

```

#! perl -w

use Tk;
require Tk::BrowseEntry;
require Tk::Menu;
require Tk::Dialog;
require Tk::Text;
require Tk::Scrollbar;
require Tk::Checkbutton;
require Tk::Radiobutton;

$h="direct";
$k="reverse";
$inifile = "Primer Wizard.ini";
&initialize;
$main = new MainWindow (-title => 'Primers Wizard');
$main->geometry("700x470");
$main->resizable('0','0');
$menubar = $main->Menu;
$main->configure(-menu=>$menubar);
$file = $menubar->cascade(-label=>'File', -tearoff=>0);
    $file->command (-label=>'Load DNA sequence', -command => \&loaddnasequence);
    $file->command (-label=>'Paste sequence', -command => \&Fix);
    $file->command (-label=>'Close sequence', -command => \&closesequence);
$configuration = $menubar->cascade(-label=>'Configuration', -tearoff=>0);
    $configuration->command (-label=>'Edit modification file', -command => \&modificationfile);
    $configuration->command (-label=>'Set extremity flag', -command => \&flag);
    $configuration->command (-label=>'Money matters', -command => \&money);
$find = $menubar->cascade(-label=>'Find', -tearoff=>0);
    $find->command (-label=>'Find start', -command => \&Start);
    $find->command (-label=>'Find stop', -command => \&Stop);
$tri = $menubar->cascade(-label=>'Triplets', -tearoff=>0);
    $tri->command (-label=>'Triplet, frame 0', -command => sub {triplet('0','1')});
    $tri->command (-label=>'Triplet, frame 1', -command => sub {triplet('1','1')});
    $tri->command (-label=>'Triplet, frame 2', -command => sub {triplet('2','1')});
    $tri->command (-label=>'No triplet', -command => sub {triplet('no')});
$tra = $menubar->cascade(-label=>'Translate', -tearoff=>0);
    $tra->radiobutton (-label=>'Frame 0', -variable=>\$selfframe, -value => 0);
    $tra->radiobutton (-label=>'Frame 1', -variable=>\$selfframe, -value => 1);
    $tra->radiobutton (-label=>'Frame 2', -variable=>\$selfframe, -value => 2);
    $tra->separator;
    $tra->command (-label=>'Translate-one letter code', -command => sub {translate('0',$selfframe)});

```

```

    $tra->command (-label=>'Translate-three letters code', -command => sub {translate('1',$selfframe)});

$textwindow = $main->Text(-height => 30, -width => 50)->place(-y=>5, -x =>5);
$textwindow->menu(undef);
$yscroll = $main->Scrollbar (-orient=>'vertical',
    -command => [yview=>$textwindow])->place(-y=>5, -x=>360, -height=>425);
$textwindow->configure (-yscrollcommand => [set => $yscroll]);

$main->Label(-text=>'Forward primer', -background=>'red')->place(-y=>10, -x=>380);
$diraccbutton = $main->Button(-text=>'Accept',
    -command=>\&AcceptDirect)->place(-y=>160, -x=>380);
$dirresbutton = $main->Button(-text=>'5\' modification',
    -command=>sub {Chooser('Direct')})->place(-y=>30, -x=>460);
$main->Checkbutton(-text=>'Flag', -variable=>\$dirflg)->place(-y=>30, -x=>542);
$dirdegbutton = $main->Button(-text=>'Degeneration',
    -command=>sub {degenerate($directprimer)})->place(-y=>30, -x=>610);
$main->Label(-text=>'Sequence :')->place(-y=>50, -x=>380);
$main->Label(-text=>'Lenght (bp):')->place(-y=>90, -x=>380);
$main->Label(-text=>'GC% :')->place(-y=>110, -x=>380);
$main->Label(-text=>'Tm :')->place(-y=>130, -x=>380);
$main->Label(-text=>'Name:')->place(-y=>90, -x=>500);
$dirname = $main->Entry(-width=>30)->place(-y=>110, -x=>500);

$main->Label(-text=>'Reverse primer', -background=>'yellow')->place(-y=>250, -x=>380);
$revaccbutton = $main->Button(-text=>'Accept',
    -command=>\&AcceptReverse)->place(-y=>400, -x=>380);
$revresbutton = $main->Button(-text=>'5\' modification',
    -command=>sub {Chooser('Reverse')})->place(-y=>270, -x=>460);
$main->Checkbutton(-text=>'Flag', -variable=>\$revflg)->place(-y=>270, -x=>542);
$revdegbutton = $main->Button(-text=>'Degeneration',
    -command=>sub {degenerate($revprimertempo)})->place(-y=>270, -x=>610);
$main->Label(-text=>'Sequence :')->place(-y=>290, -x=>380);
$main->Label(-text=>'Lenght (bp):')->place(-y=>330, -x=>380);
$main->Label(-text=>'GC% :')->place(-y=>350, -x=>380);
$main->Label(-text=>'Tm :')->place(-y=>370, -x=>380);
$main->Label(-text=>'Name:')->place(-y=>330, -x=>500);
$revname = $main->Entry(-width=>30)->place(-y=>350, -x=>500);

$dirset = $main->Button(-text=>'Set as Forward primer',
    -command=>\&Direct)->place(-y=>435, -x=>5);
$revset = $main->Button(-text=>'Set as Reverse primer',
    -command=>\&Reverse)->place(-y=>435, -x=>240);

