

```

#! C:/Programmi/Perl/bin/perl -w

use Tk;
require Tk::Text;
require Tk::Button;
require Tk::Scrollbar;

$main = new MainWindow (-title => 'Unique');
$main->geometry("515x400");
$main->resizable('0','0');
$maintext = $main->Text(height=>20, -width=>70)->place(-x=>2, -y=>2);
$maintext->menu(undef);
$main->Button(-text=>'Load Sequence',
             -command=>\&opensequence)->place(-x=>2, -y=>370);
$main->Button(-text=>'Paste sequence',
             -command=>\&fix)->place(-x=>2, -y=>340);
$main->Button(-text=>'Scan sequence',
             -command=>\&analyze)->place(-x=>405, -y=>290);
$main->Label(-text=>'Scanning window:')->place(-x=>2, -y=>290);
$entrywin = $main->Entry(-width=>3)->place(-x=>95, -y=>290);
$main->Label(-text=>'Repetitions allowed:')->place(-x=>130, -y=>290);
$entrythreshold = $main->Entry(-width=>3)->place(-x=>233, -y=>290);
$scroll = $main->Scrollbar (-orient=>'vertical',
                          -command => [yview=>$maintext])->place(-x=>498, -y=>1, -
                          height=>288);
$maintext->configure (-yscrollcommand => [set => $scroll]);
$reportbutton = $main->Button(-text=>'Show Report',
                             -command=>\&reporter);

MainLoop;

sub opensequence {
    if ($seq) {
        undef($seq);
        undef($count);
    }
    if ($sequence) {
        undef($sequence);
        undef($counter);
        if ($seq) {
            undef($seq);
            undef($count);
        }
    }
    my $type = [['Plasmid Sequence', ['.txt']]];
    $file = $main->getOpenFile(-filetypes=>$type);
    if(!$file) {return 'false';}

    open (FILE, "< $file");
    do {$char = getc(FILE);
        if ('ATGCNatkcn' =~ $char) {
            $seq .= uc($char);
            $count++;
        }
    } until eof(FILE);
    close (FILE);
    $sequence = $seq;
    $counter = $count;
    $maintext->Contents($sequence);
}

sub fix {
    if ($sequence) {
        undef($sequence);
        undef($counter);
    }
    if ($tempseq) {
        undef($tempseq);
    }
}

```

```

        undef($sequence);
        undef($counter);
    }
    $tempseq = $main->SelectionGet(-selection=>'CLIPBOARD');
    $maintext->Contents($tempseq);
    return 'false' if ($tempseq !~ /\w/);
    $tempseq =~ s/[\d\s]//g;
    $sequence = uc($tempseq);
    @temp = split(//,$sequence);
    $counter = @temp;
    undef ($tempseq);
    undef (@temp);
    $maintext->Contents($sequence);
}

sub analyze {
    if (!$sequence) {
        $main->Dialog(-title=>'Error', -text =>'Load DNA sequence first...')->Show;
        return 'false';
    }
    $win = $entrywin->get;
    $thereshold = $entrythereshold->get;
    if (!$win) {
        $main->Dialog(-title=>'Error',
            -text =>'Select scanning window...')->Show;
        return 'false';
    }
    if (!$thereshold) {
        $main->Dialog(-title=>'Error',
            -text =>'Select repetition thereshold..')->Show;
        return 'false';
    }
    if ($allresults) {
        undef ($allresults);
        undef ($noresults);
        undef ($fewresults);
        undef ($palinresults);
        $lab1->placeForget;
        $lab2->placeForget;
        $lab3->placeForget;
        $lab4->placeForget;
        if($hidereport) {
            $hidereport->placeForget;
        }
    }
    &hidereport;
    $allresults = "\n";
    $scanned = 0;
    $norip = 0;
    $majrip = 0;
    $minrip = 0;
    $palrip = 0;
    @base=split(//,"$sequence");
    for ($j=0;$j<$win;$j++) {
        $fillextremity .= $base[$j];
    }
    $seqforsearch = $sequence.$fillextremity;
    undef($fillextremity);
    @bases=split(//,"$seqforsearch");
    for ($r=0;$r<=#base;$r++) {
        $seqtosearch = &defineseq($r);
        $startfrom ="1.0";
        if ($allresults !~ $seqtosearch) {
            @result = &finder($seqtosearch);
        }
    }
    if ($result[2]) {
        $allresults.= "$response position $wherefound\n";
        if ($result[2] eq "norepetitions") {
            if ($result[3]) {
                $palinresults.= "$response position $wherefound\n";
            }
        }
    }
}

