BIOINFORMATICS SESSION 12. PRACTICE

2023-11-20

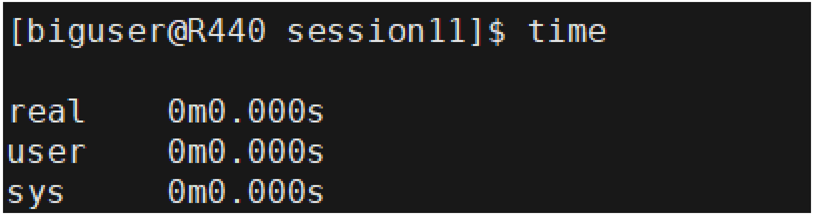
Finding genes: going ashore at CpG islands

Basic Shell Commands

```
$ cd [User_Folder]
$ mkdir session12
$ cd session12
```

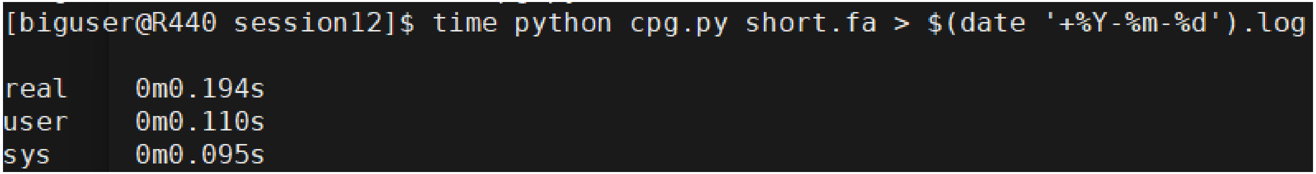
Basic Shell Commands

```
$ time
```



```
[biguser@R440 session11]$ time

real    0m0.000s
user    0m0.000s
sys     0m0.000s
```

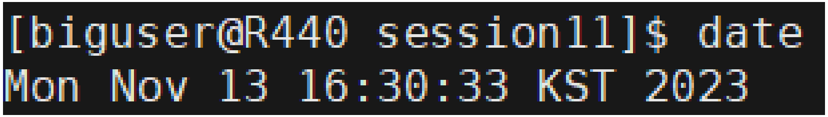
```
$ time python [script]
```



```
[biguser@R440 session12]$ time python cpg.py short.fa > $(date '+%Y-%m-%d').log

real    0m0.194s
user    0m0.110s
sys     0m0.095s
```

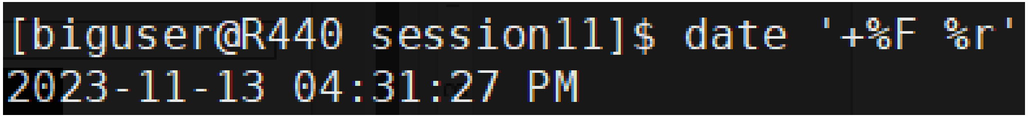
Basic Shell Commands

```
$ date                             # Current time and date
```



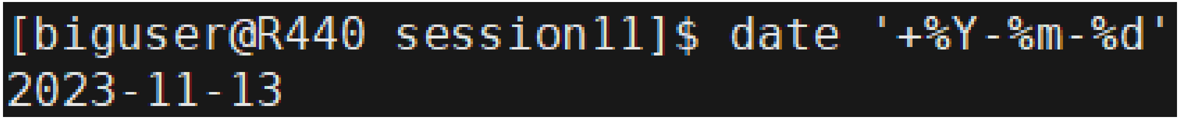
```
[biguser@R440 session11]$ date
Mon Nov 13 16:30:33 KST 2023
```

```
$ date ‘+%F %r’             # %F : YYYY-MM-DD
                              # %r : 12 hour format
```



```
[biguser@R440 session11]$ date '+%F %r'
2023-11-13 04:31:27 PM
```

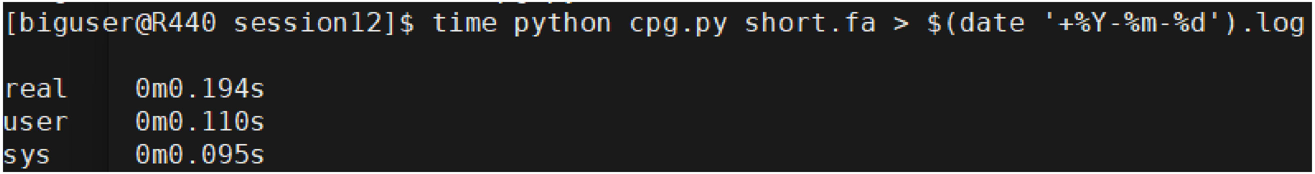
```
$ date ‘+%Y-%m-%d’                # YYYY-MM-DD (same as%F)
```



```
[biguser@R440 session11]$ date '+%Y-%m-%d'
2023-11-13
```

