

```

#! perl -w

use Tk;
require Tk::Text;
require Tk::Button;
require Tk::Scrollbar;
require Tk::BrowseEntry;
require Tk::CheckBox;
require Tk::Canvas;

open (REST,"< restlist.txt");
do {$rest .=getc(REST);} until eof(REST);
@line = split (/\\n/, $rest);
$pi='3.1415';
$main = new MainWindow (-title => 'DNA Fragmentator');
$main->geometry("515x350");
$main->resizable('0','0');
$maintext = $main->Text(height=>20, -width=>70)->place(-x=>2, -y=>2);
$Open = $main->Button(-text=>'Load Sequence',
                    -command=>\\&opensequence)->place(-x=>2, -y=>320);
$Paste = $main->Button(-text=>'Paste sequence',
                    -command=>\\&fix)->place(-x=>100, -y=>320);
$Screening = $main->Button(-text=>'Screen Enzymes',
                        -command=>\\&screen);
$New = $main->Button(-text=>'Start a new job',
                  -command=>\\&newjob);
$main->Label(-text=>'DNA type: Linear')->place(-x=>220, -y=>325);
$check2 = $main->Checkbutton(-variable=>\\$type, -onvalue=>'linear')->place(-x=>305, -y=>320);
$main->Label(-text=>'Plasmid')->place(-x=>330, -y=>325);
$check1 = $main->Checkbutton(-variable=>\\$type, -onvalue=>'plasmid')->place(-x=>370, -y=>320);
$check1->select;
$Combine = $main->Button(-text=>'Mix Enzymes', -state=>'disabled',
                      -command=>\\&add)->place(-x=>435, -y=>322);

$main->Label(-text=>'Enzyme 1')->place(-x=>2, -y=>294);
$Browse1 = $main->BrowseEntry(-label=>'name ', -width=>7, -state=>'disabled',
                            -variable=>\\$val1, -browsecmd=>sub {treat('1')})->place(-x=>50, -y=>293);
$entry1 = $main->Entry(-width=>10)->place(-x=>160, -y=>294);

$main->Label(-text=>'Enzyme 2')->place(-x=>267, -y=>294);
$Browse2 = $main->BrowseEntry(-label=>'name ', -width=>7, -state=>'disabled',
                            -variable=>\\$val2, -browsecmd=>sub {treat('2')})->place(-x=>320, -y=>293);

```

```

$entry2 = $main->Entry(-width=>10)->place(-x=>430, -y=>294);

$scroll = $main->Scrollbar (-orient=>'vertical',
                          -command => [yview=>$maintext])->place(-x=>498, -y=>1, -height=>288);

$maintext->configure (-yscrollcommand => [set => $scroll]);

$cont = 0;
while ($cont<=$#line){
    @temp = split (/ /,$line[$cont]);
    $seq[$cont] = $temp[1];
    $nam[$cont] = $temp[0];
    $Browse1->insert($cont,$temp[0]);
    $Browse2->insert($cont,$temp[0]);
    $cont++;
}

MainLoop;

sub treat {
    @which = @_ ;
    $val = $val1 if ($which[0] == 1);
    $val = $val2 if ($which[0] == 2);
    $cont = 0;
    while ($cont<=$#line){
        $found = $cont if ($val eq $nam[$cont]);
        $cont++;
    }
    $entry1->delete(0,'end') if ($which[0] == 1);
    $entry1->insert(0,$seq[$found])if ($which[0] == 1);
    $entry2->delete(0,'end') if ($which[0] == 2);
    $entry2->insert(0,$seq[$found])if ($which[0] == 2);
    &analyse("$which[0]");
}

sub opensequence {
    my $type = [['DNA Sequence', ['.txt']]];
    $file = $main->getOpenFile(-filetypes=>$type);
    if(!$file) {return 'false';}
    open (FILE, "< $file");
    do {$char =getc(FILE);
        $directseq .= $char;
    }
}