```

```

    }
    $noresults.= "$response position $wherefound\n";
}
if ($result[2] eq "fewrepetitions") {
    if ($result[3]) {
        $palinresults.= "$response position $wherefound\n";
    }
    $fewresults.= "$response position $wherefound\n";
}
}
undef (@result);
}
$report1 = "$majrip sequences with over thereshold repetitions found\n";
$report2 = "$minrip sequences under thereshold found\n";
$report3 = "$norip sequences with no repetitions found\n";
$report4 = "$palrip palindromic sequences found\n";

$maintext->Contents($sequence);
$lab1 = $main->Label(-text=>"$report1")->place(-x=>110, -y=>325);
$lab2 = $main->Label(-text=>"$report2")->place(-x=>110, -y=>340);
$lab3 = $main->Label(-text=>"$report3")->place(-x=>110, -y=>355);
$lab4 = $main->Label(-text=>"$report4")->place(-x=>110, -y=>370);
}

sub defineseq {
    if ($seqtoanalyze) {undef($seqtoanalyze);}
    $scanned++;
    @start = @_;
    for ($i=$start[0];$i<$start[0]+$win;$i++) {
        $seqtoanalyze .= $bases[$i];
    }
    return ($seqtoanalyze);
}

sub finder {
    if($wherefound) {
        undef($wherefound);
    }
    $maintext->Contents("");
    $maintext->Contents($seqforsearch);
    $times = 0;
    @what = @_;
    if ($win%2 == 0) {
        $pal = ($win/2)-1;
        $palin = &palindrom($what[0]);
    }
    sub palindrom {
        @seque = @_;
        if ($palindrome) {
            undef($palindrome);
        }
        @pali = split (//,"$seque[0]");
        $s=0;
        $e=$#pali;
        while ($s<=$pal) {
            $com = &comp($pali[$s]);
            if ($com ne $pali[$e]) {
                $ans = 'no';
                return $ans;
            }
            else {
                $ans = 'yes';
            }
            $s++;
            $e--;
        }
        return $ans;
    }

    sub comp {
        @who = @_;

```

```

        if ($who[0] eq "A") {
            $temp = "T";
        }
        if ($who[0] eq "T") {
            $temp = "A";
        }
        if ($who[0] eq "G") {
            $temp = "C";
        }
        if ($who[0] eq "C") {
            $temp = "G";
        }
        return $temp;
    }
}
&find;
sub find {
while (!$foundat) {
    $foundat = $maintext->search($what[0], $startfrom, 'end');
    if ($foundat) {
        @newindex = split (/\.\/, "$foundat");
        $nextchar = $newindex[1]+1;
        $startnew = "1.".$nextchar;
        $startfrom = $startnew;
        if ($newindex[1]<$counter) {
            $wherefound .= $nextchar." ";
            $times++;
        }
        undef($foundat);
    }
    else {return 'false'}
}
}
if ($times == 1) {
    if ($palin) {
        if ($palin eq "no") {
            $response = "$what[0] - no repetitions,";
        }
        else {
            $response = "$what[0] - (palindromic) no repetitions,";
            $palrip++;
            $palindrome = 1;
        }
    }
    else {
        $response = "$what[0] - no repetitions,";
    }
    $norip++;
    $kind = "norepetitions";
    return ($response, $wherefound, $kind, $palindrome);
}
if ($times <= $thereshold) {
    if ($palin) {
        if ($palin eq "no") {
            $response = "$what[0] - present $times times,";
        }
        else {
            $response = "$what[0] - (palindromic) present $times times,";
            $palrip++;
            $palindrome = 1;
        }
    }
    else {
        $response = "$what[0] - present $times times,";
    }
    $minrip++;
    $kind = "fewrepetitions";
    return ($response, $wherefound, $kind, $palindrome);
}
else {

