Basic R Coding

H Qin

5/18/2020

loops

```
rm(list=ls())
 for ( i in 1:3) print(i);
```

```
## [1] 1
## [1] 2
## [1] 3
```

```
 for ( i in c(0.5, 1, -2, 5) ) print(i);
```

```
## [1] 0.5
## [1] 1
## [1] -2
## [1] 5
```

```
 for ( i in c(0.5, 1, -2, 5) ) {
   x = i * i
   print( sqrt(x))
 }
```

```
## [1] 0.5
## [1] 1
## [1] 2
## [1] 5
```

```
i=2;
 while( i<=10 ) {
  print(i);
  i = i + 2;
 }
```

```
## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 10
```

conditionals

```
x = 0.9;
if ((x<1) & (x>0)) {
   print("X is between 0 and 1 ");
} else {
   print("X is not between 0 and 1");
}
```

```
## [1] "X is between 0 and 1 "
```

```
x =0.5;
if ( ( ! x>1) & ( ! x<0 ) ) {
   print("x is between 0 and 1");
}
```

```
## [1] "x is between 0 and 1"
```

```
if ( x == 0.5 ) {
   print("50%");
}
```

```
## [1] "50%"
```

```
x= -1;
if ( ! x<0 ) {
  print("non-negative");
} else {
  print("negative");
}
```

```
## [1] "negative"
```

```
x = c(-1, 3)
x = c( 1, NA, 3, 4, 5, 2)
y = ifelse( is.na(x), median(x, na.rm = T), x );
# assign abosulte value of x to y
print(y)
```

```
## [1] 1 3 3 4 5 2
```

```
#### a function
x = c( 1,2,3,4, 7, 9); #this is a global variable
#x = c(5, 4, 3, 6, 10, 12)

y = "I am global"
```

```
take_even = function( x ) {
  y = c(); # a locale copy inside of loop
  for( i in 1:length(x)) {
    if ( (x[i] %% 2 )== 0 ) { # x mod 2
        y = c(y, x[i] );  # add a new x[i] to y
    }
  }
  print ("inside y");
  print(y)
  print("end of inside")
  y; #the last line, return y to the main program
}

take_even(x); # c(2,4)
```

```
## [1] "inside y"
## [1] 2 4
## [1] "end of inside"
```

```
## [1] 2 4
```

```r
#y =  take_even(x); # c(2,4)

z = take_even(x); # c(2,4);
```

```
## [1] "inside y"
## [1] 2 4
## [1] "end of inside"
```

```r
w = take_even(1:10); #c (2,4,6,8,10)
```

```
## [1] "inside y"
## [1]  2  4  6  8 10
## [1] "end of inside"
```

```r
u = -5:10;
v = take_even(u); # -4 -2  0  2  4  6  8 10
```

```
## [1] "inside y"
## [1] -4 -2  0  2  4  6  8 10
## [1] "end of inside"
```

```r
take_even(c(5,4,10,11, 100, -3, 9, 0, 3333, 5524234234))
```

```
## [1] "inside y"
## [1]          4         10        100          0 5524234234
## [1] "end of inside"
```

```
## [1]          4         10        100          0 5524234234
```

```r
#### a function calls another function
even_total = function( x ) {
  x = take_even(x);
  sum(x)
 }

x = c( 1,2,3,4, 0, -2, -4 );
u = even_total( x );
```

```
## [1] "inside y"
## [1]  2  4  0 -2 -4
## [1] "end of inside"
```

```r
print(u)
```

```
## [1] 0
```

```r
# x will be changed if the following line is run.
x = take_even(x)
```

```
## [1] "inside y"
## [1]  2  4  0 -2 -4
## [1] "end of inside"
```

```r
 #### scope of variable
 x = c( 1,2,3,4); #this is a global variable
 y = c(-1, -2, -3, -4)# this is also a global one
 y.outside = y;
```

```
take_uphalf = function( x ) {
  y = c(); # a local copy inside of loop
  xbar = mean(x);
  for( i in 1:length(x)) {
    if ( x[i] > xbar ) {
      y = c(y, x[i] );
    }
  }
  y.inside = y;
  print( paste("y.inside=", y.inside) );
  y; #the last line, return y to the main program
}

w = take_uphalf(x);
```

```
## [1] "y.inside= 3" "y.inside= 4"
```

```
mean(x)
```

```
## [1] 2.5
```

packages

```
#install packages
 require("ape");
```

```
## Loading required package: ape
```

```
require("seqinr");
```

```
## Loading required package: seqinr
```

```
##
## Attaching package: 'seqinr'
```

