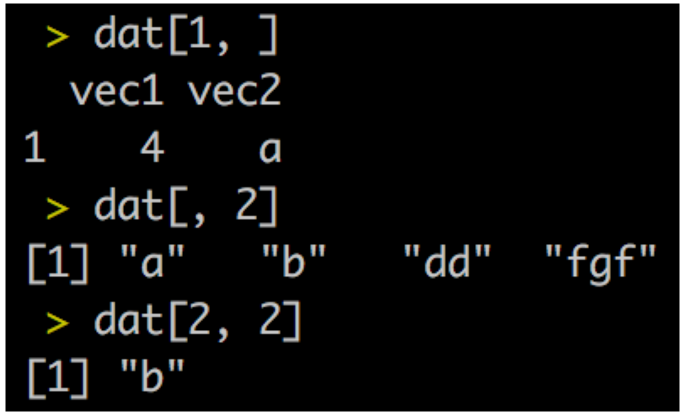
- data.frame
  - Tabular shape
  - Defined with data.frame()



```
 > dat <- data.frame(
     "vec1" = c(4, 1, 23, 59),
     "vec2" = c('a', 'b', 'dd', 'fgf')
 )
 > dat
  vec1 vec2
1    4    a
2    1    b
3   23   dd
4   59  fgf
 > summary(dat)
      vec1           vec2          
 Min.   : 1.00   Length:4          
 1st Qu.: 3.25   Class :character  
 Median :13.50   Mode  :character  
 Mean   :21.75                     
 3rd Qu.:32.00                     
 Max.   :59.00                     
```

R main classes

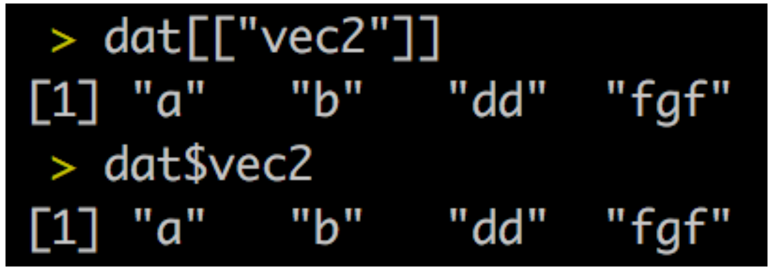
- data.frame
  - Tabular shape
  - Defined with data.frame()
  - Subset with […, …]



```
> dat[1, ]
  vec1 vec2
1    4    a
> dat[, 2]
[1] "a"   "b"   "dd"  "fgf"
> dat[2, 2]
[1] "b"
```

R main classes

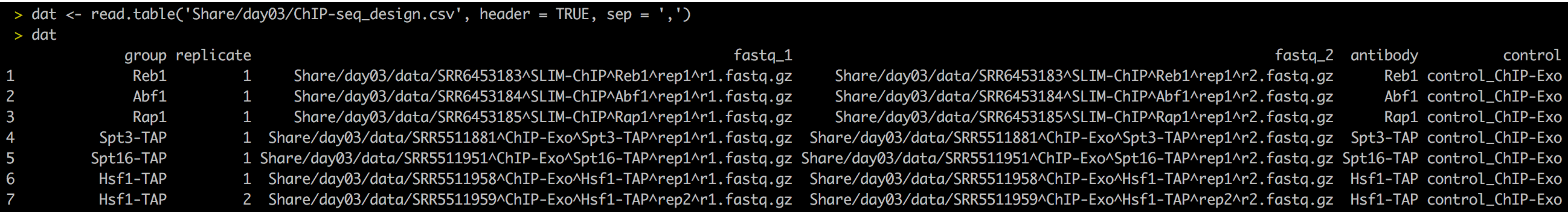
- data.frame
  - Tabular shape
  - Defined with data.frame()
  - Subset with […, …]
  - Columns can be accessed to with [[…]] or $



```
 > dat[["vec2"]]
[1] "a"   "b"   "dd"  "fgf"
 > dat$vec2
[1] "a"   "b"   "dd"  "fgf"
```

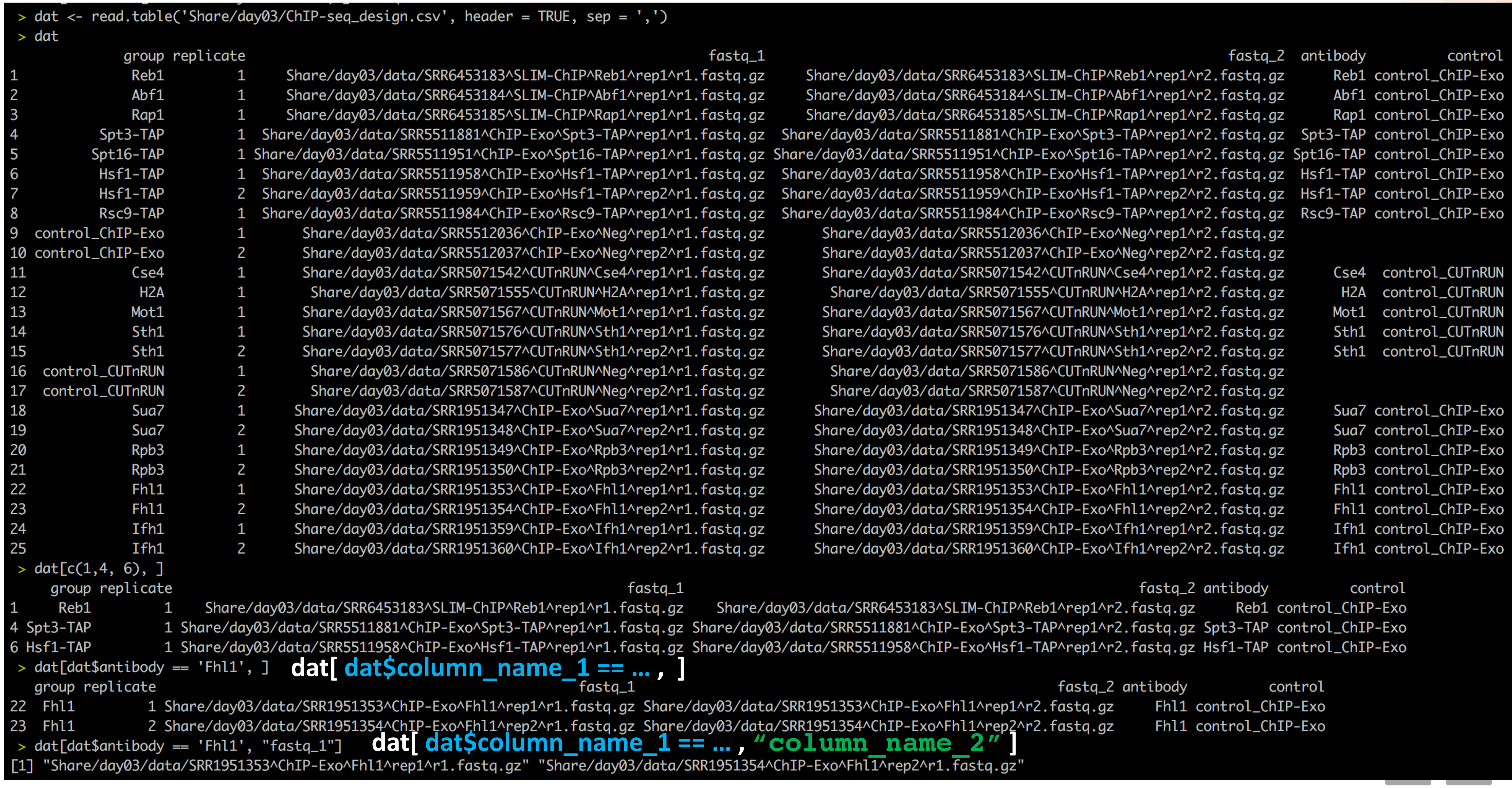
R main classes

- Data frames can be created from tabular files using read.table()



```
> dat <- read.table('Share/day03/ChIP-seq_design.csv', header = TRUE, sep = ',')
> dat
         group replicate                                                fastq_1                                                fastq_2  antibody          control
1         Reb1         1     Share/day03/data/SRR6453183^SLIM-ChIP^Reb1^rep1^r1.fastq.gz     Share/day03/data/SRR6453183^SLIM-ChIP^Reb1^rep1^r2.fastq.gz      Reb1 control_ChIP-Exo
2         Abf1         1     Share/day03/data/SRR6453184^SLIM-ChIP^Abf1^rep1^r1.fastq.gz     Share/day03/data/SRR6453184^SLIM-ChIP^Abf1^rep1^r2.fastq.gz      Abf1 control_ChIP-Exo
3         Rap1         1     Share/day03/data/SRR6453185^SLIM-ChIP^Rap1^rep1^r1.fastq.gz     Share/day03/data/SRR6453185^SLIM-ChIP^Rap1^rep1^r2.fastq.gz      Rap1 control_ChIP-Exo
4     Spt3-TAP         1  Share/day03/data/SRR5511881^ChIP-Exo^Spt3-TAP^rep1^r1.fastq.gz  Share/day03/data/SRR5511881^ChIP-Exo^Spt3-TAP^rep1^r2.fastq.gz  Spt3-TAP control_ChIP-Exo
5    Spt16-TAP         1 Share/day03/data/SRR5511951^ChIP-Exo^Spt16-TAP^rep1^r1.fastq.gz Share/day03/data/SRR5511951^ChIP-Exo^Spt16-TAP^rep1^r2.fastq.gz Spt16-TAP control_ChIP-Exo
6     Hsf1-TAP         1  Share/day03/data/SRR5511958^ChIP-Exo^Hsf1-TAP^rep1^r1.fastq.gz  Share/day03/data/SRR5511958^ChIP-Exo^Hsf1-TAP^rep1^r2.fastq.gz  Hsf1-TAP control_ChIP-Exo
7     Hsf1-TAP         2  Share/day03/data/SRR5511959^ChIP-Exo^Hsf1-TAP^rep2^r1.fastq.gz  Share/day03/data/SRR5511959^ChIP-Exo^Hsf1-TAP^rep2^r2.fastq.gz  Hsf1-TAP control_ChIP-Exo
```