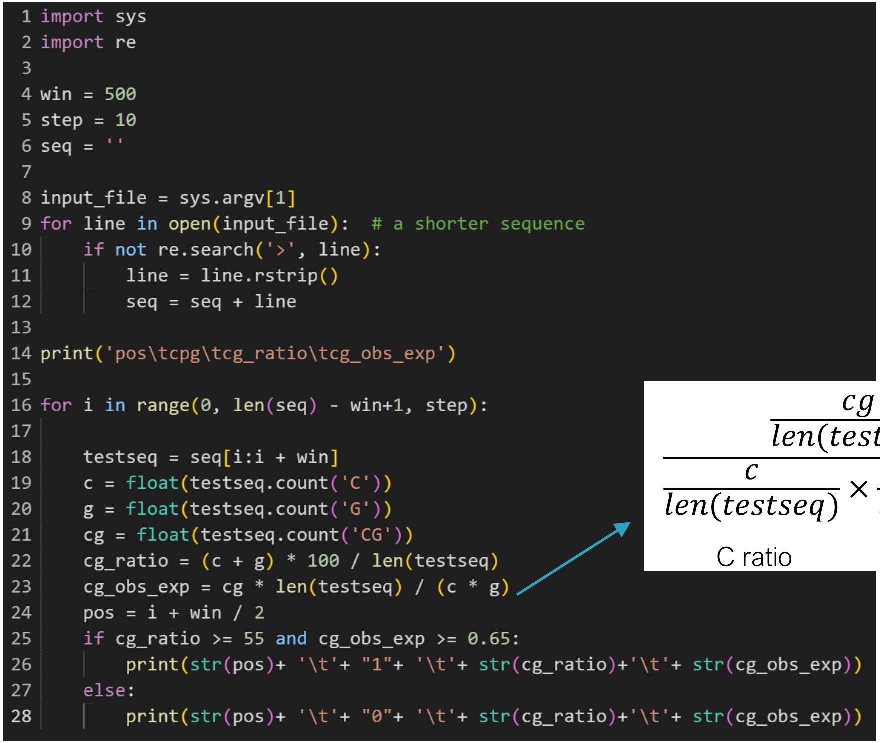
Basic Shell Commands

```
$ time python [script] > $(date ‘+%Y-%m-%d’).log
```



```
[biguser@R440 session12]$ time python cpg.py short.fa > $(date '+%Y-%m-%d').log

real	0m0.194s
user	0m0.110s
sys	0m0.095s
```

Code 15.1
cpg.py



```python
import sys
import re

win = 500
step = 10
seq = ''

input_file = sys.argv[1]
for line in open(input_file):  # a shorter sequence
    if not re.search('>', line):
        line = line.rstrip()
        seq = seq + line

print('pos\tcpg\tcg_ratio\tcg_obs_exp')

for i in range(0, len(seq) - win+1, step):

    testseq = seq[i:i + win]
    c = float(testseq.count('C'))
    g = float(testseq.count('G'))
    cg = float(testseq.count('CG'))
    cg_ratio = (c + g) * 100 / len(testseq)
    cg_obs_exp = cg * len(testseq) / (c * g)
    pos = i + win / 2
    if cg_ratio >= 55 and cg_obs_exp >= 0.65:
        print(str(pos)+ '\t'+ "1"+ '\t'+ str(cg_ratio)+'\t'+ str(cg_obs_exp))
    else:
        print(str(pos)+ '\t'+ "0"+ '\t'+ str(cg_ratio)+'\t'+ str(cg_obs_exp))
```

$$\frac{\frac{cg}{len(testseq)}}{\frac{c}{len(testseq)} \times \frac{g}{len(testseq)}}$$

CG ratio

--▶ Observed ratio

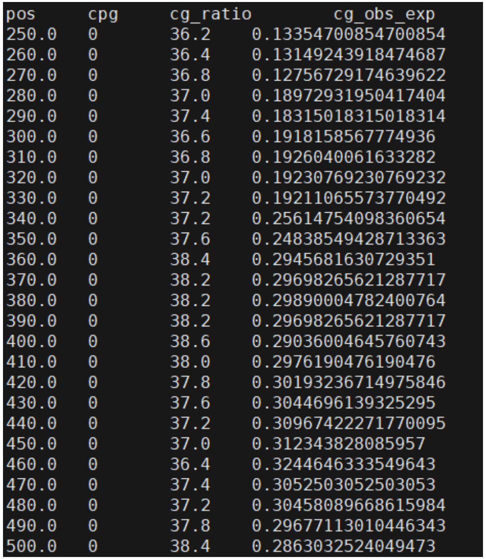
--▶ Expected ratio

C ratio

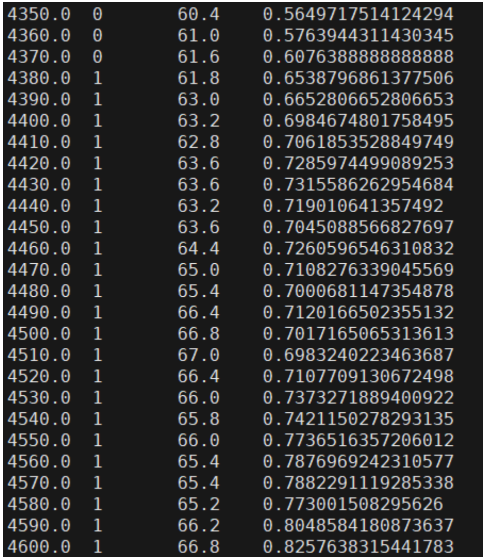
G ratio

Code 15.1
cpg.py

```
$ cp /home/biguser/tutor/session12/short.fa .
$ python cpg.py short.fa > cpg.short.out
$ less cpg.short.out
```