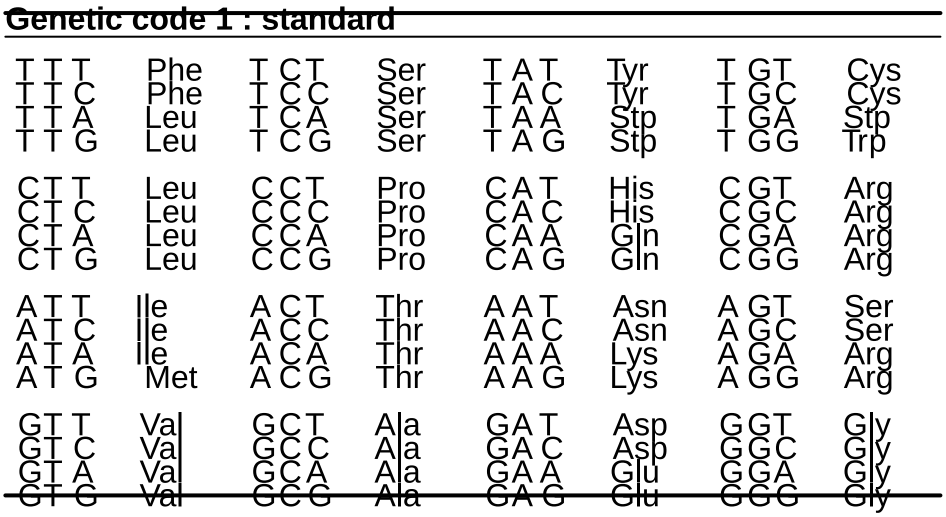
```
## The following objects are masked from 'package:ape':
##
##     as.alignment, consensus
```

```
 #load packages
 library(ape);
 library(seqinr);

 #what's in the package?
 help(package=ape);
 help(package=seqinr);
  #you should also read the manual of the packages
```

```
#### a simple example
 rm(list = ls())
 library(seqinr);

 tablecode(); #the genetic code
```



Genetic code 1 : standard

T T T	Phe	T C T	Ser	T A T	Tyr	T G T	Cys
T T C	Phe	T C C	Ser	T A C	Tyr	T G C	Cys
T T A	Leu	T C A	Ser	T A A	Stp	T G A	Stp
T T G	Leu	T C G	Ser	T A G	Stp	T G G	Trp
C T T	Leu	C C T	Pro	C A T	His	C G T	Arg
C T C	Leu	C C C	Pro	C A C	His	C G C	Arg
C T A	Leu	C C A	Pro	C A A	Gln	C G A	Arg
C T G	Leu	C C G	Pro	C A G	Gln	C G G	Arg
A T T	Ile	A C T	Thr	A A T	Asn	A G T	Ser
A T C	Ile	A C C	Thr	A A C	Asn	A G C	Ser
A T A	Ile	A C A	Thr	A A A	Lys	A G A	Arg
A T G	Met	A C G	Thr	A A G	Lys	A G G	Arg
G T T	Val	G C T	Ala	G A T	Asp	G G T	Gly
G T C	Val	G C C	Ala	G A C	Asp	G G C	Gly
G T A	Val	G C A	Ala	G A A	Glu	G G A	Gly
G T G	Val	G C G	Ala	G A G	Glu	G G G	Gly

```
# read in some bacterial 16s rDNA sequences
seqs = read.fasta( "http://www.bioinformatics.org/ctls/download/data/16srDNA.fasta",seqtype="DNA");

# look at the first sequence
seq1 = seqs[[1]]
table( seq1 ); #nucleotide composition
```

```
## seq1
##   a   c   g   n   t
## 365 325 432  28 300
```

```
GC(seq1);  # GC content
```

```
## [1] 0.5323488
```

```
# a loop for all sequences
num = 1:length(seqs);
gc  = 1:length(seqs);
out = data.frame( cbind( num, gc ) );
out$samples = "TMP"

for( i in 1:length(seqs) ) {
  out$gc[i] = GC( seqs[[i]] );
  out$samples[i] = attr(seqs[[i]], "name")
}

out;
```

```
##   num        gc           samples
## 1   1 0.5323488       B_anthracis
## 2   2 0.5333333          B_cereus
## 3   3 0.5440613      B_halodurans
## 4   4 0.5505618  Bacillus_subtilis
## 5   5 0.5175439 C_acetobutyliticum
## 6   6 0.5291777     C_perfringenes
## 7   7 0.5278515          C_tetani
## 8   8 0.5287356        O_iheyensis
## 9   9 0.5913556    T_tengcongensis
```

```r
 write.csv(out, "gc.csv", row.names=F) # output the results


#### save and load images
 save.image("Test.RData");

 # delete everyting
 rm(list=ls());

 ls(); #nothing
```

```
## character(0)
```

```r
  load("Test.RData");
 ls(); #everthing is back
```

```
## [1] "gc"   "i"    "num"  "out"  "seq1" "seqs"
```