R main classes



```
> dat <- read.table('Share/day03/ChIP-seq_design.csv', header = TRUE, sep = ',')
> dat
             group replicate                                                 fastq_1                                                 fastq_2 antibody          control
1             Reb1         1    Share/day03/data/SRR6453183^SLIM-ChIP^Reb1^rep1^r1.fastq.gz    Share/day03/data/SRR6453183^SLIM-ChIP^Reb1^rep1^r2.fastq.gz     Reb1 control_ChIP-Exo
2             Abf1         1    Share/day03/data/SRR6453184^SLIM-ChIP^Abf1^rep1^r1.fastq.gz    Share/day03/data/SRR6453184^SLIM-ChIP^Abf1^rep1^r2.fastq.gz     Abf1 control_ChIP-Exo
3             Rap1         1    Share/day03/data/SRR6453185^SLIM-ChIP^Rap1^rep1^r1.fastq.gz    Share/day03/data/SRR6453185^SLIM-ChIP^Rap1^rep1^r2.fastq.gz     Rap1 control_ChIP-Exo
4         Spt3-TAP         1  Share/day03/data/SRR5511881^ChIP-Exo^Spt3-TAP^rep1^r1.fastq.gz  Share/day03/data/SRR5511881^ChIP-Exo^Spt3-TAP^rep1^r2.fastq.gz Spt3-TAP control_ChIP-Exo
5        Spt16-TAP         1 Share/day03/data/SRR5511951^ChIP-Exo^Spt16-TAP^rep1^r1.fastq.gz Share/day03/data/SRR5511951^ChIP-Exo^Spt16-TAP^rep1^r2.fastq.gz Spt16-TAP control_ChIP-Exo
6         Hsf1-TAP         1  Share/day03/data/SRR5511958^ChIP-Exo^Hsf1-TAP^rep1^r1.fastq.gz  Share/day03/data/SRR5511958^ChIP-Exo^Hsf1-TAP^rep1^r2.fastq.gz Hsf1-TAP control_ChIP-Exo
7         Hsf1-TAP         2  Share/day03/data/SRR5511959^ChIP-Exo^Hsf1-TAP^rep2^r1.fastq.gz  Share/day03/data/SRR5511959^ChIP-Exo^Hsf1-TAP^rep2^r2.fastq.gz Hsf1-TAP control_ChIP-Exo
8         Rsc9-TAP         1  Share/day03/data/SRR5511984^ChIP-Exo^Rsc9-TAP^rep1^r1.fastq.gz  Share/day03/data/SRR5511984^ChIP-Exo^Rsc9-TAP^rep1^r2.fastq.gz Rsc9-TAP control_ChIP-Exo
9  control_ChIP-Exo         1       Share/day03/data/SRR5512036^ChIP-Exo^Neg^rep1^r1.fastq.gz       Share/day03/data/SRR5512036^ChIP-Exo^Neg^rep1^r2.fastq.gz
10 control_ChIP-Exo         2       Share/day03/data/SRR5512037^ChIP-Exo^Neg^rep2^r1.fastq.gz       Share/day03/data/SRR5512037^ChIP-Exo^Neg^rep2^r2.fastq.gz
11            Cse4         1       Share/day03/data/SRR5071542^CUTnRUN^Cse4^rep1^r1.fastq.gz       Share/day03/data/SRR5071542^CUTnRUN^Cse4^rep1^r2.fastq.gz     Cse4  control_CUTnRUN
12             H2A         1        Share/day03/data/SRR5071555^CUTnRUN^H2A^rep1^r1.fastq.gz        Share/day03/data/SRR5071555^CUTnRUN^H2A^rep1^r2.fastq.gz      H2A  control_CUTnRUN
13            Mot1         1       Share/day03/data/SRR5071567^CUTnRUN^Mot1^rep1^r1.fastq.gz       Share/day03/data/SRR5071567^CUTnRUN^Mot1^rep1^r2.fastq.gz     Mot1  control_CUTnRUN
14            Sth1         1       Share/day03/data/SRR5071576^CUTnRUN^Sth1^rep1^r1.fastq.gz       Share/day03/data/SRR5071576^CUTnRUN^Sth1^rep1^r2.fastq.gz     Sth1  control_CUTnRUN
15            Sth1         2       Share/day03/data/SRR5071577^CUTnRUN^Sth1^rep2^r1.fastq.gz       Share/day03/data/SRR5071577^CUTnRUN^Sth1^rep2^r2.fastq.gz     Sth1  control_CUTnRUN
16  control_CUTnRUN         1        Share/day03/data/SRR5071586^CUTnRUN^Neg^rep1^r1.fastq.gz        Share/day03/data/SRR5071586^CUTnRUN^Neg^rep1^r2.fastq.gz
17  control_CUTnRUN         2        Share/day03/data/SRR5071587^CUTnRUN^Neg^rep2^r1.fastq.gz        Share/day03/data/SRR5071587^CUTnRUN^Neg^rep2^r2.fastq.gz
18            Sua7         1      Share/day03/data/SRR1951347^ChIP-Exo^Sua7^rep1^r1.fastq.gz      Share/day03/data/SRR1951347^ChIP-Exo^Sua7^rep1^r2.fastq.gz     Sua7 control_ChIP-Exo
19            Sua7         2      Share/day03/data/SRR1951348^ChIP-Exo^Sua7^rep2^r1.fastq.gz      Share/day03/data/SRR1951348^ChIP-Exo^Sua7^rep2^r2.fastq.gz     Sua7 control_ChIP-Exo
20            Rpb3         1      Share/day03/data/SRR1951349^ChIP-Exo^Rpb3^rep1^r1.fastq.gz      Share/day03/data/SRR1951349^ChIP-Exo^Rpb3^rep1^r2.fastq.gz     Rpb3 control_ChIP-Exo
21            Rpb3         2      Share/day03/data/SRR1951350^ChIP-Exo^Rpb3^rep2^r1.fastq.gz      Share/day03/data/SRR1951350^ChIP-Exo^Rpb3^rep2^r2.fastq.gz     Rpb3 control_ChIP-Exo
22            Fhl1         1      Share/day03/data/SRR1951353^ChIP-Exo^Fhl1^rep1^r1.fastq.gz      Share/day03/data/SRR1951353^ChIP-Exo^Fhl1^rep1^r2.fastq.gz     Fhl1 control_ChIP-Exo
23            Fhl1         2      Share/day03/data/SRR1951354^ChIP-Exo^Fhl1^rep2^r1.fastq.gz      Share/day03/data/SRR1951354^ChIP-Exo^Fhl1^rep2^r2.fastq.gz     Fhl1 control_ChIP-Exo
24            Ifh1         1      Share/day03/data/SRR1951359^ChIP-Exo^Ifh1^rep1^r1.fastq.gz      Share/day03/data/SRR1951359^ChIP-Exo^Ifh1^rep1^r2.fastq.gz     Ifh1 control_ChIP-Exo
25            Ifh1         2      Share/day03/data/SRR1951360^ChIP-Exo^Ifh1^rep2^r1.fastq.gz      Share/day03/data/SRR1951360^ChIP-Exo^Ifh1^rep2^r2.fastq.gz     Ifh1 control_ChIP-Exo
> dat[c(1,4, 6), ]
     group replicate                                                fastq_1                                                fastq_2 antibody          control
1     Reb1         1    Share/day03/data/SRR6453183^SLIM-ChIP^Reb1^rep1^r1.fastq.gz    Share/day03/data/SRR6453183^SLIM-ChIP^Reb1^rep1^r2.fastq.gz     Reb1 control_ChIP-Exo
4 Spt3-TAP         1 Share/day03/data/SRR5511881^ChIP-Exo^Spt3-TAP^rep1^r1.fastq.gz Share/day03/data/SRR5511881^ChIP-Exo^Spt3-TAP^rep1^r2.fastq.gz Spt3-TAP control_ChIP-Exo
6 Hsf1-TAP         1 Share/day03/data/SRR5511958^ChIP-Exo^Hsf1-TAP^rep1^r1.fastq.gz Share/day03/data/SRR5511958^ChIP-Exo^Hsf1-TAP^rep1^r2.fastq.gz Hsf1-TAP control_ChIP-Exo
> dat[dat$antibody == 'Fhl1', ]   dat[ dat$column_name_1 == … , ]
   group replicate                                            fastq_1                                            fastq_2 antibody          control
22  Fhl1         1 Share/day03/data/SRR1951353^ChIP-Exo^Fhl1^rep1^r1.fastq.gz Share/day03/data/SRR1951353^ChIP-Exo^Fhl1^rep1^r2.fastq.gz     Fhl1 control_ChIP-Exo
23  Fhl1         2 Share/day03/data/SRR1951354^ChIP-Exo^Fhl1^rep2^r1.fastq.gz Share/day03/data/SRR1951354^ChIP-Exo^Fhl1^rep2^r2.fastq.gz     Fhl1 control_ChIP-Exo
> dat[dat$antibody == 'Fhl1', "fastq_1"]   dat[ dat$column_name_1 == … , "column_name_2" ]
[1] "Share/day03/data/SRR1951353^ChIP-Exo^Fhl1^rep1^r1.fastq.gz" "Share/day03/data/SRR1951354^ChIP-Exo^Fhl1^rep2^r1.fastq.gz"
```

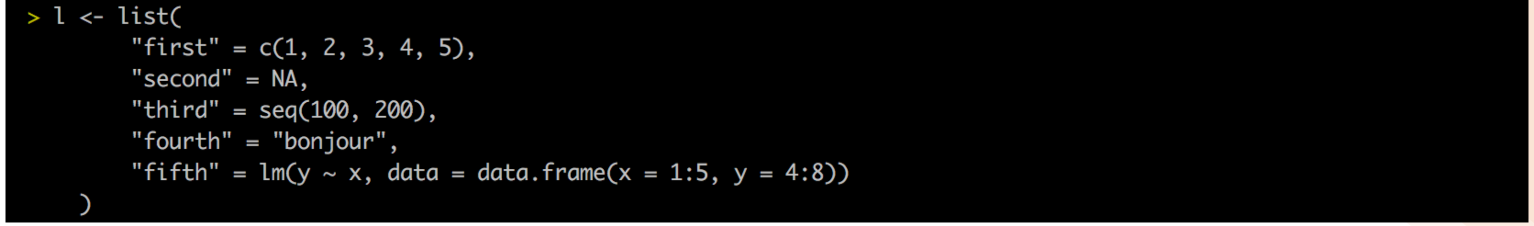
2020/01/13

Jacques Serizay

R main classes

- Lists
  - Defined with list() function
  - Each element can be whatever object you want
  - Each element can be named

R main classes

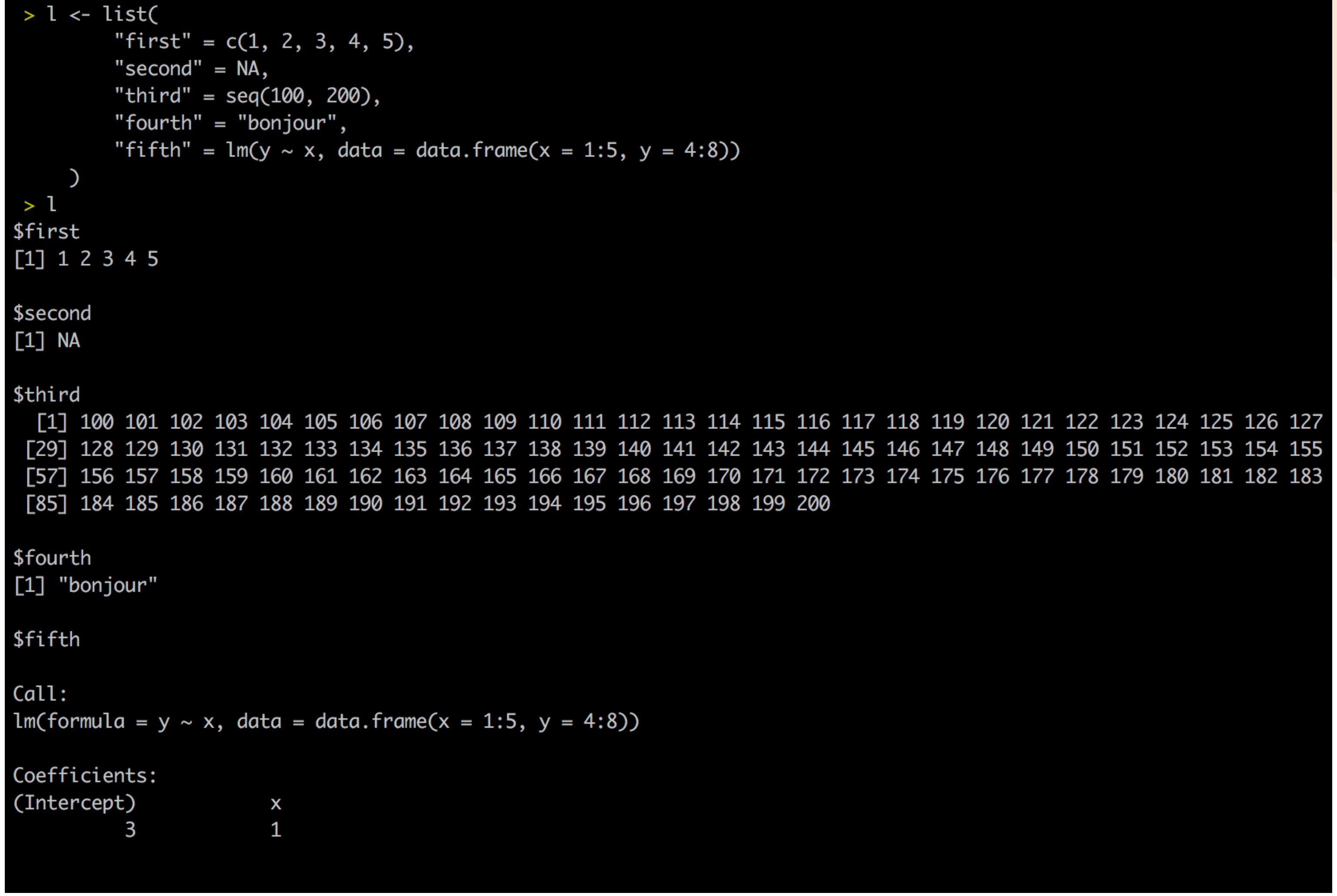


```
> l <- list(
        "first" = c(1, 2, 3, 4, 5),
        "second" = NA,
        "third" = seq(100, 200),
        "fourth" = "bonjour",
        "fifth" = lm(y ~ x, data = data.frame(x = 1:5, y = 4:8))
    )
```

- Lists

R main classes

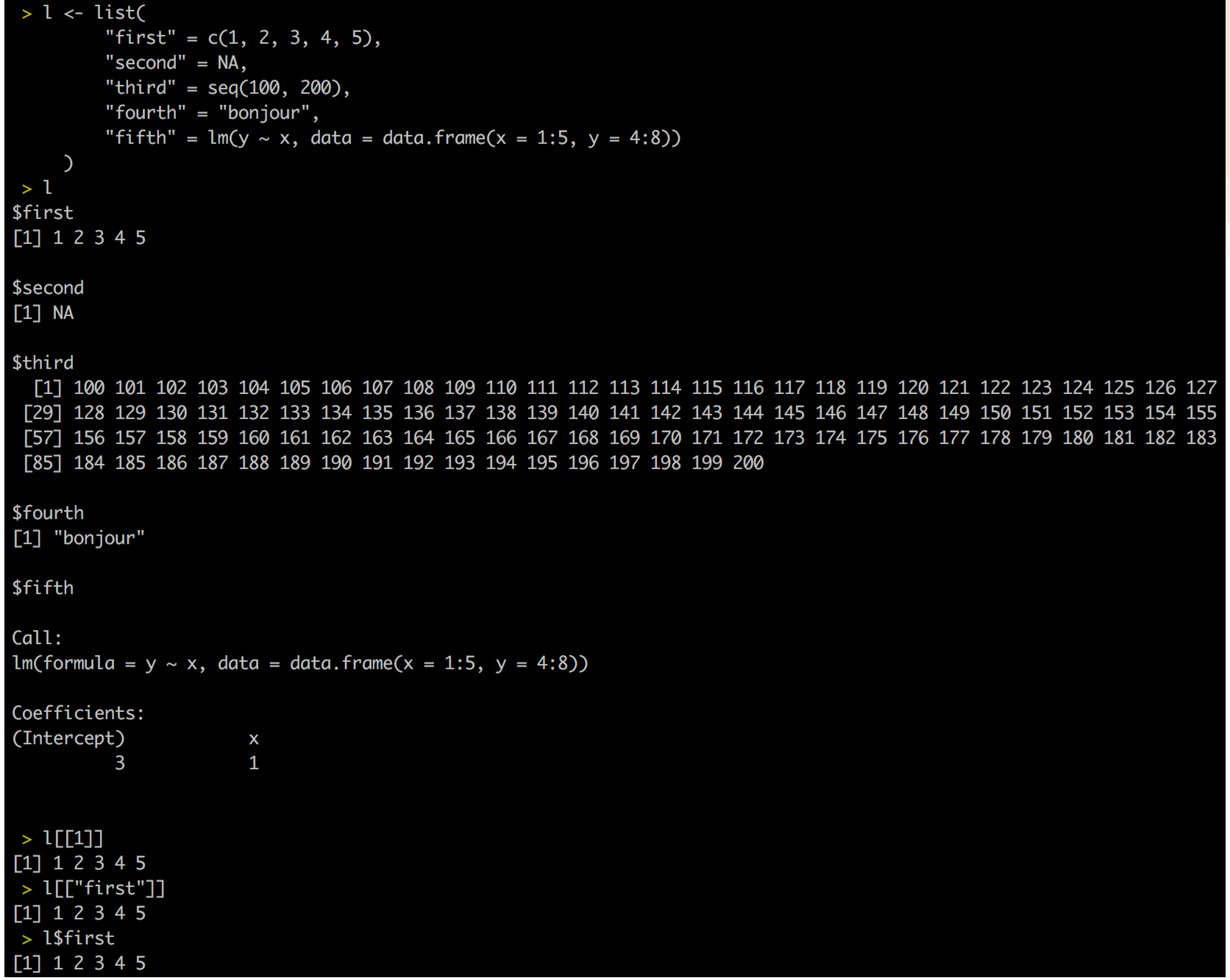
• Lists



```
 > l <- list(
          "first" = c(1, 2, 3, 4, 5),
          "second" = NA,
          "third" = seq(100, 200),
          "fourth" = "bonjour",
          "fifth" = lm(y ~ x, data = data.frame(x = 1:5, y = 4:8))
      )
 > l
$first
[1] 1 2 3 4 5

$second
[1] NA

$third
  [1] 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
 [29] 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155
 [57] 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183
 [85] 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200

$fourth
[1] "bonjour"

$fifth

Call:
lm(formula = y ~ x, data = data.frame(x = 1:5, y = 4:8))

Coefficients:
(Intercept)            x
          3            1
```