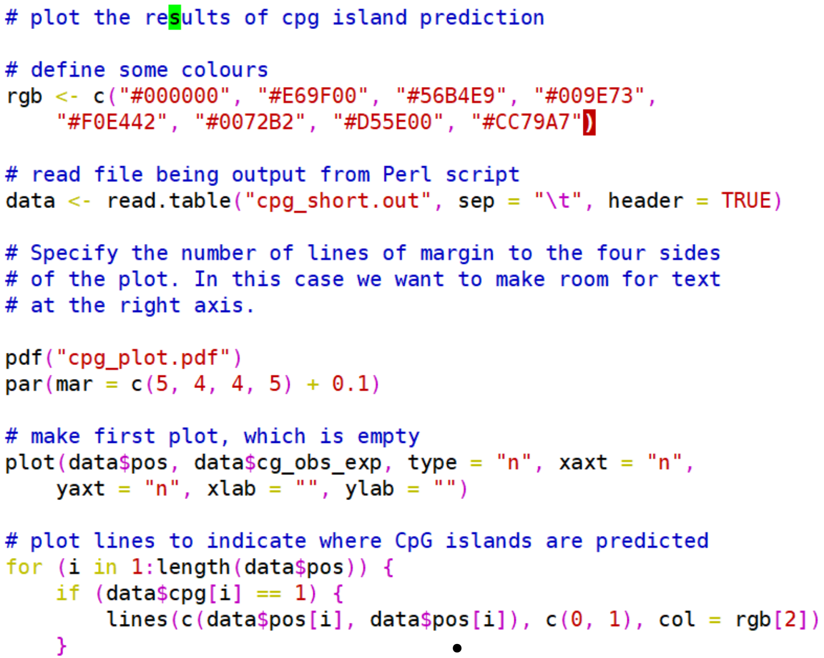
```
pos     cpg     cg_ratio        cg_obs_exp
250.0   0       36.2    0.13354700854700854
260.0   0       36.4    0.13149243918474687
270.0   0       36.8    0.12756729174639622
280.0   0       37.0    0.18972931950417404
290.0   0       37.4    0.18315018315018314
300.0   0       36.6    0.1918158567774936
310.0   0       36.8    0.1926040061633282
320.0   0       37.0    0.19230769230769232
330.0   0       37.2    0.19211065573770492
340.0   0       37.2    0.25614754098360654
350.0   0       37.6    0.24838549428713363
360.0   0       38.4    0.294568163072935
370.0   0       38.2    0.29698265621287717
380.0   0       38.2    0.29890004782400764
390.0   0       38.2    0.29698265621287717
400.0   0       38.6    0.29036004645760743
410.0   0       38.0    0.2976190476190476
420.0   0       37.8    0.30193236714975846
430.0   0       37.6    0.3044696139325295
440.0   0       37.2    0.30967422271770095
450.0   0       37.0    0.312343828085957
460.0   0       36.4    0.3244646333549643
470.0   0       37.4    0.3052503052503053
480.0   0       37.2    0.30458089668615984
490.0   0       37.8    0.29677113010446343
500.0   0       38.4    0.2863032524049473
```



```
4350.0  0       60.4    0.5649717514124294
4360.0  0       61.0    0.5763944311430345
4370.0  0       61.6    0.6076388888888888
4380.0  1       61.8    0.6538796861377506
4390.0  1       63.0    0.6652806652806653
4400.0  1       63.2    0.6984674801758495
4410.0  1       62.8    0.7061853528849749
4420.0  1       63.6    0.7285974499089253
4430.0  1       63.6    0.7315586262954684
4440.0  1       63.2    0.719010641357492
4450.0  1       63.6    0.7045088566827697
4460.0  1       64.4    0.7260596546310832
4470.0  1       65.0    0.7108276339045569
4480.0  1       65.4    0.7000681147354878
4490.0  1       66.4    0.7120166502355132
4500.0  1       66.8    0.7017165065313613
4510.0  1       67.0    0.6983240223463687
4520.0  1       66.4    0.7107709130672498
4530.0  1       66.0    0.7373271889400922
4540.0  1       65.8    0.7421150278293135
4550.0  1       66.0    0.7736516357206012
4560.0  1       65.4    0.7876969242310577
4570.0  1       65.4    0.7882291119285338
4580.0  1       65.2    0.773001508295626
4590.0  1       66.2    0.8048584180873637
4600.0  1       66.8    0.8257638315441783
```

