hw3_glukhov

Igor Glukhov

2022-06-07

```r
Sys.setenv(LANG = "en")
```

R Markdown

Solution to hw3

#0. Installation of RIdeogram

```r
#install.packages("RIdeogram")
```

```r
library(RIdeogram)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
```

#1. Read gene data

```r
gene_map <- read.csv('gene_mapping.tsv', sep='\t')
dong <- read.csv('DONGOLA_genes.tsv', sep='\t')
zanu <- read.csv('ZANU_genes.tsv', sep='\t')

head(gene_map)
```

```r
head(dong)
```

```
##                  ID start   end strand
## 1 gene-LOC120906950 59885 60345     -1
## 2 gene-LOC120906947 61728 64249      1
## 3 gene-LOC120906949 88010 88555     -1
## 4 gene-LOC120906948 90190 90789     -1
## 5 gene-LOC120906980   657  1316     -1
## 6 gene-LOC120906964 23986 24588      1
```

```
head(zanu)
```

```
##           ID  start    end strand
## 1 gene_13164   5022  23194     -1
## 2 gene_13165  40014  45938     -1
## 3 gene_13166  92876  97357     -1
## 4 gene_12497  99657 102434      1
## 5 gene_13167 106482 122413     -1
## 6 gene_13168 129453 131721     -1
```

1. Preprocessing

1.1 Selecting required chromosomes in mapping data

1.1.1 For gene mapping ZANU

```
unique(gene_map$contig)
```

```
##  [1] "2"                "3"                "HiC_scaffold_10"  "HiC_scaffold_104"
##  [5] "HiC_scaffold_107" "HiC_scaffold_111" "HiC_scaffold_112" "HiC_scaffold_115"
##  [9] "HiC_scaffold_122" "HiC_scaffold_127" "HiC_scaffold_129" "HiC_scaffold_139"
## [13] "HiC_scaffold_140" "HiC_scaffold_148" "HiC_scaffold_15"  "HiC_scaffold_156"
## [17] "HiC_scaffold_16"  "HiC_scaffold_17"  "HiC_scaffold_172" "HiC_scaffold_18"
## [21] "HiC_scaffold_184" "HiC_scaffold_185" "HiC_scaffold_19"  "HiC_scaffold_194"
## [25] "HiC_scaffold_195" "HiC_scaffold_196" "HiC_scaffold_203" "HiC_scaffold_204"
## [29] "HiC_scaffold_205" "HiC_scaffold_206" "HiC_scaffold_207" "HiC_scaffold_208"
## [33] "HiC_scaffold_209" "HiC_scaffold_21"  "HiC_scaffold_210" "HiC_scaffold_211"
## [37] "HiC_scaffold_212" "HiC_scaffold_213" "HiC_scaffold_214" "HiC_scaffold_215"
## [41] "HiC_scaffold_216" "HiC_scaffold_217" "HiC_scaffold_218" "HiC_scaffold_219"
## [45] "HiC_scaffold_22"  "HiC_scaffold_221" "HiC_scaffold_222" "HiC_scaffold_223"
## [49] "HiC_scaffold_224" "HiC_scaffold_225" "HiC_scaffold_23"  "HiC_scaffold_24"
## [53] "HiC_scaffold_28"  "HiC_scaffold_37"  "HiC_scaffold_38"  "HiC_scaffold_39"
## [57] "HiC_scaffold_42"  "HiC_scaffold_43"  "HiC_scaffold_45"  "HiC_scaffold_46"
## [61] "HiC_scaffold_47"  "HiC_scaffold_48"  "HiC_scaffold_49"  "HiC_scaffold_50"
## [65] "HiC_scaffold_51"  "HiC_scaffold_53"  "HiC_scaffold_58"  "HiC_scaffold_6"
## [69] "HiC_scaffold_64"  "HiC_scaffold_7"   "HiC_scaffold_70"  "HiC_scaffold_72"
## [73] "HiC_scaffold_73"  "HiC_scaffold_76"  "HiC_scaffold_77"  "HiC_scaffold_78"
## [77] "HiC_scaffold_79"  "HiC_scaffold_8"   "HiC_scaffold_81"  "HiC_scaffold_82"
## [81] "HiC_scaffold_91"  "HiC_scaffold_92"  "HiC_scaffold_99"  "X"
```

```
chr_list = c('X', '2', '3')
gene_map <- gene_map[gene_map$contig %in% chr_list,]
unique(gene_map$contig)
```

```
## [1] "2" "3" "X"
```