R main classes

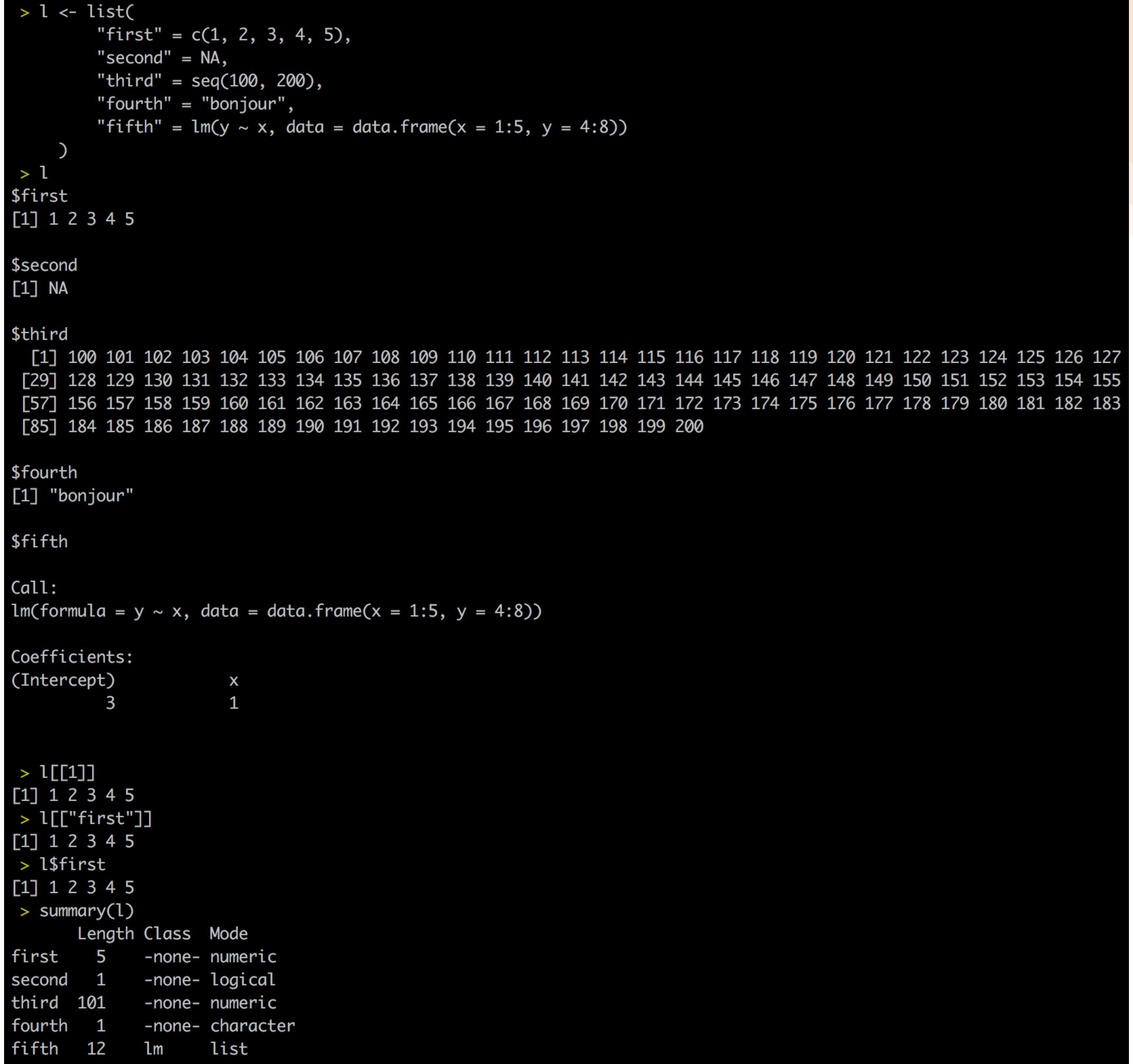
- Lists



```
 > l <- list(
         "first" = c(1, 2, 3, 4, 5),
         "second" = NA,
         "third" = seq(100, 200),
         "fourth" = "bonjour",
         "fifth" = lm(y ~ x, data = data.frame(x = 1:5, y = 4:8))
     )
 > l
$first
[1] 1 2 3 4 5

$second
[1] NA

$third
  [1] 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
 [29] 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155
 [57] 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183
 [85] 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200

$fourth
[1] "bonjour"

$fifth

Call:
lm(formula = y ~ x, data = data.frame(x = 1:5, y = 4:8))

Coefficients:
(Intercept)            x
          3            1



 > l[[1]]
[1] 1 2 3 4 5
 > l[["first"]]
[1] 1 2 3 4 5
 > l$first
[1] 1 2 3 4 5
```

R main classes

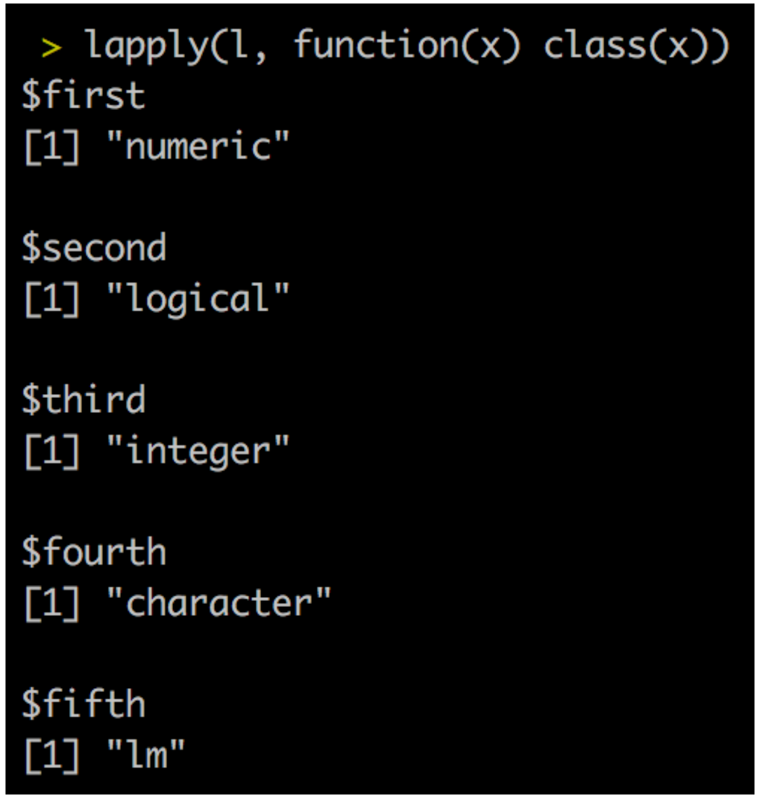
- Lists



```
> l <- list(
        "first" = c(1, 2, 3, 4, 5),
        "second" = NA,
        "third" = seq(100, 200),
        "fourth" = "bonjour",
        "fifth" = lm(y ~ x, data = data.frame(x = 1:5, y = 4:8))
    )
> l
$first
[1] 1 2 3 4 5

$second
[1] NA

$third
  [1] 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
 [29] 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155
 [57] 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183
 [85] 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200

$fourth
[1] "bonjour"

$fifth

Call:
lm(formula = y ~ x, data = data.frame(x = 1:5, y = 4:8))

Coefficients:
(Intercept)            x
          3            1



> l[[1]]
[1] 1 2 3 4 5
> l[["first"]]
[1] 1 2 3 4 5
> l$first
[1] 1 2 3 4 5
> summary(l)
       Length Class  Mode
first    5    -none- numeric
second   1    -none- logical
third  101    -none- numeric
fourth   1    -none- character
fifth   12    lm     list
```

R main classes

- You can iterate a function over each element of a list using lapply(list, function(element) {…})



```
> lapply(l, function(x) class(x))
$first
[1] "numeric"

$second
[1] "logical"

$third
[1] "integer"

$fourth
[1] "character"

$fifth
[1] "lm"
```

R main classes

- You can iterate a function over each element of a list using lapply(list, function(element) {…})
  - This is the equivalent of a for loop, but:
    - Can be significantly faster
    - Parallelizable
    - Does not modify your environment
    - Returns anything you want in a list
  - If you are writing a for loop, there are good chances you can switch to lapply()

Vectors vs. list vs. data.frames

- Vectors are series of single elements
- Lists are nested vectors
- Data.frames are a special case of lists where all elements are of the same length

R essential packages

- Tidyverse (https://rstudio-education.github.io/tidyverse-cookbook/program.html)
  - Verb-based ecosystem

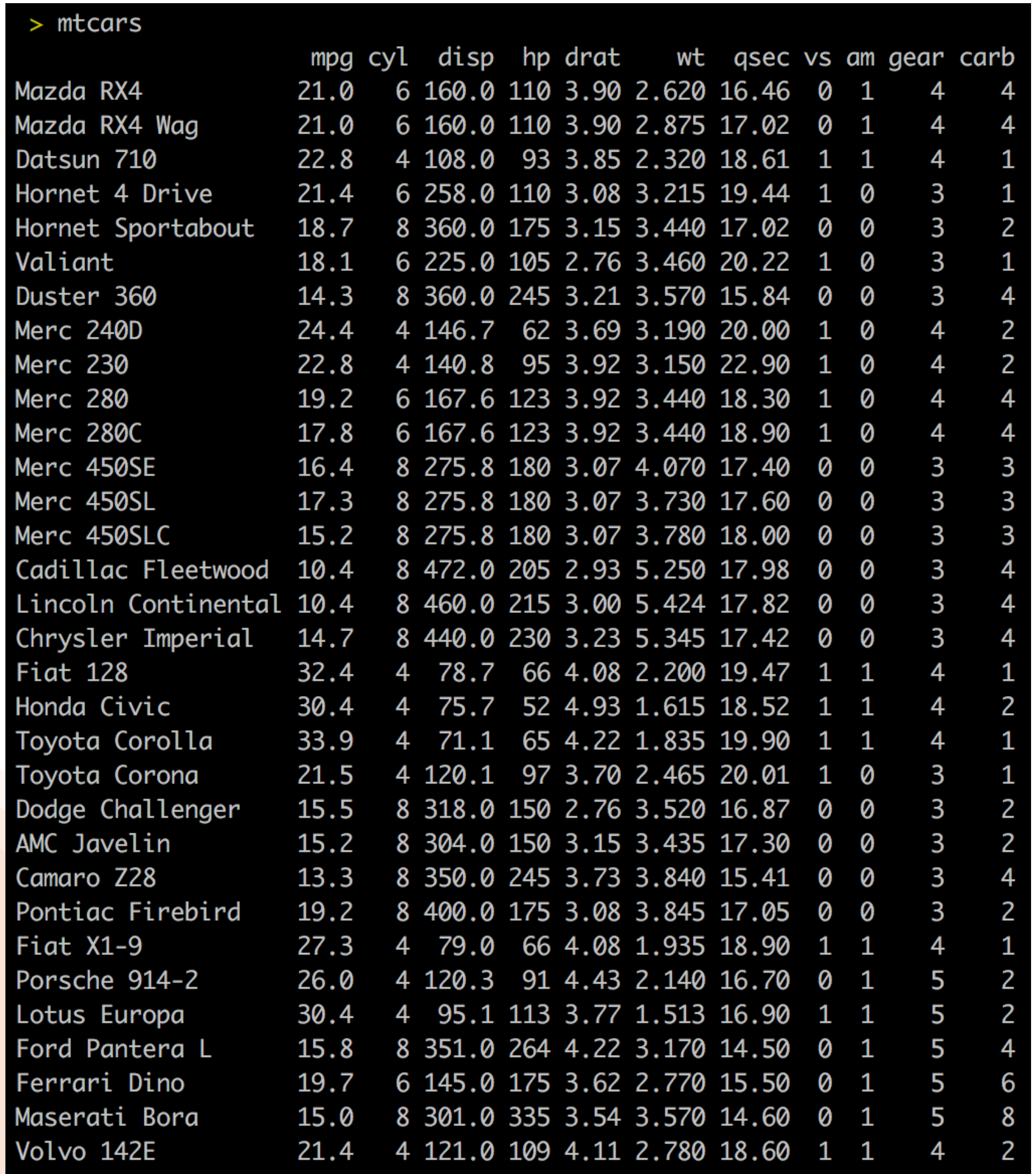
  - dplyr::filter
  - dplyr::arrange
  - dplyr::mutate
  - tidyr::gather
  - ggplot2 plotting functions