```

```

} until eof(FILE);
@lines = split(/\n/, $directseq);
shift (@lines) if ($lines[0] =~ '>');
foreach $lin (@lines) {
    @temp = split(//, $lin);
    foreach $base (@temp) {
        if ('ATGCNatgcn' =~ $base) {
            $seq .= uc($base);
            $count++;
        }
    }
}
close (FILE);
$sequence = $seq;
undef($seq);
$counter = $count;
undef($count);
$maintext->Contents($directseq);
undef($directseq);
$Browse1->configure(-state=>'normal');
$Browse2->configure(-state=>'normal');
$Open->placeForget;
$Paste->placeForget;
$New->place(-x=>2, -y=>320);
$Screening->place(-x=>100, -y=>320);
}

sub fix {
    $pastedseq = $main->SelectionGet(-selection=>'CLIPBOARD');
    $maintext->Contents($pastedseq);
    return 'false' if ($pastedseq !~ /\w/);
    $pastedseq =~ s/[\\d\\s]//g;
    $sequence = uc($pastedseq);
    @temp = split(//, $sequence);
    $counter = @temp;
    undef ($pastedseq);
    undef (@temp);
    $Browse1->configure(-state=>'normal');
    $Browse2->configure(-state=>'normal');
    $Open->placeForget;
    $Paste->placeForget;
    $New->place(-x=>2, -y=>320);
}

```

```

    $Screening->place(-x=>100, -y=>320);
}

sub newjob {
    $maintext->Contents('');
    $Browse1->configure(-state=>'disabled');
    $Browse2->configure(-state=>'disabled');
    $newwin[1]->destroy if (Exists($newwin[1]));
    $newwin[2]->destroy if (Exists($newwin[2]));
    $newwin[3]->destroy if (Exists($newwin[3]));
    undef($val1) if ($val1);
    undef($val2) if ($val2);
    $entry1->delete(0,'end');
    $entry2->delete(0,'end');
    $New->placeForget;
    $Screening->placeForget;
    $Open->place(-x=>2, -y=>320);
    $Paste->place(-x=>100, -y=>320);
    $Combine->configure(-state=>'disabled');
    $check1->configure(-state=>'normal');
    $check2->configure(-state=>'normal');
}

sub analyse {
    @onetwo = @_ ;
    $check1->configure(-state=>'disabled');
    $check2->configure(-state=>'disabled');
    $choice = $onetwo[0];
    if ($resultrest) {
        undef (@restsite);
        undef (@fragment);
        undef ($resultrest);
        undef ($resultfrag);
        undef ($partial);
        undef ($q);
    }
    if ($choice == 1 && Exists($newwin[1])) {
        $newwin[1]->destroy;
    }
    if ($choice == 2 && Exists($newwin[2])) {
        $newwin[2]->destroy;
    }
}

```

```

$site = $entry1->get if ($choice == 1);
$site = $entry2->get if ($choice == 2);
#$site is already defined if $choice = 3
$site[$choice] = $site; #sets enzyme specific restricion site
$site = uc($site);
undef ($true[$choice]) if ($true[$choice]);
@place = split (//,$site);
$i=0;
foreach $letter (@place) {
    $plus = $i if ($letter eq '*');
    $i++ if ($letter ne '*');
    $true[$choice] .= $letter if ($letter ne '*');
}
$minus = $i - $plus;
$k = 0;
$q = 0;
while ($sequence =~ /$true[$choice]/g) {
    $q=1;
    $restsite[$k] = pos $sequence;
    $restsite[$k] -= $minus;
    $fragment[$k] = $restsite[$k]-$restsite[$k-1] if $restsite[$k-1];
    $partial += $fragment[$k] if $fragment[$k];
    $k++
}
undef (@fragment) if ($#fragment == 0);
$partial += $restsite[0] if ($type eq 'linear' && @fragment);
$fragment[$k] = $counter-$partial if ($q !=0 && @fragment);
unshift(@fragment,$restsite[0]) if ($type eq 'linear' && @fragment);
splice(@fragment,1,1) if ($type eq 'linear' && @fragment);
$tot = $partial+$fragment[$k] if ($q !=0 && @fragment);
shift(@fragment) if ($type eq 'plasmid' && @fragment);
$cutnumber = @restsite;
@firstrestriction = @restsite if ($choice == 1 && $q !=0);
@secondrestriction = @restsite if ($choice == 2 && $q !=0);
$Combine->configure(-state=>'normal') if (@firstrestriction && @secondrestriction);
&windows("$choice");
}

sub add {
return 'false' if ($vall eq $val2);
if (@fusion) {
    undef(@fusion);
}
}

