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

##############################################
# SCRIPT TO PARSE BLAT RESULTS
##############################################
##### Spring 2016 - MLS - UNIL - Marie Zufferey
# Script to use after having run blat, e.g.:
#blat -t=dna -q=dna -noHead Pf5_cds_foo.fasta ../PseudS5_query.fasta
# USAGE EXAMPLE:
# ./parse_psl.pl ../data/BLAT_OUTPUT.psl ../data/Pf5_cds.fasta annotation.csv Pf5

my $blat_result = $ARGV[0];
my $db_fasta = $ARGV[1];
my $annotDB = $ARGV[2];
my $nameDB = $ARGV[3];
my $command = "";
my $outfile = "S5vs$nameDB.txt";

##############
# RETRIEVE ANNOTATIONS/FUNCTIONS FROM THE DB (.CSV)
(my $annotDBcut = $annotDB) =~ s/.csv/_cut.csv/;
$command = "cut -d \",\" -f3,9 $annotDB | tail -n +4 > $annotDBcut";  # first 3 lines are comments
system($command);
my %dbID_fction;
my $function ="";
open(CSV, $annotDBcut) || die;
while(my $line = <CSV>){
    chomp($line);
    $line =~ s/\"//g;  # not forget the g for global !
    my @line = split /,/, $line;
    my $geneID = $line[0];
    my $function = $line[1];


    if(not length $function){
        $function = "NA";
    }
    if(exists($dbID_fction{$geneID}) and $dbID_fction{$geneID} ne "NA"){
        my $prev_function = $dbID_fction{$geneID};
        my $new_function = "$prev_function,$function";  # if already exist => function1,function2
        $dbID_fction{$geneID} = $new_function;

    }else{
        $dbID_fction{$geneID} = $function;    # but some $geneID comme more than one time !!!
    }

}
close(CSV);


(my $blat_result_sorted = $blat_result) =~ s/.psl/_sorted.psl/;

system("sort -k 1 -r -n $blat_result > $blat_result_sorted");

### create the file
# S5_genome_id  $nameDB  Function
#S5_genome   PFL_0001    replication initiator


open(my $tempf, '>', $outfile) or die("open: $$");
my $first_line = "S5_genome_id\t$nameDB\tFunction\n";
print($tempf $first_line);


open(PSL, $blat_result_sorted) || die;

while(my $line = <PSL>){
    chomp($line);
    my @line = split /\t/, $line;
    my $queryID = $line[9];                # id from our genome (S5_genome_87)
    my $dbIDnr = $line[13];                    # => but the ID we retrieve for the DB is not the
```

```
appropriate one (e.g. 1893893)

    $command = "grep \">$dbIDnr\\s\" $db_fasta";  # retrieve the gene id for the number of the DB
    my $retrieveID = `$command`;                 # => >1893893 gene=PFL_0001  # in the csv we have
PFL_0001

    $retrieveID =~ s/^\s+|\s+$//g;           #trim

    my @fastaID = split /=/, $retrieveID;
    my $dbID = $fastaID[1];

    if(exists($dbID_fction{$dbID})){
        print ($tempf "$queryID\t$dbID\t$dbID_fction{$dbID}\n");
    } else {
        print ($tempf "$queryID\t$dbID\tNA\n");
       print "not found";
    }

}
close($tempf);
close(PSL);
```