Everything documented/summarized here: https://www.r-bloggers.com/2020/12/the-tidyverse-in-a-table/

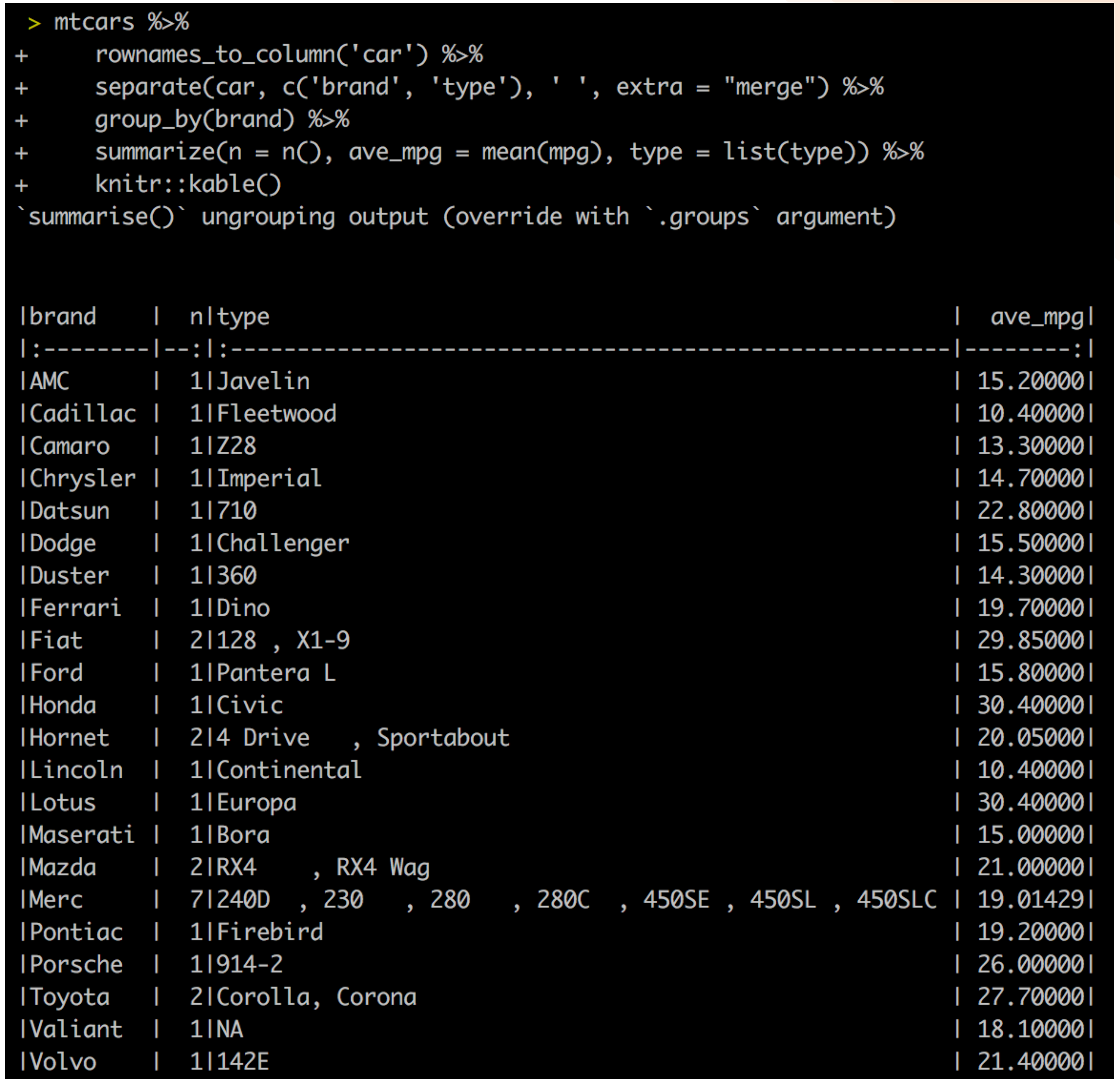
R essential packages

- Magrittr’s pipe `%>%`
    - Just like a pipe in bash, for R
    - Very useful in combination with tidyverse’s dplyr for data wrangling

R essential packages



```
> mtcars
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```



```
> mtcars %>%
+     rownames_to_column('car') %>%
+     separate(car, c('brand', 'type'), ' ', extra = "merge") %>%
+     group_by(brand) %>%
+     summarize(n = n(), ave_mpg = mean(mpg), type = list(type)) %>%
+     knitr::kable()
`summarise()` ungrouping output (override with `.groups` argument)


|brand    |  n|type                                                      |  ave_mpg|
|:--------|--:|:---------------------------------------------------------|--------:|
|AMC      |  1|Javelin                                                   | 15.20000|
|Cadillac |  1|Fleetwood                                                 | 10.40000|
|Camaro   |  1|Z28                                                       | 13.30000|
|Chrysler |  1|Imperial                                                  | 14.70000|
|Datsun   |  1|710                                                       | 22.80000|
|Dodge    |  1|Challenger                                                | 15.50000|
|Duster   |  1|360                                                       | 14.30000|
|Ferrari  |  1|Dino                                                      | 19.70000|
|Fiat     |  2|128 , X1-9                                                | 29.85000|
|Ford     |  1|Pantera L                                                 | 15.80000|
|Honda    |  1|Civic                                                     | 30.40000|
|Hornet   |  2|4 Drive    , Sportabout                                   | 20.05000|
|Lincoln  |  1|Continental                                               | 10.40000|
|Lotus    |  1|Europa                                                    | 30.40000|
|Maserati |  1|Bora                                                      | 15.00000|
|Mazda    |  2|RX4     , RX4 Wag                                         | 21.00000|
|Merc     |  7|240D   , 230    , 280    , 280C   , 450SE  , 450SL  , 450SLC | 19.01429|
|Pontiac  |  1|Firebird                                                  | 19.20000|
|Porsche  |  1|914-2                                                     | 26.00000|
|Toyota   |  2|Corolla, Corona                                           | 27.70000|
|Valiant  |  1|NA                                                        | 18.10000|
|Volvo    |  1|142E                                                      | 21.40000|
```

R essential packages

- R per-se is useful for statistical analyses.
- How do we unlock the power of R-stats in genomics?

What do you need in bioinformatics to study genomics?

- Most common genomic files:
    - BED format: essentially a set of chromosomal ranges

What do you need in bioinformatics to study genomics?

- Most common genomic files:
  - BED format: essentially a set of chromosomal ranges
  - BigWig format: essentially veeeeeeeee…eeeeery long numerical vectors

What do you need in bioinformatics to study genomics?

- Most common genomic files:
  - BED format: essentially a set of chromosomal ranges
  - BigWig format: essentially veeeeeeeee…eeeeery long numerical vectors
  - Fasta format: letters, letters, letters

What do you need in bioinformatics to study genomics?

- Most common genomic files:
  - BED format: essentially a set of chromosomal ranges
  - BigWig format: essentially veeeeeeeee…eeeeery long numerical vectors
  - Fasta format: letters, letters, letters
  - Others (bam, GFF, …): can usually be described/built on as one of the two options above

Bioconductor

https://www.bioconductor.org/

- Free, open-source repository
- Open development software project
- Based primarily on the statistical R programming language
- Bioconductor's releases are bi-annual
- **<u>Provides tools for the analysis and comprehension of genomic data</u>**

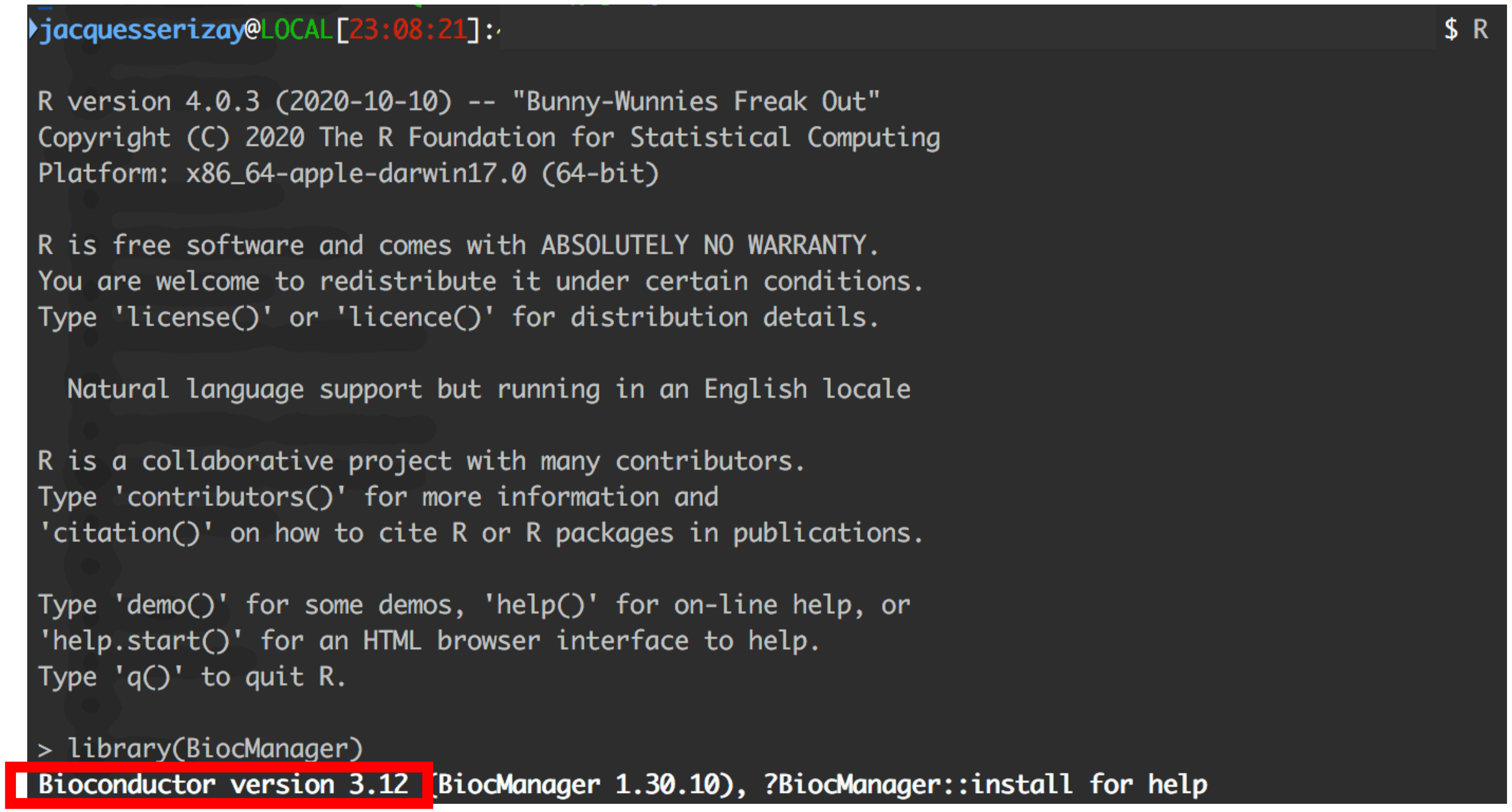
Bioconductor installation

- Through BiocManager

- As easy as: `install.packages("BiocManager")`

Bioconductor

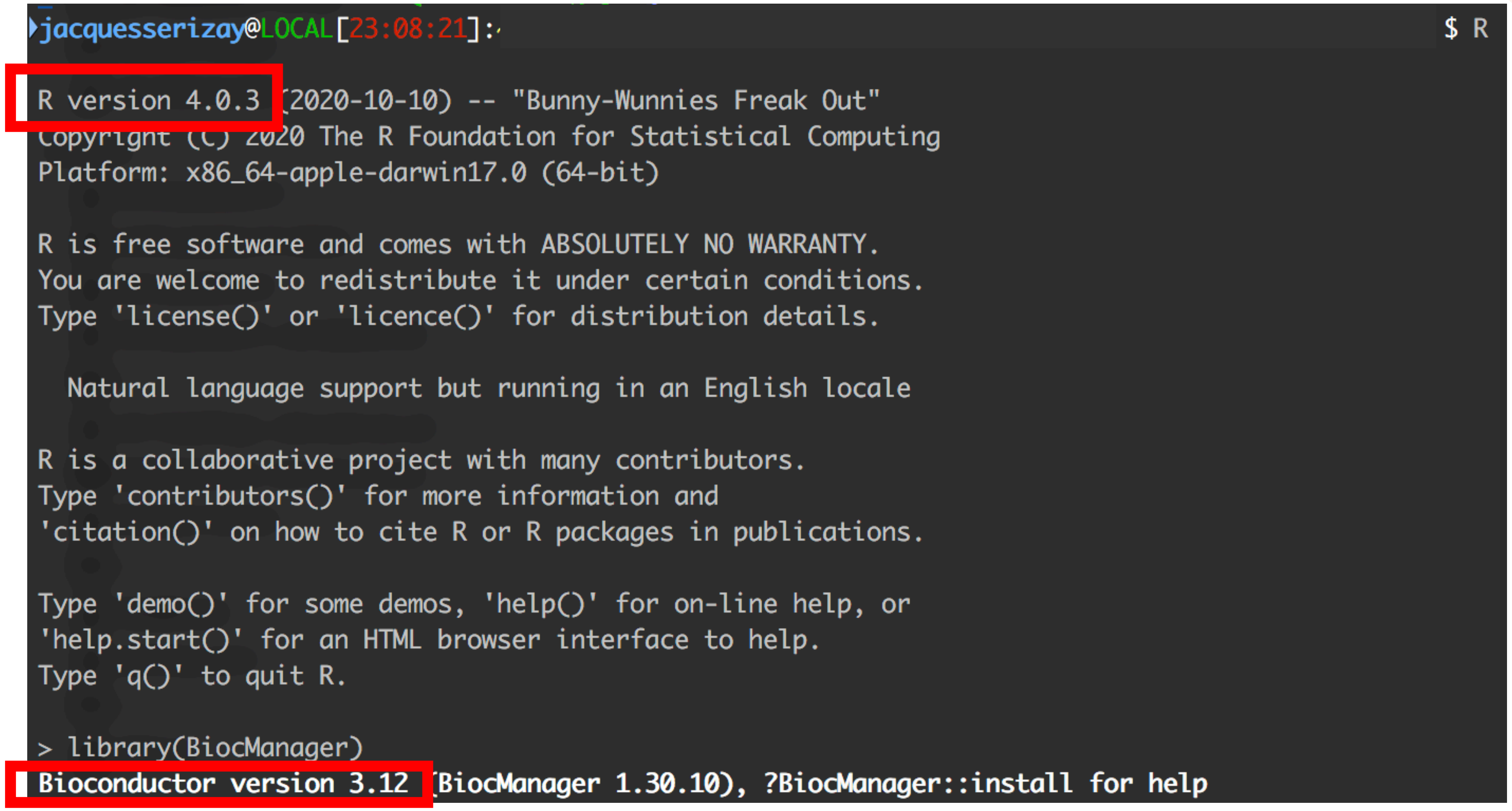
- Integrated in R