1.1.2 For DONGOLA in gene__mapping(seq__id -> elem in chr__list)

```
gene_map <- separate(data=gene_map, col=DONG, into=c("seq_id_dong", "mid_dong", 'strand_dong', 'len_dong
```

1.1.2.1 Process DONG column

```
seq_id_map = data.frame(id=c('2',"3","X"), val=c('NC_053517.1', 'NC_053518.1', 'NC_053519.1'))

gene_map$seq_id_dong <- with(seq_id_map, id[match(gene_map$seq_id_dong, val)])

head(gene_map)
```

1.1.2.2 Map seq__id of DONGOLA to chrososomes

```
##   contig middle.position strand ord      name ref.genes seq_id_dong  mid_dong
## 1      2           31135     -1   0 gene_3542         1           2 111908344
## 2      2           38868     -1   1 gene_3543         1           2 111899667
## 3      2           42746      1   2   gene_80         1           2 111895084
## 4      2           46243     -1   3 gene_3544         1           2 111891588
## 5      2           53442     -1   4 gene_3545         1           2 111884408
## 6      2           60574      1   5   gene_81         1           2 111877309
##   strand_dong len_dong              name_dong
## 1           1     6540 DONG_gene-LOC120894913
## 2           1     6539 DONG_gene-LOC120904110
## 3          -1     6538 DONG_gene-LOC120904105
## 4           1     6537 DONG_gene-LOC120904096
## 5           1     6536 DONG_gene-LOC120895288
## 6          -1     6535 DONG_gene-LOC120895290
```

```
gene_map <- gene_map[gene_map$seq_id_dong %in% chr_list,]
unique(gene_map$seq_id_dong)
```

1.1.2.3. Filter DONGOLA chromosomes

```
## [1] "2" "X" "3"
```

1.2 Matching gene names in gene_map and in DONGOLA frames

Removing “DONG_” from gene names in gene_map

```
gene_map$name_dong <- gsub("DONG_", "", gene_map$name_dong)
```

2. Mapping ZANU to DONGOLA genes

Firstly, since we need 1 to 1, but there is 1 to many relation, we need distance to get the closest DONGOLA genes to a given ZANU gene

2.1 Distance calculation

```
gene_map$dist <- abs(gene_map$middle.position - as.numeric(gene_map$mid_dong))

head(gene_map)
```

```
##   contig middle.position strand ord      name ref.genes seq_id_dong  mid_dong
## 1      2           31135     -1   0 gene_3542         1           2 111908344
## 2      2           38868     -1   1 gene_3543         1           2 111899667
## 3      2           42746      1   2   gene_80         1           2 111895084
## 4      2           46243     -1   3 gene_3544         1           2 111891588
## 5      2           53442     -1   4 gene_3545         1           2 111884408
## 6      2           60574      1   5   gene_81         1           2 111877309
##   strand_dong len_dong         name_dong      dist
## 1           1     6540 gene-LOC120894913 111877209
## 2           1     6539 gene-LOC120904110 111860799
## 3          -1     6538 gene-LOC120904105 111852338
## 4           1     6537 gene-LOC120904096 111845345
## 5           1     6536 gene-LOC120895288 111830966
## 6          -1     6535 gene-LOC120895290 111816735
```