```

```

$main->Button(-text=>'Generate Report',
             -command=>\&Report)->place(-y=>440, -x=>600);
$main->Button(-text=>'Clear Primers',
             -command=>\&Clear)->place(-y=>440, -x=>145);

MainLoop;

sub initialize {
    open (INIT, "< $inifile");
    do {$char = getc(INIT);
        $iniinfo .= $char;
    } until eof(INIT);
    close(INIT);
    if (!$iniinfo) {
        $main->Dialog(-title=>'Primer Wizard Error',
                    -text =>"Initialization file $inifile not found")->Show;
        return 'false';
    }
    @iniinformation = split (/\\n/, "$iniinfo");
    $basecost= $iniinformation[0];
    $moneytype = $iniinformation[1];
    $vat = $iniinformation[2];
    $flag = lc($iniinformation[3]);
    undef($iniinfo);
    &restrictionopen;
    &codonusageopen;
}

sub restrictionopen {
    undef ($restriction) if ($restriction);
    open (REST, "< modifications");
    do {$char = getc(REST);
        $restriction .= $char;
    } until ($char eq '-');
    close(REST);
    if (!$restriction) {
        $main->Dialog(-title=>'Primer Wizard Error',
                    -text =>'Modifications file missing!')->Show;
        return 'false';
    }
    chop ($restriction);
    &restrictioncreate;
}

```

```

sub restrictioncreate {
    if (@Enzyme) {
        undef(@Enzyme);
        undef(@Site);
    }
    @restlines = split (/\n/, "$restriction");
    $totlines = @restlines;
    $restnumber = 1;
    for ($i=0;$i<$totlines;$i++) {
        if ($restlines[$i] =~ /\*/) {
            $Enzyme[$restnumber] = $restlines[$i];
            $Site[$restnumber] = $restlines[$i+1];
            $restnumber++;
        }
    }
}

sub codonusageopen {
    undef ($AATri) if ($AATri);
    open (TRI, "< AA_triplet");
    do {$char =getc(TRI);
        $AATri .= $char;
    }
    until eof(TRI);
    close(TRI);
    if (!$AATri) {
        $main->Dialog(-title=>'Primer Wizard Error',
                    -text =>'Codon usage file missing!')->Show;
        return 'false';
    }
    @aatriplets = split (/\n/,$AATri);
}

sub money {
    $money = $main->Toplevel(-title=>'Money matters...');
    $money->geometry("290x80");
    $money->Label(-text=>'How much does a base cost?          cents')->place(-x=>5, -y=>5);
    $money->Entry(-width=>3, -textvariable=>\$basecost)->place(-x=>155, -y=>5);
    $money->Label(-text=>'Which money?')->place(-x=>5, -y=>33);
    $money->Radiobutton(-text=>"Euro", -variable=>\$moneytype, -value=>"Euro")->place(-x=>80, -y=>30);
}