```
jacquesserizay@LOCAL[23:08:21]:                                                          $ R

R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(BiocManager)
Bioconductor version 3.12 (BiocManager 1.30.10), ?BiocManager::install for help
```

Bioconductor

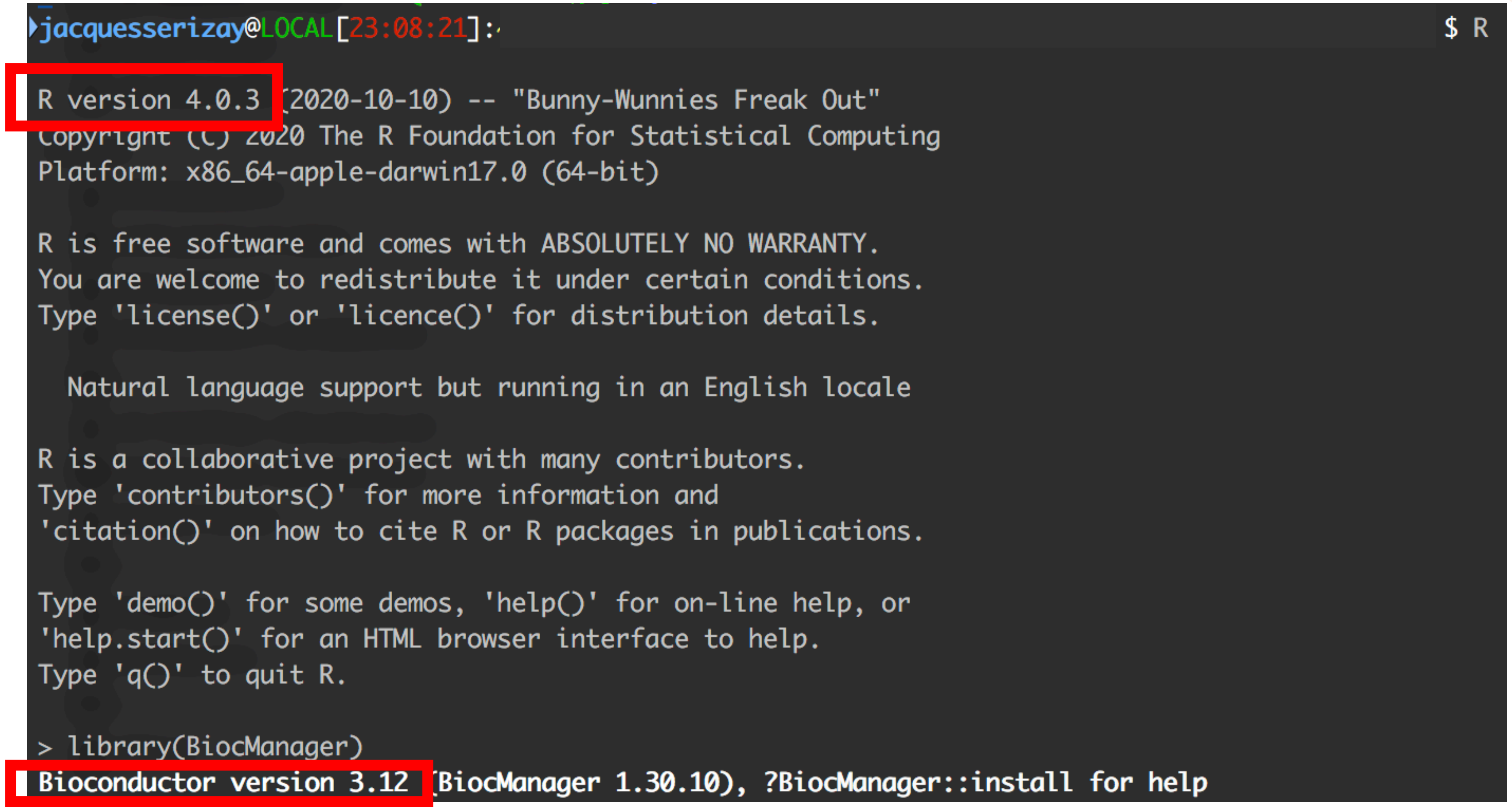
- Integrated in R
- Bioconductor's version depends on your R version



```
jacquesserizay@LOCAL[23:08:21]:~                                    $ R

R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(BiocManager)
Bioconductor version 3.12 (BiocManager 1.30.10), ?BiocManager::install for help
```

Bioconductor

- Some Bioc packages are restricted to a certain version!



```
jacquesserizay@LOCAL[23:08:21]:                                                   $ R

R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(BiocManager)
Bioconductor version 3.12 (BiocManager 1.30.10), ?BiocManager::install for help
```

Installation of Bioconductor packages

- Bioconductor packages are on Bioconductor, not CRAN
- So you install them using Bioconductor's BiocManager!

```
BiocManager::install("Biostrings")

BiocManager::install(c("GenomicRanges”, "rtracklayer"))
```

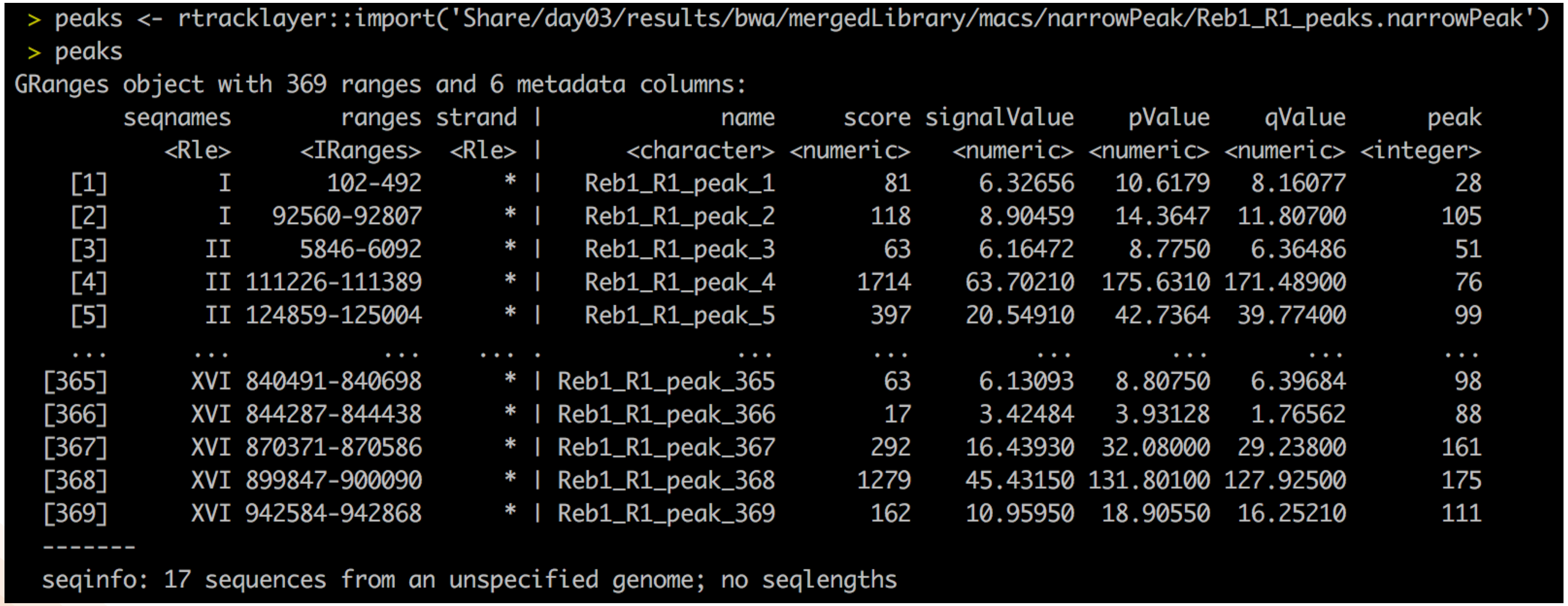
Bioconductor essentials

- GRanges (through GenomicRanges package)
- XNABiostrings (through Biostrings)

- Import/export from/to common genomic files with import() and export() (through rtracklayer package)

GRanges

- Workhorse class of Bioconductor
- Used to describe genomic intervals or ranges



```
 > peaks <- rtracklayer::import('Share/day03/results/bwa/mergedLibrary/macs/narrowPeak/Reb1_R1_peaks.narrowPeak')
 > peaks
GRanges object with 369 ranges and 6 metadata columns:
        seqnames        ranges strand |             name     score signalValue    pValue    qValue      peak
           <Rle>     <IRanges>  <Rle> |      <character> <numeric>   <numeric> <numeric> <numeric> <integer>
    [1]        I       102-492      * |   Reb1_R1_peak_1        81     6.32656   10.6179   8.16077        28
    [2]        I   92560-92807      * |   Reb1_R1_peak_2       118     8.90459   14.3647  11.80700       105
    [3]       II     5846-6092      * |   Reb1_R1_peak_3        63     6.16472    8.7750   6.36486        51
    [4]       II 111226-111389      * |   Reb1_R1_peak_4      1714    63.70210  175.6310 171.48900        76
    [5]       II 124859-125004      * |   Reb1_R1_peak_5       397    20.54910   42.7364  39.77400        99
    ...      ...           ...    ... .              ...       ...         ...       ...       ...       ...
  [365]      XVI 840491-840698      * | Reb1_R1_peak_365        63     6.13093   8.80750   6.39684        98
  [366]      XVI 844287-844438      * | Reb1_R1_peak_366        17     3.42484   3.93128   1.76562        88
  [367]      XVI 870371-870586      * | Reb1_R1_peak_367       292    16.43930  32.08000  29.23800       161
  [368]      XVI 899847-900090      * | Reb1_R1_peak_368      1279    45.43150 131.80100 127.92500       175
  [369]      XVI 942584-942868      * | Reb1_R1_peak_369       162    10.95950  18.90550  16.25210       111
  -------
  seqinfo: 17 sequences from an unspecified genome; no seqlengths
```

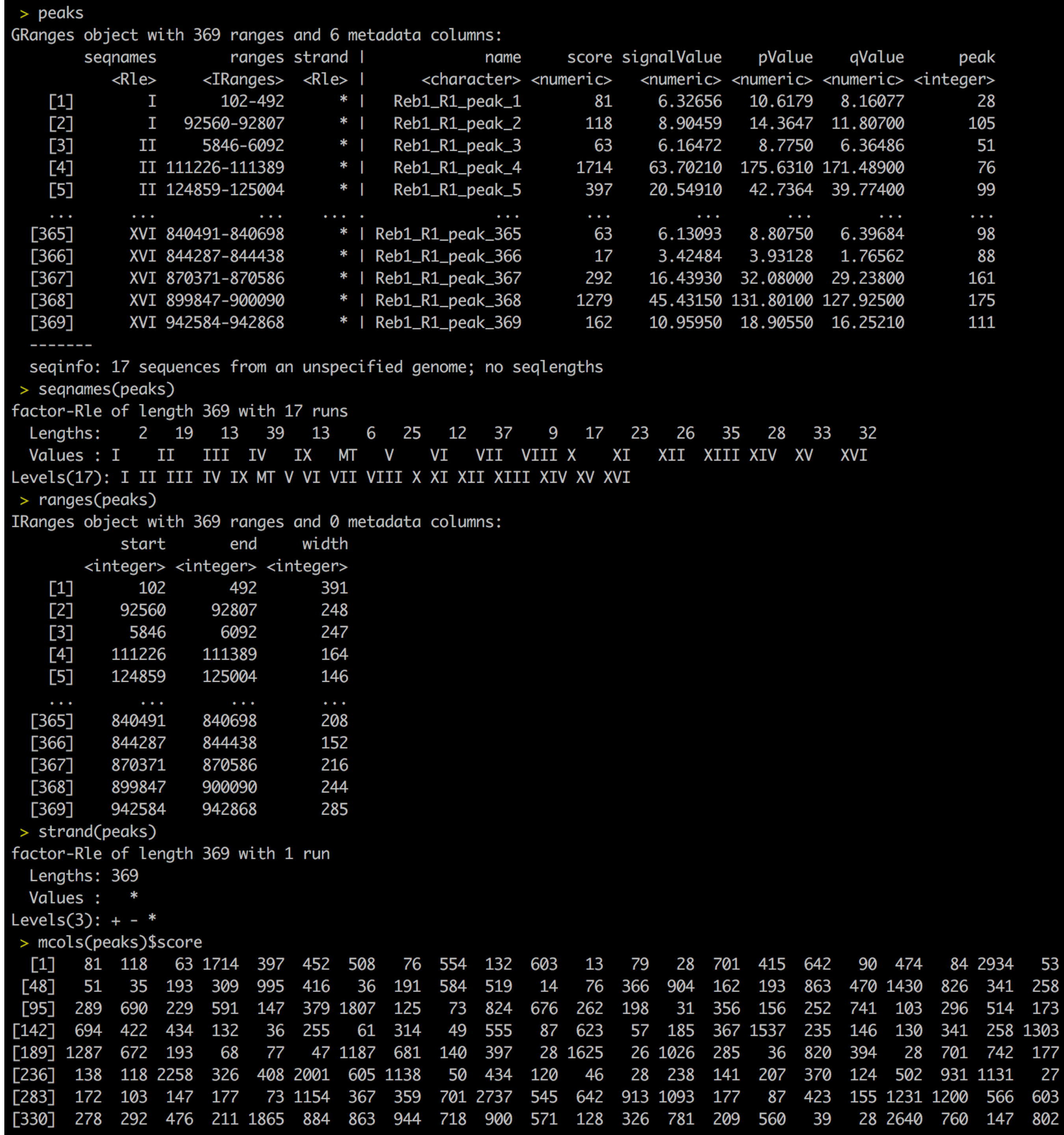
GRanges

Getter functions

- seqnames() → chromosome
- ranges() → location
- strand()
- start()
- end()
- width()
- mcols(…)$... → metadata
- …$... → metadata

2020/01/13

Jacques Serizay