2.2 Drop duplicated ZANU gene names based on dist

```
gene_map[gene_map$name == 'gene_10008', ]
```

```
##      contig middle.position strand ord       name ref.genes seq_id_dong
## 9409      3         8443412     -1 628 gene_10008         2           3
## 9410      3         8443412     -1 628 gene_10008         2           3
##      mid_dong strand_dong len_dong         name_dong     dist
## 9409 87237344           1     4092 gene-LOC120901883 78793932
## 9410 87239970           1     4093 gene-LOC120901884 78796558
```

```
gene_map_dropped <- gene_map[order(gene_map['dist',])]
```

```
## Warning in xtfrm.data.frame(x): cannot xtfrm data frames
```

```
gene_map_dropped <- gene_map[!duplicated(gene_map$name),]

gene_map_dropped[gene_map_dropped$name == 'gene_10008', ]
```

```
##      contig middle.position strand ord       name ref.genes seq_id_dong
## 9409      3         8443412     -1 628 gene_10008         2           3
##      mid_dong strand_dong len_dong         name_dong     dist
## 9409 87237344           1     4092 gene-LOC120901883 78793932
```

```
gene_map_dropped[gene_map_dropped$name == 'gene_10008', ]
```

```
##      contig middle.position strand ord       name ref.genes seq_id_dong
## 9409      3         8443412     -1 628 gene_10008         2           3
##      mid_dong strand_dong len_dong         name_dong     dist
## 9409 87237344           1     4092 gene-LOC120901883 78793932
```

3 Prepare tables (karyotype and synteny) for ideogram

3.1 Karyotype table

3.1.1 Template of data frame

```
karyotype_table <- setNames(data.frame(matrix(ncol=7, nrow=0)), c("Chr", "Start", "End", "fill", "speci

karyotype_table
```

```
## [1] Chr     Start   End     fill    species size    color  
## <0 rows> (or 0-length row.names)
```

3.1.2 Add ZENU data

```
karyotype_table <- rbind(karyotype_table, data.frame(Chr=c('X','2','3'), Start=c(1, 1, 1), End=c(272380

karyotype_table
```

```
##   Chr Start       End   fill species size  color
## 1   X     1  27238055 969696    ZANU   12 252525
## 2   2     1 114783175 969696    ZANU   12 252525
## 3   3     1  97973315 969696    ZANU   12 252525
```

3.1.3 Add DONGOLA data (lengths of chrs were googled)

```
karyotype_table <- rbind(karyotype_table, data.frame(Chr=c('X','2','3'), Start=c(1, 1, 1), End=c(269100

karyotype_table
```

```
##   Chr Start       End   fill species size  color
## 1   X     1  27238055 969696    ZANU   12 252525
## 2   2     1 114783175 969696    ZANU   12 252525
## 3   3     1  97973315 969696    ZANU   12 252525
## 4   X     1  26910000 969696 DONGOLA   12 252525
## 5   2     1 111990000 969696 DONGOLA   12 252525
## 6   3     1  95710000 969696 DONGOLA   12 252525
```

3.2 Synteny table

```
colnames(zanu) <- c('ID_1', 'Start_1', 'End_1', 'Strand_1')
colnames(dong) <- c('ID_2', 'Start_2', 'End_2', 'Strand_2')

synteny_table <- merge(gene_map_dropped, zanu, by.x='name', by.y='ID_1')
synteny_table <- merge(synteny_table, dong, by.x='name_dong', by.y='ID_2')
names(synteny_table)[names(synteny_table) == 'contig'] <- 'Species_1'
names(synteny_table)[names(synteny_table) == 'seq_id_dong'] <- 'Species_2'
synteny_table$Species_1[synteny_table$Species_1=='X'] <- 1
synteny_table$Species_1[synteny_table$Species_1=='2'] <- 2
synteny_table$Species_1[synteny_table$Species_1=='3'] <- 3
synteny_table$Species_2[synteny_table$Species_2=='X'] <- 1
synteny_table$Species_2[synteny_table$Species_2=='2'] <- 2
synteny_table$Species_2[synteny_table$Species_2=='3'] <- 3
synteny_table$Species_1 <- as.integer(synteny_table$Species_1)
synteny_table$Species_2 <- as.integer(synteny_table$Species_2)
head(synteny_table)
```

```
##            name_dong      name Species_1 middle.position strand  ord ref.genes
## 1 gene-LOC120893177 gene_5019         2        48531603     -1 2862         1
## 2 gene-LOC120893178 gene_6182         2        86040949     -1 5204         1
## 3 gene-LOC120893179 gene_2643         2        86040395      1 5203         1
## 4 gene-LOC120893180 gene_5313         2        58398932     -1 3461         1
## 5 gene-LOC120893183 gene_2537         2        82790246      1 4995         1
## 6 gene-LOC120893185 gene_6082         2        82797727     -1 4998         1
##   Species_2 mid_dong strand_dong len_dong     dist  Start_1    End_1 Strand_1
## 1         2 65514822           1     3925 16983219 48528403 48534803       -1
## 2         2 28681053           1     1788 57359896 86040710 86041188       -1
## 3         2 28681607          -1     1789 57358788 86040192 86040598        1
## 4         2 55921684           1     3534  2477248 58381587 58416277       -1
## 5         2 31941591          -1     1998 50848655 82789431 82791062        1
## 6         2 31934112           1     1995 50863615 82796508 82798947       -1
##    Start_2    End_2 Strand_2
## 1 65511152 65519724        1
## 2 28680597 28681368        1
## 3 28681316 28681908       -1
## 4 55853085 55941166        1
## 5 31940683 31942410       -1
## 6 31932898 31935462        1
```

