```perl
#!/usr/bin/perl
use strict;
use warnings;
use File::Basename;

#################################################
# SCRIPT TO CALCULATE SOME GENE PARAMETERS -> 3d codon position only !
#################################################
##### Spring 2016 - MLS - UNIL - Marie Zufferey
# USAGE :
# ./stat_CDS_3dpos.pl  my_cds.fasta outfile.txt
#    -> my_cds.fasta  => file with the CDS
#    -> outfile.txt   => file in which output written !!!! if already exist, will be overwritten !
# output as follow (tab-separated):
#    -> the tag of the sequence, its length, the ratio of purine and the ratio of GC-content (for the
# 3d codon position only):
#    SEQ_TAG     LENGTH      ratioGC      ratioPu

my $cds = $ARGV[0];
my $outfile = $ARGV[1];


print("WARNING: outfile will be overwritten !\n");
system("rm -f $outfile");
system("touch $outfile");

# write the header in the output file
open(my $out, '>>', $outfile) or die("open: $$");
print($out "Seq_tag\tn3dpos\tratioGC\tratioPu\n");
close($out);

my ($id, $seq, $nG, $nC, $nA, $size, $nPu, $nGC);

open(CDS, $cds) || die;
while(my $line = <CDS>){
    chomp($line);
    if($line =~ /^>/){  # => it is the ID line

        if(length $seq){          # we have a new ID (all IDs except the 1st)

            # select every 3d character only
            $seq =~ s/..(.)/$1/g;
            # calculate and write the sequence information in output file
            $nG = ($seq =~ tr/gG//);
            $nC = ($seq =~ tr/cC//);
            $nA = ($seq =~ tr/aA//);
            $size = length($seq);
            $nPu = ($nA+$nG)/$size;
            $nGC = ($nC+$nG)/$size;
            open(my $out, '>>', $outfile) or die("open: $$");
            print($out "$id\t$size\t$nGC\t$nPu\n" );
            close($out);

            # reinitialize the seq
            $seq = "";

        }
        ($id = $line) =~ s/^>//; # store the id without the > that starts the line

    } else{         # => it is one seq line (can be separated with \n ...)
        $seq .= $line;  # concatenate the sequence parts

    }
}
# do not forget the last sequence  # TODO put in subroutine to avoid repetition of code...

# calculate and write the sequence information in output file
$seq =~ s/..(.)/$1/g;
$nG = ($seq =~ tr/gG//);
$nC = ($seq =~ tr/cC//);
$nA = ($seq =~ tr/aA//);
$size = length($seq);
$nPu = ($nA+$nG)/$size;
$nGC = ($nC+$nG)/$size;
```

```perl
open($out, '>>', $outfile) or die("open: $$");
print($out "$id\t$size\t$nGC\t$nPu\n" );
close($out);

close(CDS);


#foreach my $pos(@query_id){
#    print("$pos\n");
 #   open(my $queryF, '>>', $query_file) or die("open: $$");
 #   print($queryF ">$pos\n$all_seq{$pos}\n" );
#    print $x;
#}
```