Visualization of CPG landscape with R

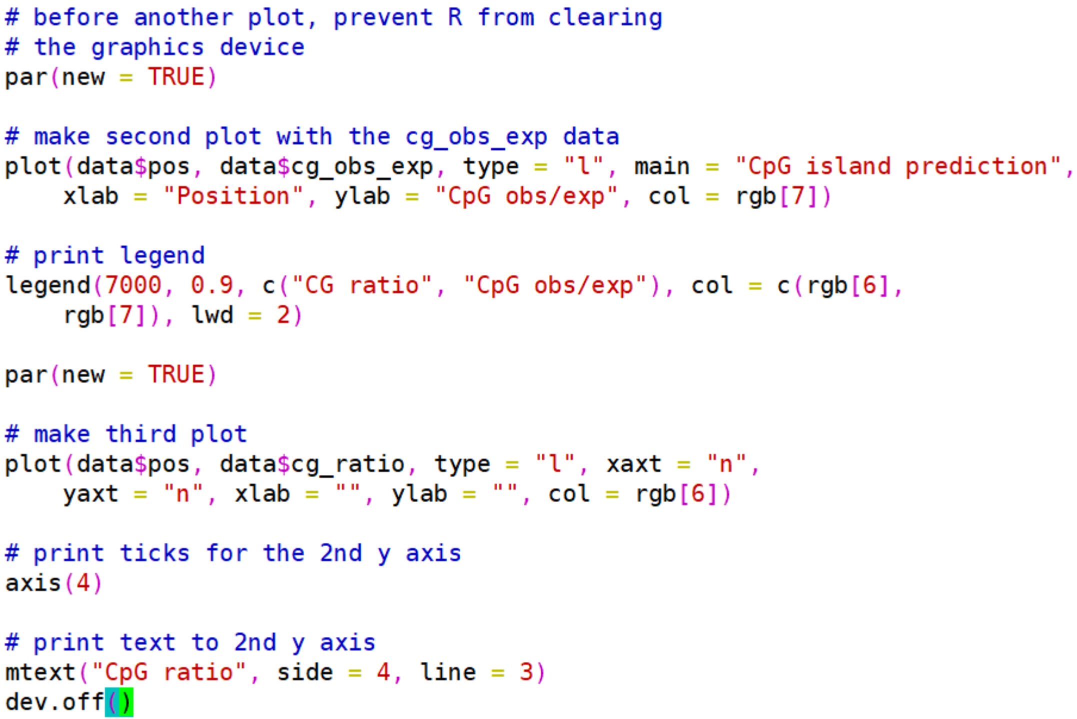
```
$ cp /home/biguser/tutor/Week12/cpg_short.r .
$ vi cpg_short.r
```



```
# plot the results of cpg island prediction

# define some colours
rgb <- c("#000000", "#E69F00", "#56B4E9", "#009E73",
    "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

# read file being output from Perl script
data <- read.table("cpg_short.out", sep = "\t", header = TRUE)

# Specify the number of lines of margin to the four sides
# of the plot. In this case we want to make room for text
# at the right axis.

pdf("cpg_plot.pdf")
par(mar = c(5, 4, 4, 5) + 0.1)

# make first plot, which is empty
plot(data$pos, data$cg_obs_exp, type = "n", xaxt = "n",
    yaxt = "n", xlab = "", ylab = "")

# plot lines to indicate where CpG islands are predicted
for (i in 1:length(data$pos)) {
    if (data$cpg[i] == 1) {
        lines(c(data$pos[i], data$pos[i]), c(0, 1), col = rgb[2])
    }
```

⋮

Visualization of CPG landscape with R

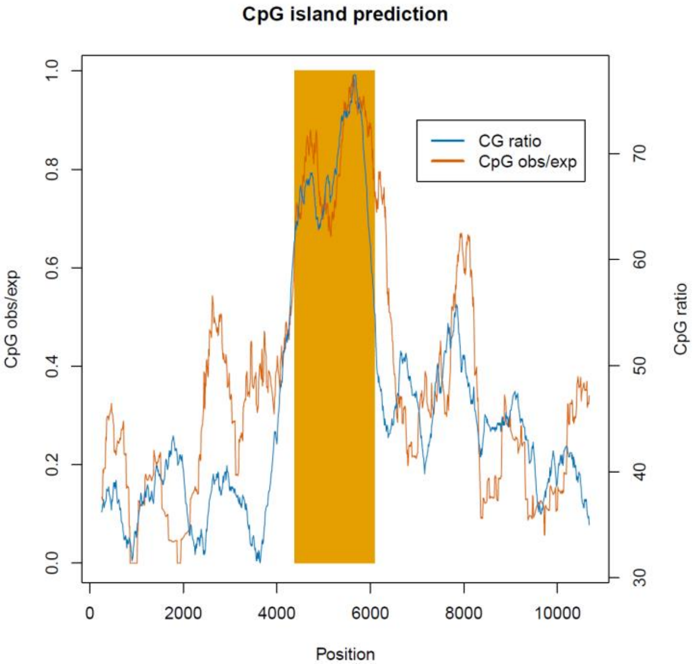


```r
# before another plot, prevent R from clearing
# the graphics device
par(new = TRUE)

# make second plot with the cg_obs_exp data
plot(data$pos, data$cg_obs_exp, type = "l", main = "CpG island prediction",
    xlab = "Position", ylab = "CpG obs/exp", col = rgb[7])

# print legend
legend(7000, 0.9, c("CG ratio", "CpG obs/exp"), col = c(rgb[6],
    rgb[7]), lwd = 2)

par(new = TRUE)

# make third plot
plot(data$pos, data$cg_ratio, type = "l", xaxt = "n",
    yaxt = "n", xlab = "", ylab = "", col = rgb[6])

# print ticks for the 2nd y axis
axis(4)

# print text to 2nd y axis
mtext("CpG ratio", side = 4, line = 3)
dev.off()
```