```
 > peaks
GRanges object with 369 ranges and 6 metadata columns:
        seqnames          ranges strand |             name     score signalValue    pValue    qValue      peak
           <Rle>       <IRanges>  <Rle> |      <character> <numeric>   <numeric> <numeric> <numeric> <integer>
    [1]        I         102-492      * |   Reb1_R1_peak_1        81     6.32656   10.6179   8.16077        28
    [2]        I     92560-92807      * |   Reb1_R1_peak_2       118     8.90459   14.3647  11.80700       105
    [3]       II       5846-6092      * |   Reb1_R1_peak_3        63     6.16472    8.7750   6.36486        51
    [4]       II   111226-111389      * |   Reb1_R1_peak_4      1714    63.70210  175.6310 171.48900        76
    [5]       II   124859-125004      * |   Reb1_R1_peak_5       397    20.54910   42.7364  39.77400        99
    ...      ...             ...    ... .              ...       ...         ...       ...       ...       ...
  [365]      XVI   840491-840698      * | Reb1_R1_peak_365        63     6.13093   8.80750   6.39684        98
  [366]      XVI   844287-844438      * | Reb1_R1_peak_366        17     3.42484   3.93128   1.76562        88
  [367]      XVI   870371-870586      * | Reb1_R1_peak_367       292    16.43930  32.08000  29.23800       161
  [368]      XVI   899847-900090      * | Reb1_R1_peak_368      1279    45.43150 131.80100 127.92500       175
  [369]      XVI   942584-942868      * | Reb1_R1_peak_369       162    10.95950  18.90550  16.25210       111
  -------
  seqinfo: 17 sequences from an unspecified genome; no seqlengths
 > seqnames(peaks)
factor-Rle of length 369 with 17 runs
  Lengths:    2   19   13   39   13    6   25   12   37    9   17   23   26   35   28   33   32
  Values : I    II   III  IV   IX   MT   V    VI   VII  VIII X    XI   XII  XIII XIV  XV   XVI
Levels(17): I II III IV IX MT V VI VII VIII X XI XII XIII XIV XV XVI
 > ranges(peaks)
IRanges object with 369 ranges and 0 metadata columns:
            start       end     width
        <integer> <integer> <integer>
    [1]       102       492       391
    [2]     92560     92807       248
    [3]      5846      6092       247
    [4]    111226    111389       164
    [5]    124859    125004       146
    ...       ...       ...       ...
  [365]    840491    840698       208
  [366]    844287    844438       152
  [367]    870371    870586       216
  [368]    899847    900090       244
  [369]    942584    942868       285
 > strand(peaks)
factor-Rle of length 369 with 1 run
  Lengths: 369
  Values :   *
Levels(3): + - *
 > mcols(peaks)$score
  [1]   81  118   63 1714  397  452  508   76  554  132  603   13   79   28  701  415  642   90  474   84 2934   53
 [48]   51   35  193  309  995  416   36  191  584  519   14   76  366  904  162  193  863  470 1430  826  341  258
 [95]  289  690  229  591  147  379 1807  125   73  824  676  262  198   31  356  156  252  741  103  296  514  173
[142]  694  422  434  132   36  255   61  314   49  555   87  623   57  185  367 1537  235  146  130  341  258 1303
[189] 1287  672  193   68   77   47 1187  681  140  397   28 1625   26 1026  285   36  820  394   28  701  742  177
[236]  138  118 2258  326  408 2001  605 1138   50  434  120   46   28  238  141  207  370  124  502  931 1131   27
[283]  172  103  147  177   73 1154  367  359  701 2737  545  642  913 1093  177   87  423  155 1231 1200  566  603
[330]  278  292  476  211 1865  884  863  944  718  900  571  128  326  781  209  560   39   28 2640  760  147  802
```

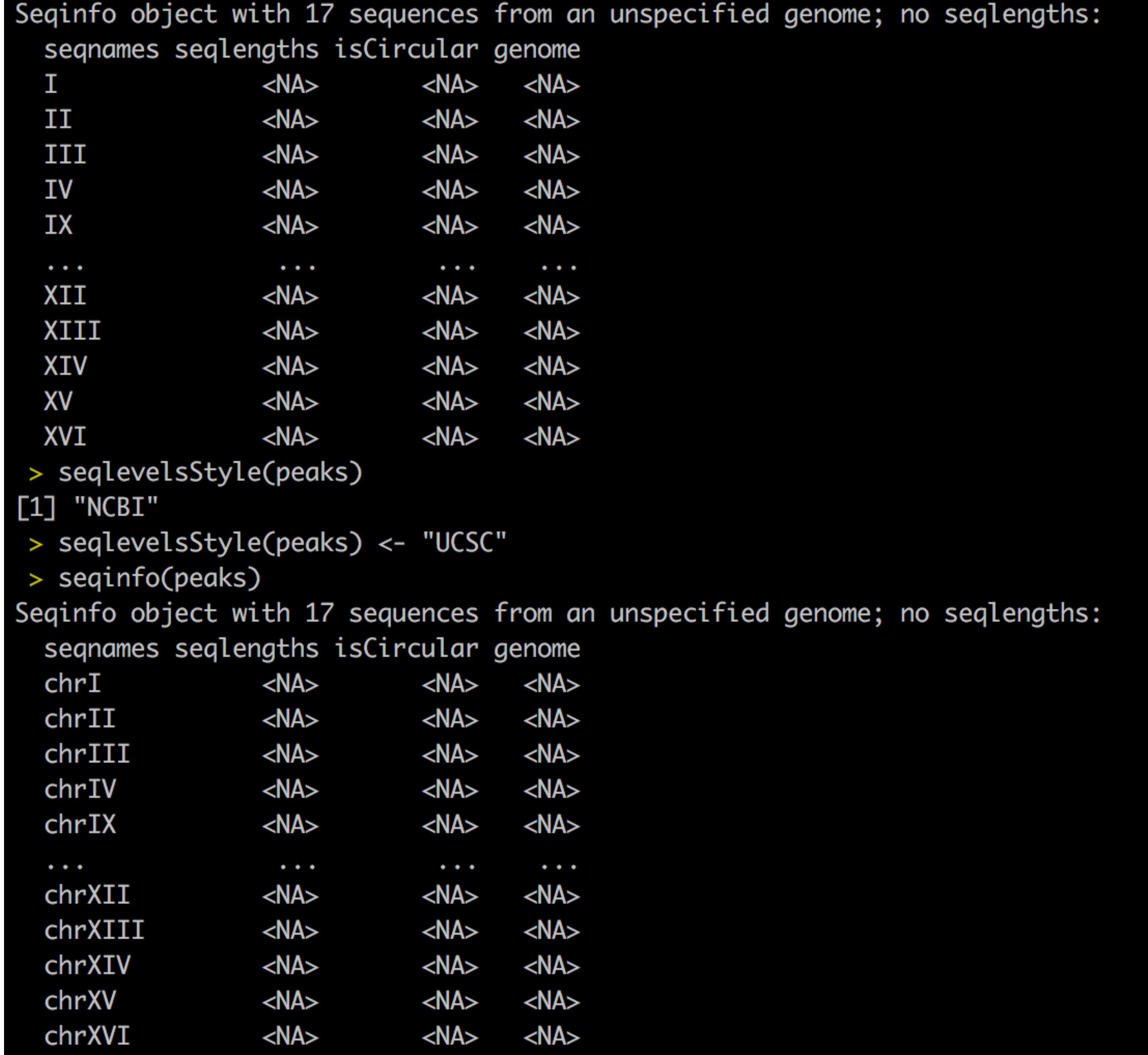
GRanges

Genome information

- seqinfo()
- seqlevelsStyle()

2020/01/13

Jacques Serizay



```
Seqinfo object with 17 sequences from an unspecified genome; no seqlengths:
  seqnames seqlengths isCircular genome
  I              <NA>       <NA>   <NA>
  II             <NA>       <NA>   <NA>
  III            <NA>       <NA>   <NA>
  IV             <NA>       <NA>   <NA>
  IX             <NA>       <NA>   <NA>
  ...             ...        ...    ...
  XII            <NA>       <NA>   <NA>
  XIII           <NA>       <NA>   <NA>
  XIV            <NA>       <NA>   <NA>
  XV             <NA>       <NA>   <NA>
  XVI            <NA>       <NA>   <NA>
 > seqlevelsStyle(peaks)
[1] "NCBI"
 > seqlevelsStyle(peaks) <- "UCSC"
 > seqinfo(peaks)
Seqinfo object with 17 sequences from an unspecified genome; no seqlengths:
  seqnames seqlengths isCircular genome
  chrI           <NA>       <NA>   <NA>
  chrII          <NA>       <NA>   <NA>
  chrIII         <NA>       <NA>   <NA>
  chrIV          <NA>       <NA>   <NA>
  chrIX          <NA>       <NA>   <NA>
  ...             ...        ...    ...
  chrXII         <NA>       <NA>   <NA>
  chrXIII        <NA>       <NA>   <NA>
  chrXIV         <NA>       <NA>   <NA>
  chrXV          <NA>       <NA>   <NA>
  chrXVI         <NA>       <NA>   <NA>
```

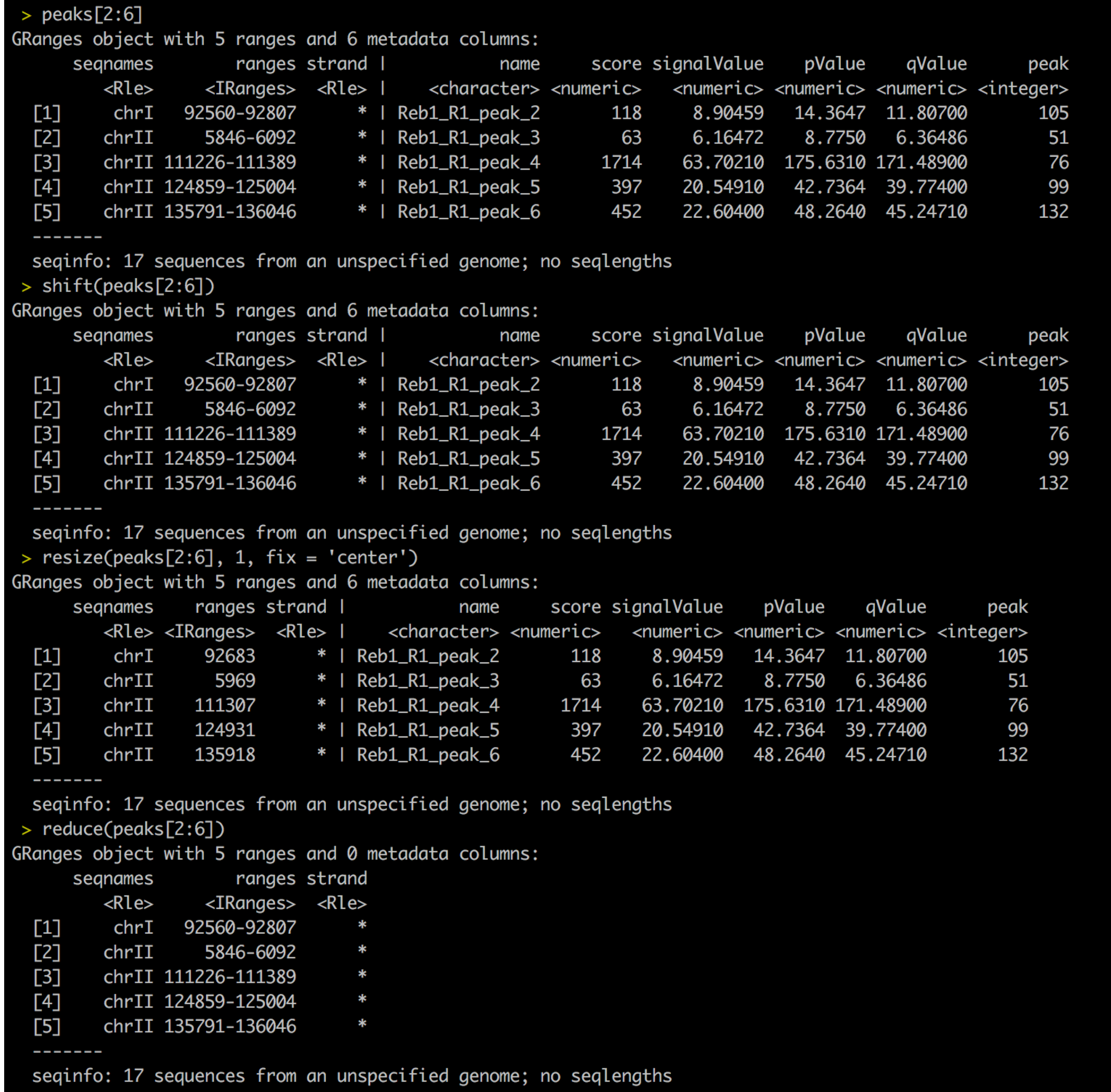
GRanges

Action functions

- …[…] (to subset)
- shift()
- resize()
- reduce()
- coverage()
- …

2020/01/13

Jacques Serizay