```

```

$money->Radiobutton(-text=>"Dollar", -variable=>\$moneytype, -value=>"Dollar")->place(-x=>130, -y=>30);
$money->Label(-text=>'I.V.A. (V.A.T) %')->place(-x=>5, -y=>60);
$money->Entry(-width=>3, -textvariable=>\$vat)->place(-x=>80, -y=>60);
$money->Button(-text=>'Set', -command=>\&setmoney)->place(-x=>250, -y=>50);

sub setmoney {
    if (!$newini) {undef($newini);}
    $newini = $basecost."\n".$moneytype."\n".$vat."\n".$flag;
    unlink ($inifile);
    open (INIT, "> $inifile");
    print {INIT} "$newini";
    close (INIT);
    $money->destroy;
}

}

sub flag {
    $flagwin = $main->Toplevel(-title=>'Flag settings...');
    $flagwin->geometry("180x80");
    $flagwin->Label(-text=>'Set the restriction enzyme flag')->place(-x=>5, -y=>5);
    $flagwin->Entry(-width=>20, -textvariable=>\$flag)->place(-x=>15, -y=>25);
    $flagwin->Button(-text=>'Set', -command=>\&setflag)->place(-x=>140, -y=>50);

    sub setflag {
        if (!$newini) {undef($newini);}
        $flag = lc($flag);
        $newini = $basecost."\n".$moneytype."\n".$vat."\n".$flag;
        unlink ($inifile);
        open (INIT, "> $inifile");
        print {INIT} "$newini";
        close (INIT);
        $flagwin->destroy;
    }
}

sub modificationfile {
    $modifiedit = $main->Toplevel(-title=>'Edit modifications file');
    $modifiedittext = $modifiedit->Scrolled('Text', -width=>30, -height=>26, -scrollbars=>"e")->pack;
    $modifiedittext->menu(undef);
    $header = "Respect the format, please.\n";
    $modifiedittext->Contents($header,$restriction);
}

```

```

$modifiedit->Button(-text=>'Update', -command=>\&updatemodfile)->pack;
sub updatemodfile {
    $newmodifications = $modifiedittext->get('2.0', 'end');
    chomp ($newmodifications);
    chomp ($newmodifications);
    chomp ($newmodifications);
    chomp ($newmodifications);
    $newmodifications .= "\n---";
    unlink ($restrictionfile);
    open (INIT, "> $restrictionfile");
    print {INIT} "$newmodifications";
    close (INIT);
    undef($newmodifications);
    &restrictionopen;
    $modifiedit->destroy;
}
}

sub loadnasequence {
    if ($sequence) {undef($sequence);}
    my $type = [['DNA Sequence', ['.txt']]];
    $sequencefile = $main->getOpenFile(-filetypes =>$type);
    if(!$sequencefile){return 'false';}
    open (FILE, "< $sequencefile");
    do {$char =getc(FILE);
        if ($char =~ /[aAtTgGcCnN]/){
            $sequence .= uc($char);
        }
    } until eof(FILE);
    close(FILE);
    $textwindow->Contents($sequence);
}

sub closesequenece {
    $textwindow->Contents("");
    undef ($sequence);
}

sub Fix {
    undef($sequence) if ($sequence);
    $pastedseq = $textwindow->SelectionGet(-selection=>'CLIPBOARD');
}

```

```

@se = split (//, "$pastedseq");
$ind = 0;
while ($ind<=#se) {
    if ($se[$ind] =~ /[aAtTgGcCnN]{3}/){
        $sequence .= uc($se[$ind]);
    }
    $ind++;
}
$textwindow->Contents($sequence);
undef(@se);
undef($pastedseq);
}

sub triplet {
    if ($seqtripletted) {undef($seqtripletted);}
    if (!$sequence) {
        $main->Dialog(-title=>'Primer Wizard Error',
            -text =>'DNA sequence not found')->Show;
        return 'false';
    }
    @frame = @_ ;
    @sequencetemp = split (//,$sequence);
    if ($frame[0] eq 'no') {
        $textwindow->Contents($sequence);
        return 'false';
    }
    if ($frame[0] == 0) {
        if ($toadd) {undef($toadd);}
    }
    if ($frame[0] == 1) {
        $toadd = $sequencetemp[0];
        shift (@sequencetemp);
    }
    if ($frame[0] == 2) {
        $toadd = $sequencetemp[0].$sequencetemp[1];
        shift (@sequencetemp);
        shift (@sequencetemp);
    }
    for ($i=0;$i<=#sequencetemp;$i++) {
        if ($i%3 == 0 && $i != 0) {$seqtripletted .=" ";}
        $seqtripletted .= $sequencetemp[$i];
    }
}