Visualization of CPG landscape with R

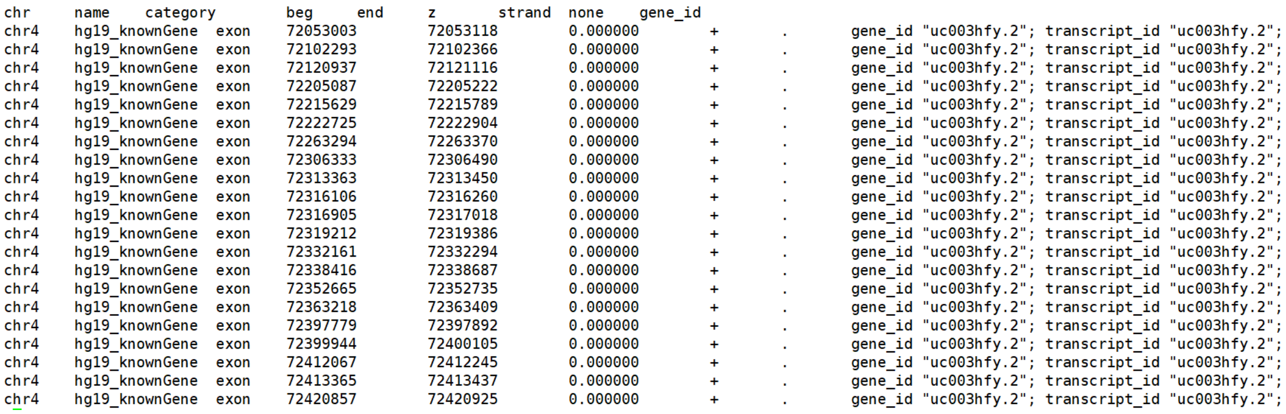
```
$ Rscript cpg_short.r
```

cpg_plot.pdf



Visualization of CPG landscape with R

```
$ cp /home/biguser/tutor/session12/chr4_region.fa .
$ python cpg.py chr4_region.fa > cpg_chr4.out
$ cp /home/biguser/tutor/session12/cpg_chr4.r .
$ cp /home/biguser/tutor/session12/chr4_annotation.txt .
$ less –S chr4_annotation.txt
```

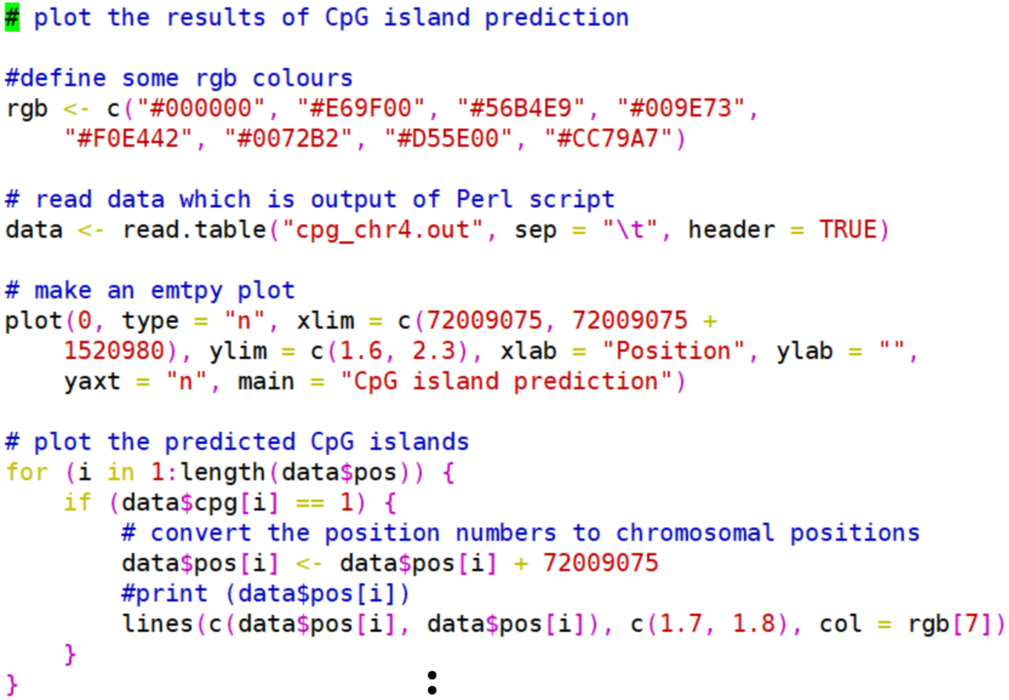


```
chr     name    category        beg     end     z       strand  none    gene_id
chr4    hg19_knownGene  exon    72053003        72053118        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72102293        72102366        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72120937        72121116        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72205087        72205222        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72215629        72215789        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72222725        72222904        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72263294        72263370        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72306333        72306490        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72313363        72313450        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72316106        72316260        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72316905        72317018        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72319212        72319386        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72332161        72332294        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72338416        72338687        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72352665        72352735        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72363218        72363409        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72397779        72397892        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72399944        72400105        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72412067        72412245        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72413365        72413437        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
chr4    hg19_knownGene  exon    72420857        72420925        0.000000        +       .       gene_id "uc003hfy.2"; transcript_id "uc003hfy.2";
```