```
> peaks[2:6]
GRanges object with 5 ranges and 6 metadata columns:
      seqnames          ranges strand |            name     score signalValue    pValue    qValue      peak
         <Rle>       <IRanges>  <Rle> |     <character> <numeric>   <numeric> <numeric> <numeric> <integer>
  [1]     chrI     92560-92807      * | Reb1_R1_peak_2       118     8.90459   14.3647  11.80700       105
  [2]    chrII       5846-6092      * | Reb1_R1_peak_3        63     6.16472    8.7750   6.36486        51
  [3]    chrII   111226-111389      * | Reb1_R1_peak_4      1714    63.70210  175.6310 171.48900        76
  [4]    chrII   124859-125004      * | Reb1_R1_peak_5       397    20.54910   42.7364  39.77400        99
  [5]    chrII   135791-136046      * | Reb1_R1_peak_6       452    22.60400   48.2640  45.24710       132
  -------
  seqinfo: 17 sequences from an unspecified genome; no seqlengths
> shift(peaks[2:6])
GRanges object with 5 ranges and 6 metadata columns:
      seqnames          ranges strand |            name     score signalValue    pValue    qValue      peak
         <Rle>       <IRanges>  <Rle> |     <character> <numeric>   <numeric> <numeric> <numeric> <integer>
  [1]     chrI     92560-92807      * | Reb1_R1_peak_2       118     8.90459   14.3647  11.80700       105
  [2]    chrII       5846-6092      * | Reb1_R1_peak_3        63     6.16472    8.7750   6.36486        51
  [3]    chrII   111226-111389      * | Reb1_R1_peak_4      1714    63.70210  175.6310 171.48900        76
  [4]    chrII   124859-125004      * | Reb1_R1_peak_5       397    20.54910   42.7364  39.77400        99
  [5]    chrII   135791-136046      * | Reb1_R1_peak_6       452    22.60400   48.2640  45.24710       132
  -------
  seqinfo: 17 sequences from an unspecified genome; no seqlengths
> resize(peaks[2:6], 1, fix = 'center')
GRanges object with 5 ranges and 6 metadata columns:
      seqnames    ranges strand |            name     score signalValue    pValue    qValue      peak
         <Rle> <IRanges>  <Rle> |     <character> <numeric>   <numeric> <numeric> <numeric> <integer>
  [1]     chrI     92683      * | Reb1_R1_peak_2       118     8.90459   14.3647  11.80700       105
  [2]    chrII      5969      * | Reb1_R1_peak_3        63     6.16472    8.7750   6.36486        51
  [3]    chrII    111307      * | Reb1_R1_peak_4      1714    63.70210  175.6310 171.48900        76
  [4]    chrII    124931      * | Reb1_R1_peak_5       397    20.54910   42.7364  39.77400        99
  [5]    chrII    135918      * | Reb1_R1_peak_6       452    22.60400   48.2640  45.24710       132
  -------
  seqinfo: 17 sequences from an unspecified genome; no seqlengths
> reduce(peaks[2:6])
GRanges object with 5 ranges and 0 metadata columns:
      seqnames          ranges strand
         <Rle>       <IRanges>  <Rle>
  [1]     chrI     92560-92807      *
  [2]    chrII       5846-6092      *
  [3]    chrII   111226-111389      *
  [4]    chrII   124859-125004      *
  [5]    chrII   135791-136046      *
  -------
  seqinfo: 17 sequences from an unspecified genome; no seqlengths
```

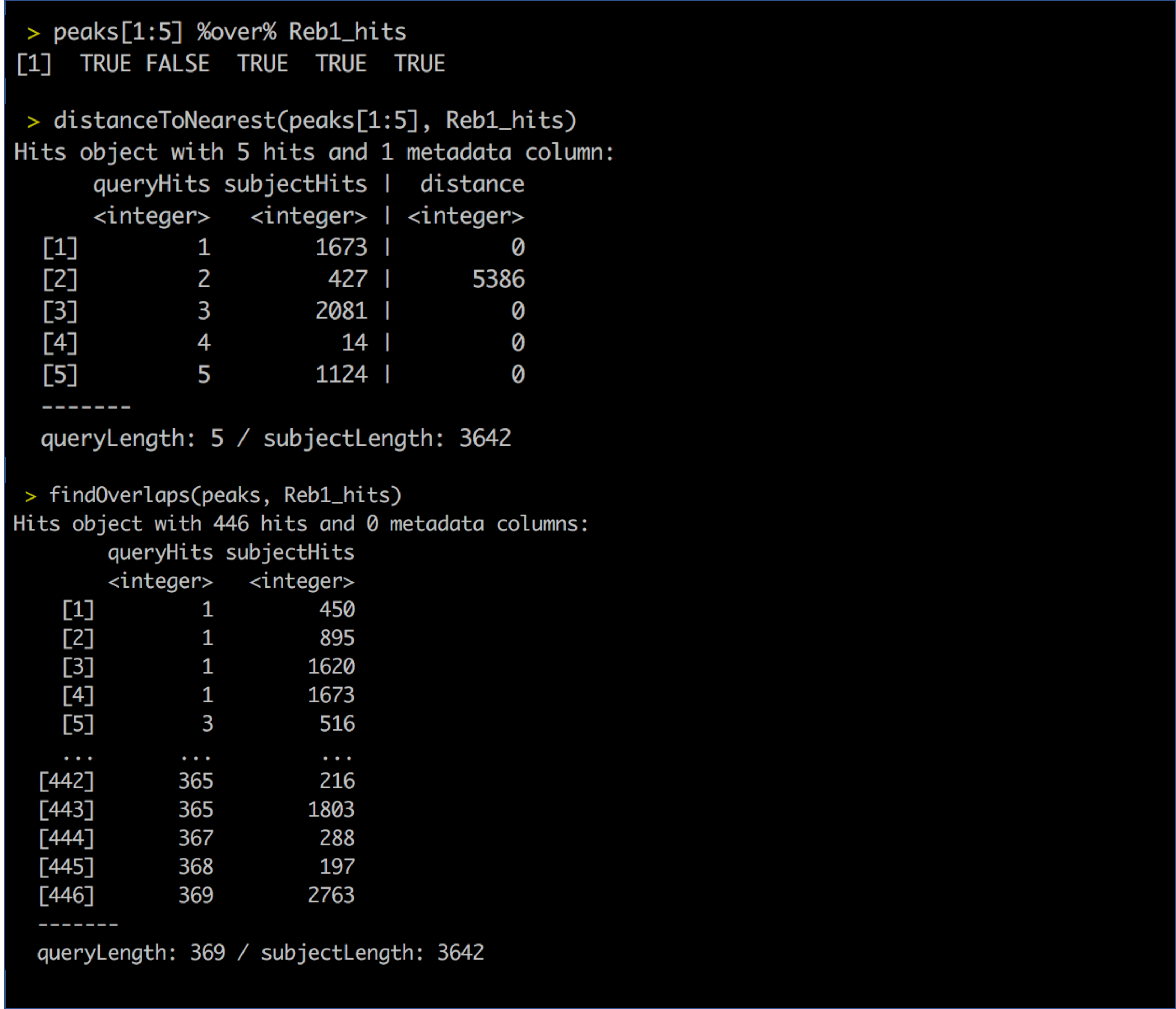
GRanges

Comparison functions

- %over%
- distance()
- distanceToNearest()
- findOverlaps()
- subsetByOverlaps()

2020/01/13

Jacques Serizay

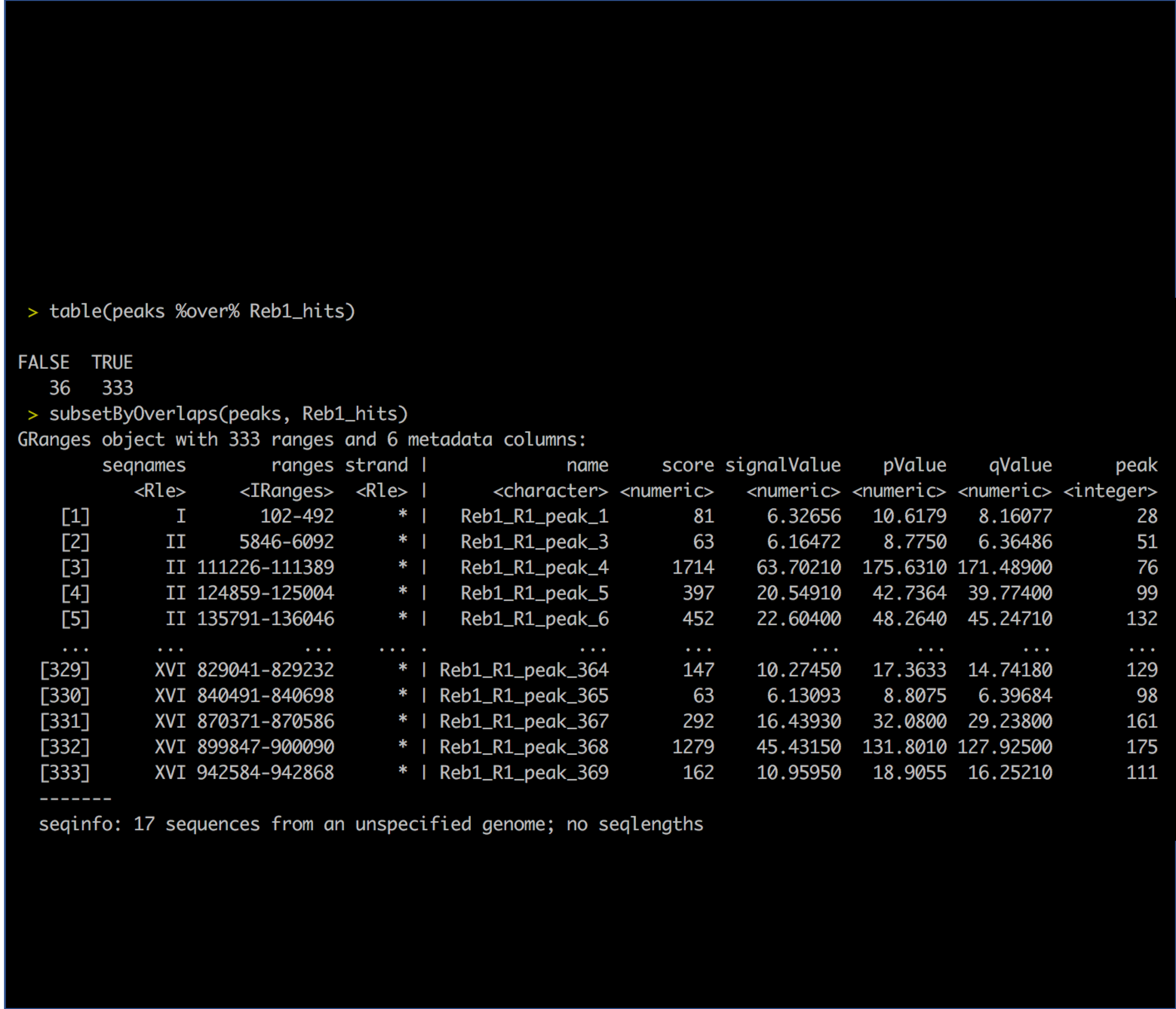


```
 > peaks[1:5] %over% Reb1_hits
[1]  TRUE FALSE  TRUE  TRUE  TRUE

 > distanceToNearest(peaks[1:5], Reb1_hits)
Hits object with 5 hits and 1 metadata column:
      queryHits subjectHits |  distance
      <integer>   <integer> | <integer>
  [1]         1        1673 |         0
  [2]         2         427 |      5386
  [3]         3        2081 |         0
  [4]         4          14 |         0
  [5]         5        1124 |         0
  -------
  queryLength: 5 / subjectLength: 3642