```

```

    undef(@sortedrestsite);
    undef(@purged);
    undef(@newfragment);
    undef($newpartial);
    undef(@sortedfrag);
    undef($true[3]);
}
if (Exists($newwin[3])) {
    $newwin[3]->destroy;
}
$val = "$val1 and $val2";
$site = "$site[1] and $site[2]";
>true[3] = "$true[1] and $true[2]";
@fusion = (@firstrestriction,@secondrestriction);
@sortedrestsite = sort {$a <=> $b} @fusion;
for ($j=0; $j<=#sortedrestsite;$j++) {
    if ( !$sortedrestsite[$j+1] || ($sortedrestsite[$j] != $sortedrestsite[$j+1])) {
        push(@purged, $sortedrestsite[$j]);
    }
}
$cutnumber = @purged;
$newpartial = 0;
for ($j=0; $j<=#purged;$j++) {
    $newfragment[$j] = $purged[$j+1]-$purged[$j] if ($purged[$j+1]);
    $newpartial += $newfragment[$j] if ($j<=#purged);
}
$newpartial += $purged[0] if ($type eq 'linear');
$lastpartial = $counter-$newpartial;
$tot = $newpartial+$lastpartial;
push (@newfragment,$lastpartial);
unshift(@newfragment,$purged[0]) if ($type eq 'linear');
@fragment = @newfragment;
@restsite = @purged;
$q = 1;
&windows('3');
}

#inizio della parte grafica

sub windows {
    @_index = @_;
    $ind = $index[0];

```

```

$newwin[$ind] = $main->Toplevel(-title=>"$val analysis ($true[$ind])");
$newwin[$ind]->resizable('0','0');
$newtext = $newwin[$ind]->Text(-width=>40, -height=>15)->pack(side=>'left', -fill=>'both');
if ($cutnumber > 4) {
    $newsroll = $newwin[$ind]->Scrollbar(-orient=>'vertical',
                                        -command => [yview=>$newtext])->pack(-side=>'left',-fill=>'both');
    $newtext->configure(-yscrollcommand => [set => $newsroll]);
}
$resultrest = "Restriction analysis of\n  $val ($site)\n\n";
if ($q == 0) {
    $resultrest .= "$val does not cut the sequence!\n";
    $newtext->Contents($resultrest);
}
else {
    $resultrest .= "$site found $cutnumber times\n\n";
    if ($type eq 'plasmid') {
        for ($j=1; $j<=$cutnumber;$j++) {
            $resultrest .= "Site n°$j: $restsite[$j-1]\n" if ($restsite[$j-1]);
            $resultrest .= "          Fragment $j lenght: $fragment[$j-1]\n" if (@fragment);
        }
    }
    if ($type eq 'linear') {
        for ($j=1; $j<=$cutnumber+1;$j++) {
            $resultrest .= "          Fragment $j lenght: $fragment[$j-1]\n" if (@fragment);
            $resultrest .= "Site n°$j: $restsite[$j-1]\n" if ($restsite[$j-1]);
        }
    }
    $newtext->Contents($resultrest);
    if (@fragment) {
        @sortedfrag = sort {$b <=> $a} @fragment;
        $newtext1 = $newwin[$ind]->Text(-width=>16, -height=>15)->pack(side=>'left', -fill=>'both');
        if ($#fragment > 13) {
            $newsroll1 = $newwin[$ind]->Scrollbar(-orient=>'vertical',
                                                -command => [yview=>$newtext1])->pack(-side=>'left',-fill=>'both');
            $newtext1->configure(-yscrollcommand => [set => $newsroll1]);
        }
        $resultfrag = "Sorted fragments\n\n";
        for ($j=0; $j<=$#sortedfrag;$j++) {
            $resultfrag .= "          $sortedfrag[$j]\n";
        }
        $newtext1->Contents($resultfrag);
    }
}