Visualization of CPG landscape with R

Step 1: CpG island plot

```
$ vi cpg_chr4.r
```

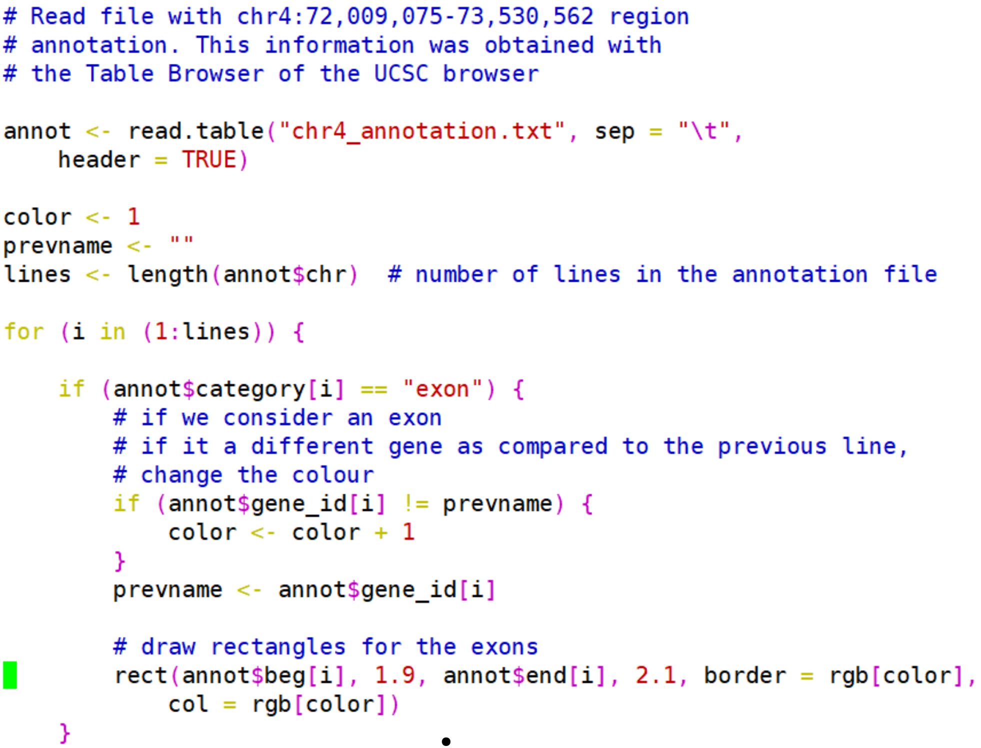


```
# plot the results of CpG island prediction

#define some rgb colours
rgb <- c("#000000", "#E69F00", "#56B4E9", "#009E73",
    "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

# read data which is output of Perl script
data <- read.table("cpg_chr4.out", sep = "\t", header = TRUE)

# make an emtpy plot
plot(0, type = "n", xlim = c(72009075, 72009075 +
    1520980), ylim = c(1.6, 2.3), xlab = "Position", ylab = "",
    yaxt = "n", main = "CpG island prediction")

# plot the predicted CpG islands
for (i in 1:length(data$pos)) {
    if (data$cpg[i] == 1) {
        # convert the position numbers to chromosomal positions
        data$pos[i] <- data$pos[i] + 72009075
        #print (data$pos[i])
        lines(c(data$pos[i], data$pos[i]), c(1.7, 1.8), col = rgb[7])
    }
}
```

⋮

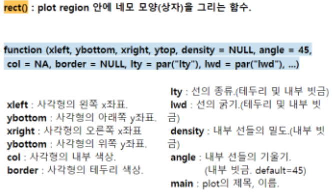
Visualization of CPG landscape with R

Step 2: Gene structure plot



```r
# Read file with chr4:72,009,075-73,530,562 region
# annotation. This information was obtained with
# the Table Browser of the UCSC browser

annot <- read.table("chr4_annotation.txt", sep = "\t",
    header = TRUE)

color <- 1
prevname <- ""
lines <- length(annot$chr)  # number of lines in the annotation file

for (i in (1:lines)) {

    if (annot$category[i] == "exon") {
        # if we consider an exon
        # if it a different gene as compared to the previous line,
        # change the colour
        if (annot$gene_id[i] != prevname) {
            color <- color + 1
        }
        prevname <- annot$gene_id[i]

        # draw rectangles for the exons
        rect(annot$beg[i], 1.9, annot$end[i], 2.1, border = rgb[color],
            col = rgb[color])
    }
```

⋮



rect() : plot region 안에 네모 모양(상자)을 그리는 함수.