```

```

    $seqtripletpadded = $toadd." ".$seqtripletpadded;
    $textwindow->Contents($seqtripletpadded) if ($frame[1]);
    return ($seqtripletpadded);
}

sub translate {
    undef($translation) if ($translation);
    @codetype = @_ ;
    $totransl = &triplet($codetype[1]);
    @readytotransl = split (/ /, "$totransl");
    for ($i=0; $i<=$#readytotransl; $i++) {
        if ($readytotransl[$i] =~ /[nN]/) {
            $translation .= '?' if ($codetype[0] == 0);
            $translation .= '???' if ($codetype[0] == 1);
        }
        else {
            for ($j=0; $j<=$#aatriplets; $j++) {
                @temp = split (/\.\/, "$aatriplets[$j]");
                if ($temp[2] eq $readytotransl[$i]) {
                    $translation .= $temp[$codetype[0]]." ";
                }
            }
        }
    }
    $textwindow->Contents($translation);
}

sub Direct {
    if ($warning) {undef($warning);}
    if ($directprimer) {
        undef($directprimer);
        undef($dirprimertemp);
        undef($dirprimerlength);
        $textwindow->tagDelete('dir');
        $dirlabel1->placeForget;
        $dirlabel2->placeForget;
        $dirlabel3->placeForget;
        $dirlabel4->placeForget;
    }
    if ($directprimermod) {
        undef($directprimermod);
    }
}

```

```

    undef($dirsiteseq);
    undef($direnzyme);
}
if ($dirprimermod) {
    undef($dirprimermod);
}
if ($anotherdir) {
    undef($anotherdir);
    $textwindow->tagDelete('dirfound');
}

($dirstart,$dirend) = $textwindow->tagRanges('sel');
$textwindow->tagAdd('dir', $dirstart,$dirend);
$textwindow->tagConfigure('dir', -background=>'red');
$directprimertemp = $textwindow->get($dirstart,$dirend);
$directprimer = uc($directprimertemp);
@directbasetemp = split (//,"$directprimer");
$anotherdir = $textwindow->search($directprimer,$dirend,'end');
    if ($anotherdir) {
        if ($anotherdir ne $dirstart) {
            $main->Dialog(-title=>'Primer Wizard Warning',
                -text =>'This primer will match more than one time in sequence (look at orange mark)')->Show;
            $textwindow->tagAdd('dirfound', $anotherdir);
            $textwindow->tagConfigure('dirfound', -background=>'orange');
        }
    }
}
for ($i=0;$i<=#directbasetemp;$i++) {
    if ($directbasetemp[$i] ne " ") {
        $dirprimertemp .= $directbasetemp[$i];
    }
}
$directprimer = $dirprimertemp;
@directbase = split (//,"$directprimer");
for ($i=0;$i<=#directbase;$i++) {
    if ($directbase[$i] !~ /[AaTtCcGgNn]/) {
        $warning = "yes";
    }
}
}
if ($warning) {
    $main->Dialog(-title=>'Primer Wizard Error',
        -text =>'Please select the primer from DNA sequence...')->Show;
    return 'false';
}

```

```

}
else {
for ($i=0;$i<=#directbase;$i++) {
    if($directbase[$i] ne " ") {
        $dirprimerlenght++;
    }
}
$GC=0;
for ($i=0;$i<=#directbase;$i++) {
    if ($directbase[$i] eq "C" || $directbase[$i] eq "G") {
        $GC++;
    }
}
$dirGCpercent = substr((( $GC/$dirprimerlenght)*100),0,5);
$directTm = 69.3+(0.41*$dirGCpercent)-(650/$dirprimerlenght);
$dirTm = substr($directTm,0,5);
$dirlabel1 = $main->Label(-text=>"$directprimer")->place(-y=>70, -x=>380);
$dirlabel2 = $main->Label(-text=>"$dirprimerlenght")->place(-y=>90, -x=>450);
$dirlabel3 = $main->Label(-text=>"$dirGCpercent")->place(-y=>110, -x=>420);
$dirlabel4 = $main->Label(-text=>"$dirTm")->place(-y=>130, -x=>420);
}
}

sub Reverse {
if ($warning) {undef($warning);}
if ($reverseprimer) {
    undef($reverseprimer);
    undef($revprimerlenght);
    undef($revprimertempo);
    $textwindow->tagDelete('rev');
    $revlabel1->placeForget;
    $revlabel2->placeForget;
    $revlabel3->placeForget;
    $revlabel4->placeForget;
}
if ($reverseprimermod) {
    undef($reverseprimermod);
}
if ($revprimermod) {
    undef($revprimermod);
}
if ($anotherrev) {