```

```

if ($type eq 'plasmid') {
    $plasmid= $newwin[$ind]->Canvas(-width=>250, -height=>230)->pack(-side=>'left');
    $plasmid->createOval(40,30,220,210, -outline=>'blue');
    $plasmid->createLine(130,28,130,32, -fill=>'yellow', -width=>2);
    $plasmid->createText(135,120, -font=> 'Times', -text=>"Total lenght\n      $counter");
    for ($j=0; $j<=$cutnumber-1;$j++) {
        $threshold = $fragment[$j]/$counter if (@fragment);
        $threshold = 1 if (!@fragment);
        $rad = -(((($restsite[$j]/$counter)*360+180)*$pi); #trasforma la frazione frammento/totale in radianti e aggiusta
        $x1[$j] = sin(($rad)/180)*85+130; #aggiusta il punto di fine riga su x
        $y1[$j] = cos(($rad)/180)*85+120; #aggiusta il punto di fine riga su y
        $x2[$j] = sin(($rad)/180)*95+130; #aggiusta il punto di fine riga su x
        $y2[$j] = cos(($rad)/180)*95+120; #aggiusta il punto di fine riga su y
        $x3[$j] = sin(($rad)/180)*100+130 if ($threshold > 0.02);
        $x3[$j] = sin(($rad)/180)*80+130 if ($threshold <= 0.02);
        $y3[$j] = cos(($rad)/180)*100+120 if ($threshold > 0.02);
        $y3[$j] = cos(($rad)/180)*80+120 if ($threshold <= 0.02);
    }
    for ($j=0; $j<=$cutnumber-1;$j++) {
        $truej = $j+1;
        $plasmid->createText($x3[$j],$y3[$j], -font=> 'Times',-text=>"$truej", -fill=>'black');
        $plasmid->createLine($x1[$j],$y1[$j],$x2[$j],$y2[$j], -fill=>'red');
    }
}
if ($type eq 'linear') {
    $plasmid= $newwin[$ind]->Canvas(-width=>250, -height=>100)->pack(-side=>'left');
    $plasmid->createLine(5,40, 250, 40, -fill=>'blue');
    for ($j=0; $j<$cutnumber;$j++) {
        $truej = $j+1;
        $threshold = $fragment[$j+1]/$counter if (@fragment);
        $threshold = 1 if !@fragment;
        $x = ($restsite[$j]/$counter)*245+5;
        $plasmid->createLine($x,37,$x,43, -fill=>'red');
        $plasmid->createText($x,48, -font=> 'Times',-text=>"$truej") if ($threshold && $threshold > 0.05);
        $plasmid->createText($x,32, -font=> 'Times',-text=>"$truej") if ($threshold && $threshold <= 0.05);
        $plasmid->createText($x,32, -font=> 'Times',-text=>"$truej") if (!$threshold);
    }
    $plasmid->createText(120,80, -font=> 'Times', -text=>"Total lenght: $counter");
}
}
}

```

```
sub screen {
  foreach $enz (@line) {
    @cut = split (/ /, $enz);
    $cut[1] =~ s/\*//;
    while ($sequence =~ /$cut[1]/gi) {$k++;}
    $output .= "$cut[0]: $k site/s\n";
    $k=0;
  }
  $screen = $main->Toplevel(-title=>'Screening');
  $screen->resizable('0','0');
  $screentext = $screen->Text(-width=>20, -height=>45)->pack(side=>'left', -fill=>'both');
  $screenscroll = $screen->Scrollbar(-orient=>'vertical',
    -command => [yview=>$screentext])->pack(-side=>'left',-fill=>'both');
  $screentext->configure(-yscrollcommand => [set => $screenscroll]);
  $screentext->Contents($output);
  undef($output);
}
```