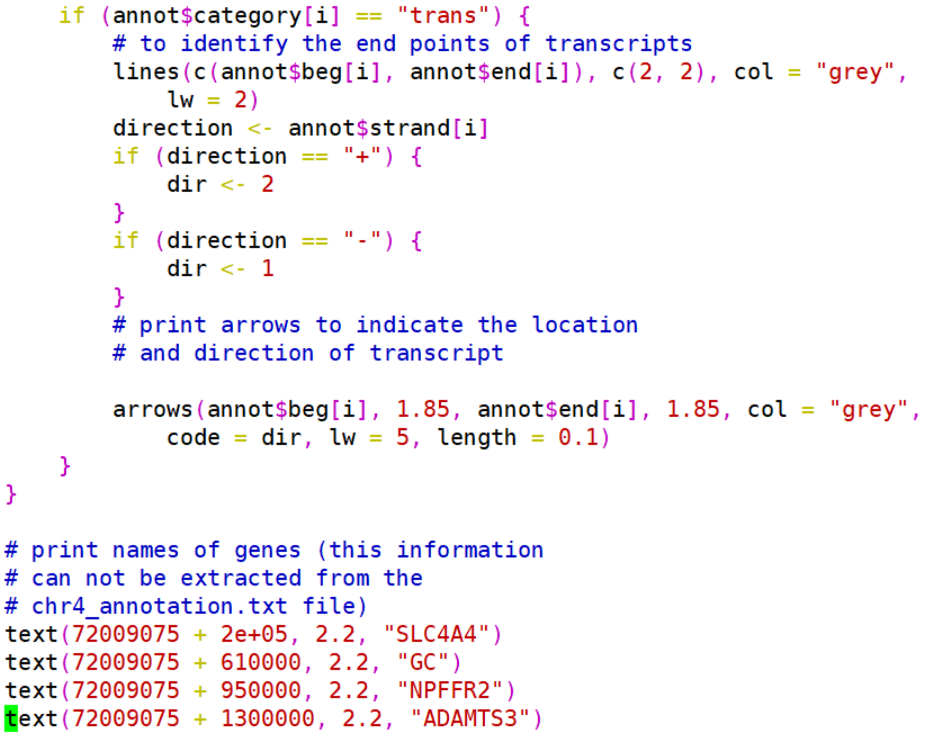
```
function (xleft, ybottom, xright, ytop, density = NULL, angle = 45,
 col = NA, border = NULL, lty = par("lty"), lwd = par("lwd"), ...)
```

xleft : 사각형의 왼쪽 x좌표.
ybottom : 사각형의 아래쪽 y좌표.
xright : 사각형의 오른쪽 x좌표
ybottom : 사각형의 위쪽 y좌표.
col : 사각형의 내부 색상.
border : 사각형의 테두리 색상.

lty : 선의 종류.(테두리 및 내부 빗금)
lwd : 선의 굵기.(테두리 및 내부 빗금)
density : 내부 선들의 밀도.(내부 빗금)
angle : 내부 선들의 기울기.
(내부 빗금. default=45)
main : plot의 제목, 이름.

Visualization of CPG landscape with R

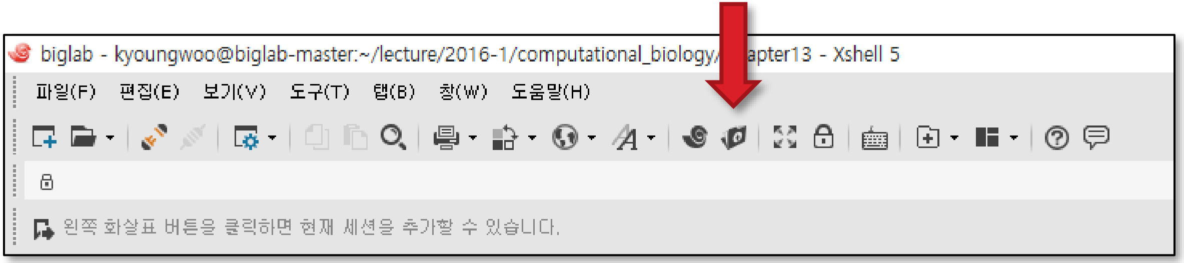
Step 2: Gene structure plot



```
  if (annot$category[i] == "trans") {
    # to identify the end points of transcripts
    lines(c(annot$beg[i], annot$end[i]), c(2, 2), col = "grey",
      lw = 2)
    direction <- annot$strand[i]
    if (direction == "+") {
      dir <- 2
    }
    if (direction == "-") {
      dir <- 1
    }
    # print arrows to indicate the location
    # and direction of transcript

    arrows(annot$beg[i], 1.85, annot$end[i], 1.85, col = "grey",
      code = dir, lw = 5, length = 0.1)
  }
}

# print names of genes (this information
# can not be extracted from the
# chr4_annotation.txt file)
text(72009075 + 2e+05, 2.2, "SLC4A4")
text(72009075 + 610000, 2.2, "GC")
text(72009075 + 950000, 2.2, "NPFFR2")
text(72009075 + 1300000, 2.2, "ADAMTS3")
```

Visualization of CPG landscape with R

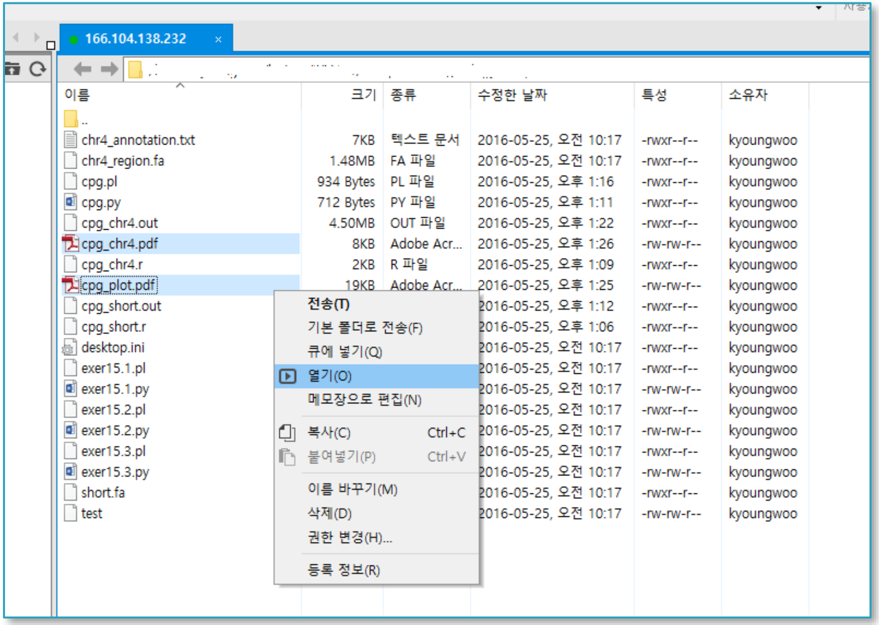
```
$ Rscript cpg_chr4.r
```



biglab - kyoungwoo@biglab-master:~/lecture/2016-1/computational_biology/ apter13 - Xshell 5

왼쪽 화살표 버튼을 클릭하면 현재 세션을 추가할 수 있습니다.