```

```

        $majrip++;
        $response = "$what[0] is present a number over thereshold times";
        return ($response, $wherefound);
    }
}

sub reporter {
    $main->geometry("515x630");
    $reportertext = $main->Text(height=>10, -width=>70)->place(-x=>2, -y=>410);
    $reporterscroll = $main->Scrollbar (-orient=>'vertical',
        -command => [yview=>$reportertext])->place(-x=>
            498, -y=>409, -height=>144);
    $reportertext->configure (-yscrollcommand => [set => $reporterscroll]);
    $reportertext->Contents($allresults);
    $reportbutton->placeForget;
    $main->Button(-text=>'No repetitions', -command=>\&showno)->place(-x=>2, -y=>560);
    $main->Button(-text=>'Under thereshold', -command=>\&showfew)->place(-x=>92, -y=>560
    );
    $main->Button(-text=>'All Results', -command=>\&showall)->place(-x=>202, -y=>560);
    $main->Button(-text=>'Palindromic', -command=>\&showpali)->place(-x=>282, -y=>560);
    $main->Button(-text=>'Sort', -command=>\&sorter)->place(-x=>2, -y=>600);
    $main->Button(-text=>'Locate', -command=>\&locate)->place(-x=>52, -y=>600);
    $hidereport = $main->Button(-text=>'Hide Report', -command=>\&hidereport)->place(-x=
        >435, -y=>370);
    $main->Button(-text=>'Save Report', -command=>\&savereport)->place(-x=>435, -y=>600)
    ;
}

sub locate {
    if ($start) {
        $maintext->tagDelete('ILLO');
    }
    ($start,$end) = $reportertext->tagRanges('sel');
    return 'false' if (!$start || !$end);
    $selected = $reportertext->get($start,$end);
    $from = '1.0';
    while (!$foundat) {
        $foundat = $maintext->search($selected,$from,'end');
        if ($foundat) {
            @newindex = split (/\.\/, "$foundat");
            $nextchar = $newindex[1]+1;
            $foundwin = $newindex[1]+$win;
            $startnew = "1.".$nextchar;
            $selection = "1.".$foundwin;
            $from = $startnew;
            $maintext->tagAdd('ILLO', $foundat, $selection);
            $maintext->tagConfigure('ILLO', -background=>'yellow');
            undef($foundat);
        }
        else {return 'false'}
    }
}

sub sorter {
    if ($sortedseq) {
        undef($sortedseq);
    }
    $stosort = $reportertext->get('1.0','end');
    @lines = split (/\\n/, "$stosort");
    @sorted = sort {$a cmp $b} @lines;
    for ($i=0;$i<@sorted;$i++) {
        $sortedseq .= $sorted[$i]."\n";
    }
    $reportertext->Contents($sortedseq);
}

sub savereport {
    $header = "UNIQUE software results on $counter base long sequence\n Scanning window:"

```

```
    $win bases\n Windows scanned: $scanned\n Repetition thereshold: $thereshold\n\n"
;
$reports = $report1.$report2.$report3.$report4;
$data = $reportertext->get ('1.0','end');
$reportcomplete = $header.$reports.$data;
$savefile = $main->getSaveFile;
if(!$savefile){return 'false';}
open (SAVEFILE, "> $savefile");
print {SAVEFILE} "$reportcomplete";
close (SAVEFILE);
}

sub showno {
    $reportertext->Contents($noresults);
}

sub showpali {
    $reportertext->Contents($palinresults);
}

sub showfew {
    $reportertext->Contents($fewresults);
}

sub showall {
    $reportertext->Contents($allresults);
}

sub hidereport {
    $main->geometry("515x400");
    if ($hidereport) {
        $hidereport->placeForget;
    }
    $reportbutton->place(-x=>435, -y=>370);
}
```