```

```

    undef($anotherrev);
    $textwindow->tagDelete('revfound');
}
($revstart,$revend) = $textwindow->tagRanges('sel');
$textwindow->tagAdd('rev', $revstart,$revend);
$textwindow->tagConfigure('rev', -background=>'yellow');
$revprimertemp = $textwindow->get($revstart,$revend);
@reversebasetemp = split (//,"$revprimertemp");
$anotherrev = $textwindow->search($revprimertemp,0.1,$revend);
    if ($anotherrev) {
        if ($anotherrev ne $revstart) {
            $main->Dialog(-title=>'Primer Wizard Warning',
                -text =>'This primer will match more than one time in sequence (look at orange mark)')->Show;
            $textwindow->tagAdd('revfound', $anotherrev);
            $textwindow->tagConfigure('revfound', -background=>'orange');
        }
    }
}

for ($i=0;$i<=#reversebasetemp;$i++) {
    if ($reversebasetemp[$i] ne " ") {
        $revprimertempo .= $reversebasetemp[$i];
    }
}
@reversebase = split (//,"$revprimertempo");
for ($i=0;$i<=#reversebase;$i++) {
    if ($reversebase[$i] !~ /[AaTtCcGgNn]/) {
        $warning = "yes";
    }
}
if ($warning) {
    $main->Dialog(-title=>'Primer Wizard Error',
        -text =>'Please select the primer from DNA sequence...')->Show;
}
else {
for ($i=0;$i<=#reversebase;$i++) {
    if($reversebase[$i] ne " ") {
        $revprimerlenght++;
    }
}
}
$GC=0;
for ($i=0;$i<=#reversebase;$i++) {
    if (uc($reversebase[$i]) eq "C" || uc($reversebase[$i]) eq "G") {

```

```

        $GC++;
    }
    if (uc($reversebase[$i]) eq "C") {
        $compbase[$i] = "G";
    }
    if (uc($reversebase[$i]) eq "G") {
        $compbase[$i] = "C";
    }
    if (uc($reversebase[$i]) eq "A") {
        $compbase[$i] = "T";
    }
    if (uc($reversebase[$i]) eq "T") {
        $compbase[$i] = "A";
    }
}
for ($i=$#reversebase;$i>=0;$i--) {
    $reverseprimer .= $compbase[$i];
}
$revGCpercent = substr(($GC/$revprimerlenght)*100),0,5);
$reverseTm = 69.3+(0.41*$revGCpercent)-(650/$revprimerlenght);
$revTm = substr($reverseTm,0,5);
$revlabel1 = $main->Label(-text=>"$reverseprimer")->place(-y=>310, -x=>380);
$revlabel2 = $main->Label(-text=>"$revprimerlenght")->place(-y=>330, -x=>450);
$revlabel3 = $main->Label(-text=>"$revGCpercent")->place(-y=>350, -x=>420);
$revlabel4 = $main->Label(-text=>"$revTm")->place(-y=>370, -x=>420);
}
}

sub Chooser {
    if (!$directprimer && !$reverseprimer) {
        $main->Dialog(-title=>'Primer Wizard Error',
            -text =>'Select a primer,first...')->Show;
    }
    return 'false';
}
@inuse = @_ ;
$chooser = $main->Toplevel(-title=>'Cloning tail chooser');
$restchoose = $chooser->BrowseEntry (-label =>"Choose 5\' modification",
    -width=>20,
    -variable=> \ $enzyme,
    -browsecmd=> \&fill)->pack;

$counter = 1;
while ($counter< $restnumber) {