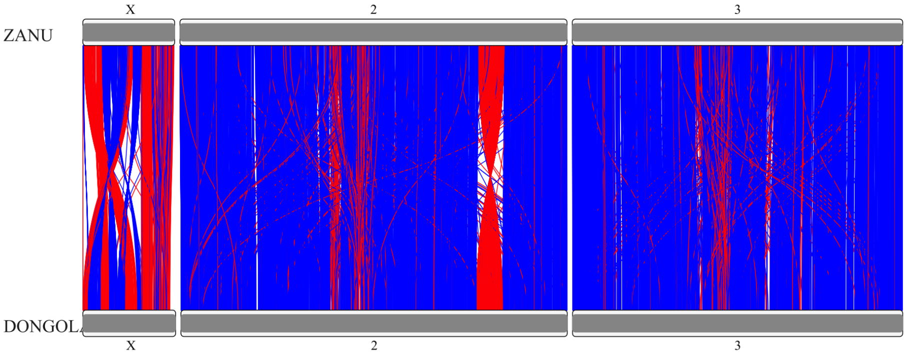
```
blue_col <- '0000FF'
red_col <- 'FF0000'
```

```
dong_max_2 <- 111990000
dong_max_3 <- 95710000

map_func <- function(strand1, strand2){
  if (strand1 == strand2)
    return(red_col)
  return(blue_col)
}

#chr 2 and chr 3 need inversion
inv_func_fill <- function(chr1, strand1, strand2, prev_fill){
  if (chr1 == 2 || chr1 == 3){
    if (strand1 == strand2)
      return(red_col)
    return(blue_col)
  }
  return(prev_fill)
}

inv_func <- function(chr1, pos2){

  if (chr1 == 2 || chr1 == 3){
    if (chr1 == 2)
      return(dong_max_2 - pos2 + 1)
    return(dong_max_3 - pos2 + 1)
  }
  return(pos2)
}


synteny_table$fill <- mapply(map_func, synteny_table$Strand_1, synteny_table$Strand_2)

synteny_table$fill <- mapply(inv_func_fill, synteny_table$Species_1, synteny_table$Strand_1, synteny_tal

synteny_table$Start_2 <- mapply(inv_func, synteny_table$Species_1, synteny_table$Start_2)

synteny_table$End_2 <- mapply(inv_func, synteny_table$Species_1, synteny_table$End_2)

synteny_table_cut <- synteny_table[c('Species_1', 'Start_1', 'End_1', 'Species_2', 'Start_2', 'End_2',



synteny_table_cut <- synteny_table_cut[synteny_table_cut$Species_1==synteny_table_cut$Species_2, ]

head(synteny_table_cut)
```

```
##   Species_1  Start_1    End_1 Species_2  Start_2    End_2   fill
## 1         2 48528403 48534803         2 46478849 46470277 0000FF
## 2         2 86040710 86041188         2 83309404 83308633 0000FF
## 3         2 86040192 86040598         2 83308685 83308093 0000FF
## 4         2 58381587 58416277         2 56136916 56048835 0000FF
## 5         2 82789431 82791062         2 80049318 80047591 0000FF
## 6         2 82796508 82798947         2 80057103 80054539 0000FF
```

4. Plot

```
ideogram(karyotype=karyotype_table, synteny=synteny_table_cut)
convertSVG("chromosome.svg", device="png")
```


ZANU
DONGOLA
X
2
3
X
2
3