Universitat Autonoma de Barcelona, Module 01 Fall 2016
MSc in Bioinformatics

Perl challenges

Edgar G Nogales, 1441015

11/08/2016

This work is personal and the student must submit the source code developed to the Masters web: mscbioinformatics.uab.es Remember to upload all your answers in a single text file with the format: NIU_perl_version.txt. DO NOT UPLOAD .PL FILES!

1 Problem Q1

Rearrange a fasta file so that each sequence occupies only one line. Open a fasta input file and write an output file with ID and sequence in two consecutive lines. Solution:

```
open(my $fh, "<", $ARGV[0]) or die "Can't open the file!"
    ;
my $seq = "";
my $first = 1;
while (my $row = <$fh>) {
        if($row =~ m/^>(.*)/){
                if (!$first){

                        #to avoid print the previous
                    seq for the 1st line
                        print "$seq\n";

                                        #print the seq
                            before
```

```
                $seq="";
            }

                                        #ini seq
                string
            my ($seqname, $seqinfo) = split / /,
                $row, 2;
            print "$seqname\n";

                        #print seqname of actually seq
            $first=0;
        }else{
            chomp($row);

                        #delte \n
            $seq .= $row;

                        #adding sequences
        }
}
print "$seq\n";
                                        #print last
    seq
```

2 Problem Q2

Create a script that receives a name as a parameter, a protein fasta file, and find which proteins (provide the protein id) contain a person name in their sequence. Use names without B,J,O,U,X or Z to test the script.
Solution:

```
my ($name) = @ARGV;
open(PROTEIN, "<example.fasta") or die "Can't open the
    file!";
while(my $line=<PROTEIN>){
        if($line =~ m/^>(.*)/){
            ($seqname, $seqinfo) = split / /, $line,
                2;
        }
        if($line =~ /$name/){
            print "result: $seqname\n";
        }
}
```

3 Problem Q3

Create a script that provides all IDs of a fasta file given as the input.
Solution:

```
open(FASTA, "<example.fasta") or die "Can't open the file
    !";
while(my $line=<FASTA>){
        if($line =~ m/^>(.*)/){
                ($seqname, $seqinfo) = split / /, $line,
                    2;
                print "$seqname \n";
        }
}
```

4 Problem Q4

Considering the file example.bed from the Dropbox folder, how would get the smallest exon size from each of the records? The result should provide a number for each line of the input.
Solution:

```
use List::Util qw( min );

open(BED, "<example.bed") or die "Can't open the file!";

while($line=<BED>){
        @var = split (/\t/, $line);
        @col = split (/,/, $var[10]);
        my $min = min @col;
        print "$min\n";
}
```

5 Problem Q5

Provide the top 10 mRNA ID with the largest exon sizes of the example.bed file.
Solution:

```
use List::Util qw( max );

open(BED, "<example.bed") or die "Can't open the file!";
```

```
my @exons =();
$max = 10 -1;
$i=0;   #var para el index in while
while($line=<BED>){
        @var = split (/\t/,  $line);
        $exons[$i][0] = $var[3];
        $exons[$i][1] = max(split (/,/, $var[10]));
        $i=$i+1;
}
@filtered = sort { $b->[1] <=> $a->[1] } @exons;

for (my $i=0; $i <= $max; $i++) {
   print "$filtered[$i][0] - $filtered[$i][1]\n";
}
```

6 Problem Q6

Create a script that generates a randomly generated FASTA file. Parameters are: number of sequences to generate and number of nucleotides per line.
You must build each id and each random sequence made of a combination of "ACGTN" of the required length. An example of the expected output is:
>sequence1
ACGT
Solution:

```
my $outfile = "nuevo.fasta";
my $numberSeq = $ARGV[0];
my $numberNucleo = $ARGV[1];

#print "$numberSeq \n";
#print "$numberNucleo \n";
my @chars=('A','C','G','T','N');

open(MYFILE, ">>$outfile") or die "Can't open the
    $outfile!\n";
#print MYFILE "Bob\n";
for (my $i=1; $i <= $numberSeq; $i++) {
   #print ">sequence$i\n";
   print MYFILE ">sequence$i\n";
   my $random_string;
   foreach (1..$numberNucleo) {
                $random_string.=$chars[rand @chars];
        }
```

```
        #print "$random_string\n";
        print MYFILE "$random_string\n";
}
close (MYFILE)
```

7 Problem Q7

Find palindromes in the sequences of a given fasta file. Program must find how many sequences have palindromes of a minimum length of 6 characters, and store the top 10 more frequent palindromes found in the input fasta file.
Only character-by- character palindromes with 0 or 1 central character must be considered. For example: radar, level, rotor, noon, ACTGGTCA or GGAGG.
Solution:

```
#!/usr/bin/perl
open(FASTA, "<example.fasta") or die "Can't open the file
    !";
%data = ();
my @palindromes = ();
@names = ();
my $maxElements=10;
#my $re;
$re = qr/((.)(?:(??{$re})|.?)\2)/;
while($_=<FASTA>){
        if($_ =~ m/^>(.*)/){
                ($seqname, $seqinfo) = split / /, $_, 2;
        }else{
        chomp($_);
        push @palindromes, "$1" while (/(?=$re)/g);
        foreach $pali (@palindromes){
          if(length($pali)>=6){
          if (!grep( /^$seqname$/, @names)) {
              push @names, "$seqname";
          }
          if(exists($data{$pali})){
              $data{$pali} += 1;
          }
          else{
              $data{$pali} = 1;
          }
        }
        }
  }
}
```

```
$i=0;
foreach my $name (sort { $data{$b} <=> $data{$a} } keys %
    data) {
     if($i<$maxElements){
          printf "%-8s %s\n", $name, $data{$name};
          $i++;
     }
}
print "**** SEQNAMES **** \n";
print join(", ", @names),"\n";
```

8 Problem Q8

Join all six previous exercises in a single program that calls the needed functionality by using a command parameter:
Solution:
Attached file with the code.