```

```

    $restchoose->insert($counter-1,$Enzyme[$counter]);
    $counter++;
}
sub fill {
    if ($foundrest) {
        $textwindow->tagDelete('ERsite');
    }
    if ($enzyme eq "\*User Defined\*") {
        &editmodif;
    }
    else {
        for ($check=1;$check<$restnumber;$check++) {
            if ($Enzyme[$check] eq $enzyme) {
                $siteseq = $Site[$check];
                $enzy = $Enzyme[$check];
                &modifind;
            }
        }
    }
}
}
}

sub editmodif {
    $modifwin = $main->Toplevel(-title=>'Edit 5\' modification');
    $modifwin->geometry("270x60");
    $modifwin->Label(-text=>'Name')->place(-x=>5, -y=>5);
    $modifwin->Entry(-width=>20, -textvariable=>\$modifnam)->place(-x=>80, -y=>5);
    $modifwin->Label(-text=>'Sequence')->place(-x=>5, -y=>30);
    $modifwin->Entry(-width=>20, -textvariable=>\$modifseq)->place(-x=>80, -y=>30);
    $modifwin->Button(-text=>'Accept',
        -command=>\&acce)->place(-x=>215, -y=>12);
    sub acce {
        @modsplit = split (//,$modifseq);
        foreach $val (@modsplit){
            print "$val\n";
            if ($val !~ /[AaTtCcGgNn]/) {
                $main->Dialog(-title=>'Primer Wizard Error',
                    -text =>'The modification contains non-DNA characters')->Show;
                return 'false';
            }
        }
    }
    $enzy = $modifnam;
}

```

```

        $siteseq = $modifseq;
        $modifwin->destroy;
        &modifind;
    }
}

sub modifind {
    $foundrest = $textwindow->search(uc($siteseq),0.1,'end');
    if ($foundrest) {
        @sequelements = split (//, "$siteseq");
        $seqlen = @sequelements;
        @tagstart = split (/\.\/, "$foundrest");
        $tagending = $tagstart[1]+$seqlen;
        $tagend = $tagstart[0].".$tagending;
        $textwindow->tagAdd('ERsite',$foundrest,$tagend);
        $textwindow->tagConfigure('ERsite', -background=>'grey');
        $main->Dialog(-title=>'Primer Wizard Warning',
                    -text =>'Warning:the chosen 5\' modification is present in the sequence (marked in grey)')->Show;
    }
    &gotorestriction($inuse[0]);
}

sub gotorestriction {
    @fromwho = @_ ;
    if ($fromwho[0] eq "Direct") {
        &RestrictionDirect;
    }
    if ($fromwho[0] eq "Reverse"){
        &RestrictionReverse;
    }
}

sub RestrictionDirect {
    $chooser->destroy;
    undef ($flagdir) if ($flagdir && !$dirflg);
    $flagdir = $flag if ($dirflg);
    if ($zeny eq "*None (Remove)*") {
        if ($dirprimermod) {
            $directprimermod = $dirprimermod;
        }
        else {
            $directprimermod = $directprimer;
        }
    }
}

```

```

    }
    undef($dirsiteseq);
    undef($direnzyme);
}
else {
    $dirsiteseq = uc($siteseq);
    if ($dirprimermod) {
        $directprimermod = $flagdir.$dirsiteseq.$dirprimermod;
    }
    else {
        $directprimermod = $flagdir.$dirsiteseq.$directprimer;
    }
    $dirflag = $flagdir if ($dirflg);
    $dirflag = 'none' if (!$dirflg);
    $dirsiteseq = uc($siteseq);
    $direnzyme = $enzy;
}
$dirlabell->placeForget;
$dirlabell = $main->Label(-text=>"$directprimermod")->place(-y=>70, -x=>380);
}

```

```

sub RestrictionReverse {
    $chooser->destroy;
    undef ($flagrev) if ($flagrev && !$revflg);
    $flagrev = $flag if ($revflg);
    if ($enzy eq "*None (Remove)*") {
        if ($revprimermod) {
            $reverseprimermod = $revprimermod;
        }
        else {
            $reverseprimermod = $reverseprimer;
        }
        undef($revsiteseq);
        undef($revenzyme);
    }
    else {
        $revsiteseq = uc($siteseq);
        if ($revprimermod) {
            $reverseprimermod = $flagrev.$revsiteseq.$revprimermod;
        }
        else {
            $reverseprimermod = $flagrev.$revsiteseq.$reverseprimer;
        }
    }
}