 > findOverlaps(peaks, Reb1_hits)
Hits object with 446 hits and 0 metadata columns:
        queryHits subjectHits
        <integer>   <integer>
    [1]         1         450
    [2]         1         895
    [3]         1        1620
    [4]         1        1673
    [5]         3         516
    ...       ...         ...
  [442]       365         216
  [443]       365        1803
  [444]       367         288
  [445]       368         197
  [446]       369        2763
  -------
  queryLength: 369 / subjectLength: 3642
```

GRanges

Comparison functions

- %over%
- distance()
- distanceToNearest()
- findOverlaps()
- subsetByOverlaps()



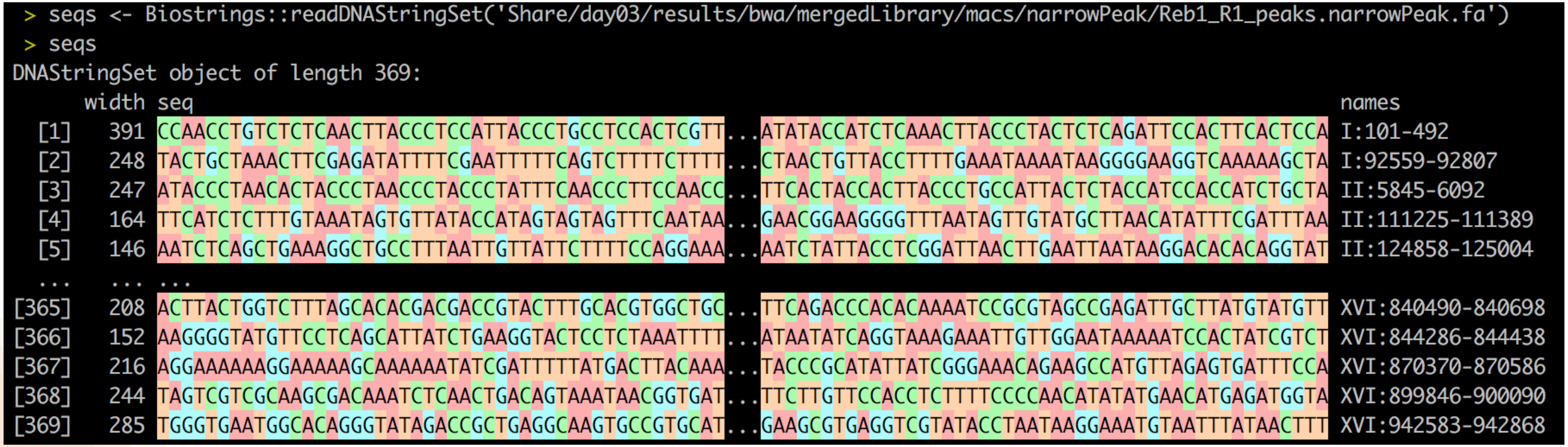
```
> table(peaks %over% Reb1_hits)

FALSE  TRUE 
   36   333 
> subsetByOverlaps(peaks, Reb1_hits)
GRanges object with 333 ranges and 6 metadata columns:
        seqnames        ranges strand |            name     score signalValue    pValue    qValue      peak
           <Rle>     <IRanges>  <Rle> |     <character> <numeric>   <numeric> <numeric> <numeric> <integer>
    [1]        I       102-492      * |   Reb1_R1_peak_1        81     6.32656   10.6179   8.16077        28
    [2]       II     5846-6092      * |   Reb1_R1_peak_3        63     6.16472    8.7750   6.36486        51
    [3]       II 111226-111389      * |   Reb1_R1_peak_4      1714    63.70210  175.6310 171.48900        76
    [4]       II 124859-125004      * |   Reb1_R1_peak_5       397    20.54910   42.7364  39.77400        99
    [5]       II 135791-136046      * |   Reb1_R1_peak_6       452    22.60400   48.2640  45.24710       132
    ...      ...           ...    ... .             ...       ...         ...       ...       ...       ...
  [329]      XVI 829041-829232      * | Reb1_R1_peak_364       147    10.27450   17.3633  14.74180       129
  [330]      XVI 840491-840698      * | Reb1_R1_peak_365        63     6.13093    8.8075   6.39684        98
  [331]      XVI 870371-870586      * | Reb1_R1_peak_367       292    16.43930   32.0800  29.23800       161
  [332]      XVI 899847-900090      * | Reb1_R1_peak_368      1279    45.43150  131.8010 127.92500       175
  [333]      XVI 942584-942868      * | Reb1_R1_peak_369       162    10.95950   18.9055  16.25210       111
  -------
  seqinfo: 17 sequences from an unspecified genome; no seqlengths
```

2020/01/13

Jacques Serizay

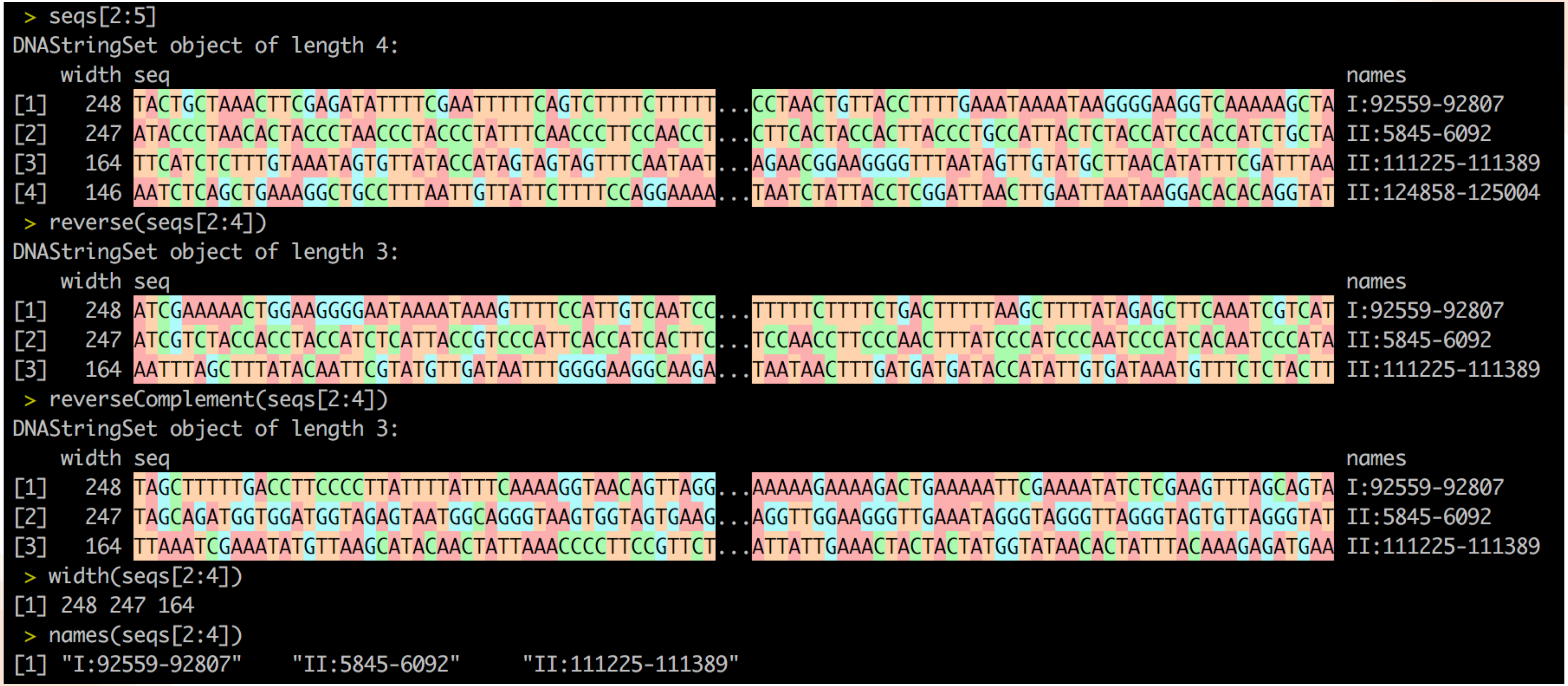
Biostrings

Biostrings in R



```
 > seqs <- Biostrings::readDNAStringSet('Share/day03/results/bwa/mergedLibrary/macs/narrowPeak/Reb1_R1_peaks.narrowPeak.fa')
 > seqs
DNAStringSet object of length 369:
      width seq                                                     names
  [1]   391 CCAACCTGTCTCTCAACTTACCCTCCATTACCCTGCCTCCACTCGTT...ATATACCATCTCAAACTTACCCTACTCTCAGATTCCACTTCACTCCA I:101-492
  [2]   248 TACTGCTAAACTTCGAGATATTTTCGAATTTTTCAGTCTTTTCTTTT...CTAACTGTTACCTTTTGAAATAAAATAAGGGGAAGGTCAAAAAGCTA I:92559-92807
  [3]   247 ATACCCTAACACTACCCTAACCCTACCCTATTTCAACCCTTCCAACC...TTCACTACCACTTACCCTGCCATTACTCTACCATCCACCATCTGCTA II:5845-6092
  [4]   164 TTCATCTCTTTGTAAATAGTGTTATACCATAGTAGTAGTTTCAATAA...GAACGGAAGGGGTTTAATAGTTGTATGCTTAACATATTTCGATTTAA II:111225-111389
  [5]   146 AATCTCAGCTGAAAGGCTGCCTTTAATTGTTATTCTTTTCCAGGAAA...AATCTATTACCTCGGATTAACTTGAATTAATAAGGACACACAGGTAT II:124858-125004
  ...   ... ...
[365]   208 ACTTACTGGTCTTTAGCACACGACGACCGTACTTTGCACGTGGCTGC...TTCAGACCCACACAAAATCCGCGTAGCCGAGATTGCTTATGTATGTT XVI:840490-840698
[366]   152 AAGGGGTATGTTCCTCAGCATTATCTGAAGGTACTCCTCTAAATTTT...ATAATATCAGGTAAAGAAATTGTTGGAATAAAAATCCACTATCGTCT XVI:844286-844438
[367]   216 AGGAAAAAAGGAAAAAGCAAAAAATATCGATTTTTATGACTTACAAA...TACCCGCATATTATCGGGAAACAGAAGCCATGTTAGAGTGATTTCCA XVI:870370-870586
[368]   244 TAGTCGTCGCAAGCGACAAATCTCAACTGACAGTAAATAACGGTGAT...TTCTTGTTCCACCTCTTTTCCCCAACATATATGAACATGAGATGGTA XVI:899846-900090
[369]   285 TGGGTGAATGGCACAGGGTATAGACCGCTGAGGCAAGTGCCGTGCAT...GAAGCGTGAGGTCGTATACCTAATAAGGAAATGTAATTTATAACTTT XVI:942583-942868
```

Biostrings



```
 > seqs[2:5]
DNAStringSet object of length 4:
    width seq                                                   names
[1]   248 TACTGCTAAACTTCGAGATATTTTCGAATTTTTCAGTCTTTTCTTTTT...CCTAACTGTTACCTTTTGAAATAAAATAAGGGGAAGGTCAAAAAGCTA I:92559-92807
[2]   247 ATACCCTAACACTACCCTAACCCTACCCTATTTCAACCCTTCCAACCT...CTTCACTACCACTTACCCTGCCATTACTCTACCATCCACCATCTGCTA II:5845-6092
[3]   164 TTCATCTCTTTGTAAATAGTGTTATACCATAGTAGTAGTTTCAATAAT...AGAACGGAAGGGGTTTAATAGTTGTATGCTTAACATATTTCGATTTAA II:111225-111389
[4]   146 AATCTCAGCTGAAAGGCTGCCTTTAATTGTTATTCTTTTCCAGGAAAA...TAATCTATTACCTCGGATTAACTTGAATTAATAAGGACACACAGGTAT II:124858-125004
 > reverse(seqs[2:4])
DNAStringSet object of length 3:
    width seq                                                   names
[1]   248 ATCGAAAAACTGGAAGGGGAATAAAATAAAGTTTTCCATTGTCAATCC...TTTTTCTTTTCTGACTTTTTAAGCTTTTATAGAGCTTCAAATCGTCAT I:92559-92807
[2]   247 ATCGTCTACCACCTACCATCTCATTACCGTCCCATTCACCATCACTTC...TCCAACCTTCCCAACTTTATCCCATCCCAATCCCATCACAATCCCATA II:5845-6092
[3]   164 AATTTAGCTTTATACAATTCGTATGTTGATAATTTGGGGAAGGCAAGA...TAATAACTTTGATGATGATACCATATTGTGATAAATGTTTCTCTACTT II:111225-111389
 > reverseComplement(seqs[2:4])
DNAStringSet object of length 3:
    width seq                                                   names
[1]   248 TAGCTTTTTGACCTTCCCCTTATTTTATTTCAAAAGGTAACAGTTAGG...AAAAAGAAAAGACTGAAAAATTCGAAAATATCTCGAAGTTTAGCAGTA I:92559-92807
[2]   247 TAGCAGATGGTGGATGGTAGAGTAATGGCAGGGTAAGTGGTAGTGAAG...AGGTTGGAAGGGTTGAAATAGGGTAGGGTTAGGGTAGTGTTAGGGTAT II:5845-6092
[3]   164 TTAAATCGAAATATGTTAAGCATACAACTATTAAACCCCTTCCGTTCT...ATTATTGAAACTACTACTATGGTATAACACTATTTACAAAGAGATGAA II:111225-111389
 > width(seqs[2:4])
[1] 248 247 164
 > names(seqs[2:4])
[1] "I:92559-92807"    "II:5845-6092"     "II:111225-111389"
```