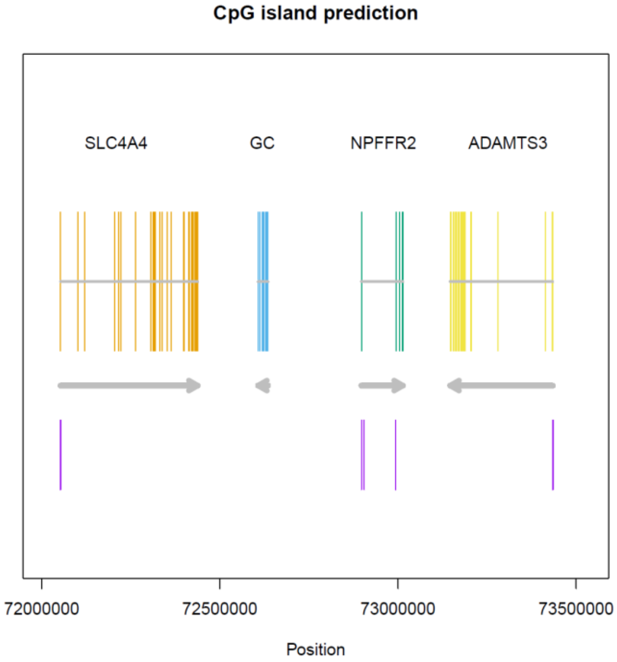
Visualization of CPG landscape with R



이름	크기	종류	수정한 날짜	특성	소유자
..					
chr4_annotation.txt	7KB	텍스트 문서	2016-05-25, 오전 10:17	-rwxr--r--	kyoungwoo
chr4_region.fa	1.48MB	FA 파일	2016-05-25, 오전 10:17	-rwxr--r--	kyoungwoo
cpg.pl	934 Bytes	PL 파일	2016-05-25, 오후 1:16	-rwxr--r--	kyoungwoo
cpg.py	712 Bytes	PY 파일	2016-05-25, 오후 1:11	-rwxr--r--	kyoungwoo
cpg_chr4.out	4.50MB	OUT 파일	2016-05-25, 오후 1:22	-rwxr--r--	kyoungwoo
cpg_chr4.pdf	8KB	Adobe Acr...	2016-05-25, 오후 1:26	-rw-rw-r--	kyoungwoo
cpg_chr4.r	2KB	R 파일	2016-05-25, 오후 1:09	-rwxr--r--	kyoungwoo
cpg_plot.pdf	19KB	Adobe Acr...	2016-05-25, 오후 1:25	-rw-rw-r--	kyoungwoo
cpg_short.out			2016-05-25, 오후 1:12	-rwxr--r--	kyoungwoo
cpg_short.r			2016-05-25, 오후 1:06	-rwxr--r--	kyoungwoo
desktop.ini			2016-05-25, 오전 10:17	-rwxr--r--	kyoungwoo
exer15.1.pl			2016-05-25, 오전 10:17	-rwxr--r--	kyoungwoo
exer15.1.py			2016-05-25, 오전 10:17	-rw-rw-r--	kyoungwoo
exer15.2.pl			2016-05-25, 오전 10:17	-rwxr--r--	kyoungwoo
exer15.2.py			2016-05-25, 오전 10:17	-rw-rw-r--	kyoungwoo
exer15.3.pl			2016-05-25, 오전 10:17	-rwxr--r--	kyoungwoo
exer15.3.py			2016-05-25, 오전 10:17	-rw-rw-r--	kyoungwoo
short.fa			2016-05-25, 오전 10:17	-rwxr--r--	kyoungwoo
test			2016-05-25, 오전 10:17	-rw-rw-r--	kyoungwoo

Visualization of CPG landscape with R

cpg_chr4.pdf



Assignment

- In the output from Code 15.1 there is a “1” in the second column in the case that the position is part of a CpG island. Design a python script that uses the output file “cpg_chr4.out” and prints the begin and end positions of the different CpG islands. For instance, the first CpG island that should be printed is 43070-44360

- cpg_chr4.out의 column 2번 0 / 1 정보를 활용하여 CpG island의 시작과 끝 position을 출력하는 python 코드를 구현하세요.

- 과제 제출 기한: 11/26 Sunday 23:59 @ LMS
- 작성한 코드와 해당 코드의 결과를 캡처한 뒤 워드에 첨부(코드만 긁어와서 붙여넣지 말기), 코드에 대한 설명 간략히 작성
  워드 파일명은 n주차_학번_이름 형식으로 제출(e.g. 12주차_2023123456_김현우)