```

```

    }
    $revflag = $flagrev if ($revflg);
    $revflag = 'none' if (!$revflg);
    $revenzyme = $enzy;
    $revsiteseq = uc($siteseq);
}
$revlabell1->placeForget;
$revlabell1 = $main->Label(-text=>"$reverseprimermod")->place(-y=>310, -x=>380);
}

sub Clear {
if ($warning) {undef($warning);}
if ($directprimer) {
    undef($directprimer);
    undef($dirprimertemp);
    undef($dirprimerlenght);
    $textwindow->tagDelete('dir');
    $dirlabell1->placeForget;
    $dirlabel2->placeForget;
    $dirlabel3->placeForget;
    $dirlabel4->placeForget;
}
if ($directprimermod) {
    undef($directprimermod);
    undef($dirsiteseq);
    undef($direnzyme);
}
if ($dirprimermod) {
    undef($dirprimermod);
}
if ($anotherdir) {
    undef($anotherdir);
    $textwindow->tagDelete('dirfound');
}
if ($reverseprimer) {
    undef($reverseprimer);
    undef($revprimerlenght);
    undef($revprimertempo);
    $textwindow->tagDelete('rev');
    $revlabell1->placeForget;
    $revlabel2->placeForget;
    $revlabel3->placeForget;
}

```

```

        $revlabel4->placeForget;
    }
    if ($reverseprimermod) {
        undef($reverseprimermod);
    }
    if ($revprimermod) {
        undef($revprimermod);
    }
    if ($anotherrev) {
        undef($anotherrev);
        $textwindow->tagDelete('revfound');
    }
}

sub AcceptDirect {
    if (!$directprimer) {return 'false'}
    $textwindow->Contents($sequence);
    &dirmark;
    if ($revaccept) {
        &revmark;
    }
    $dirresbutton->configure(-state=>'disabled');
    $dirdegbutton->configure(-state=>'disabled');
    $dirset->configure(-state=>'disabled');
    $diraccbutton->configure(-state=>'disabled');
    $directname = $dirname->get;
    $diraccept = $main->Label(-text=>"$directname accepted")->place(-x=>500, -y=>130);
    $dirreject = $main->Button(-text=>"Reject",
        -command=>sub {reject($h)})->place(-x=>450, -y=>160);
}

sub dirmark {
    $wheredirstart = $textwindow->search("$directprimer",0.1,'end');
    @temp = split (/\.\/, "$wheredirstart");
    $directend = $temp[1]+$dirprimerlenght;
    $wheredirend = $temp[0].".$directend;
    $textwindow->tagAdd('Direct',$wheredirstart,$wheredirend);
    $textwindow->tagConfigure('Direct', -background=>'red');
}

sub revmark {
    $whererevstart = $textwindow->search("$revprimertempo",0.1,'end');

```

```

    @temp = split (/\/\./, "$whererevstart");
    $reverseend = $temp[1]+$revprimerlenght;
    $whererevend = $temp[0].".".$reverseend;
    $textwindow->tagAdd('Reverse', $whererevstart, $whererevend);
    $textwindow->tagConfigure('Reverse', -background=>'yellow');
}

```

```

sub reject {
    @fromwho = @_ ;
    if ($fromwho[0] eq "direct") {
        $diraccept->placeForget;
        undef($diraccept);
        $dirreject->placeForget;
        $textwindow->tagDelete('Direct');
        $dirresbutton->configure(-state=>'normal');
        $dirdegbutton->configure(-state=>'normal');
        $dirset->configure(-state=>'normal');
        $diraccbutton->configure(-state=>'normal');
    }
    if ($fromwho[0] eq "reverse") {
        $revaccept->placeForget;
        undef($revaccept);
        $revreject->placeForget;
        $textwindow->tagDelete('Reverse');
        $revresbutton->configure(-state=>'normal');
        $revdegbutton->configure(-state=>'normal');
        $revset->configure(-state=>'normal');
        $revaccbutton->configure(-state=>'normal');
    }
}

```

```

sub AcceptReverse {
    if (!$reverseprimer) {return 'false'}
    $textwindow->Contents($sequence);
    &revmark;
    if ($diraccept) {
        &dirmark;
    }
    $revresbutton->configure(-state=>'disabled');
}

```

```

$revdegbutton->configure(-state=>'disabled');
$revset->configure(-state=>'disabled');
$revaccbutton->configure(-state=>'disabled');
$reversename = $revname->get;
$revaccept = $main->Label(-text=>"$reversename accepted")->place(-x=>500, -y=>370);
$revreject = $main->Button(-text=>"Reject",
                          -command=>sub {reject($k)})->place(-x=>450, -y=>400);
}

sub Report {
  if (!$diraccept || !$revaccept) {
    $main->Dialog(-title=>'Primer Wizard Error',
                -text =>'You must accept primers, first...')->Show;
    return 'false';
  }
  $header = "Primers generated by   =\\=PrimerWizard=/=   written by Matram\\n\\n";
  @tempstart = split (/\.\/, "$dirstart");
  @tempend = split (/\.\/, "$revend");
  $amplification = $tempend[1]-$tempstart[1];
  if ($dirdegen) {
    $finaldirect4 = "Degeneration: yes\\n";
    $directprimer = $dirprimermod;
  }
  else {$finaldirect4 = "Degeneration: no\\n"}

  $finaldirect1 = "Direct Primer\\n\\nName : $directname\\nSequence: $directprimer\\n";
  $finaldirect2 = "Lenght: $dirprimerlenght\\nGC%: $dirGCpercent   Tm: $dirTm\\n";
  print "dirsiteseq = $dirsiteseq\\n";
  if ($dirsiteseq) {
    if ($dirflag eq 'none') {
      $finaldirect3 = "5\' modification: $direnzyme\\n";
      $dirdefinitive = $dirsiteseq.$directprimer;
    }
    else {
      $finaldirect3 = "5\' modification: $direnzyme with flag $dirflag\\n";
      $dirdefinitive = uc($dirflag).$dirsiteseq.$directprimer;
    }
  }
  else {

```

```

    $finaldirect3 = "5\' modification: none\n";
    $dirdefinitive = $directprimer;
}
@dirtotbase = split (//,"$dirdefinitive");
$dirnumber = @dirtotbase;
$dircost = ($basecost*($dirnumber))/100;
$dirwithvat = $dircost + ($dircost * $vat / 100);
$finaldirect5 = "Complete sequence: $dirdefinitive\nFull lenght: $dirnumber\nThis primer will cost $dircost $moneytype tax excluded, $dirwithvat $moneytype tax included\n";
$finaldirect = $finaldirect1.$finaldirect2.$finaldirect3.$finaldirect4.$finaldirect5;

if ($revdegen) {
    $finalreverse4 = "Degeneration: yes\n";
    $reverseprimer = $revprimermod;
}
else {$finalreverse4 = "Degeneration: no\n"}
$finalreverse1 = "Reverse Primer\n\nName : $reversename\nSequence: $reverseprimer\n";
$finalreverse2 = "Lenght: $revprimerlenght\nGC%: $revGCpercent      Tm: $revTm\n";
print "revsitieseq = $revsitieseq\n";
if ($revsitieseq) {
    if ($revflag eq 'none') {
        $finalreverse3 = "5\' modification: $revenzyme\n";
        $revdefinitive = $revsitieseq.$reverseprimer;
    }
    else {
        $finalreverse3 = "5\' modification: $revenzyme with flag $revflag\n";
        $revdefinitive = uc($revflag).$revsitieseq.$reverseprimer;
    }
}
else {
    $finalreverse3 = "5\' modification: none\n";
    $revdefinitive = $reverseprimer;
}
@revtotbase = split (//,"$revdefinitive");
$revnumber = @revtotbase;
$revcost = ($basecost*($revnumber))/100;
$revwithvat = $revcost + ($revcost * $vat / 100);
$finalreverse5 = "Complete sequence: $revdefinitive\nFull lenght: $revnumber\nThis primer will cost $revcost $moneytype tax excluded, $revwithvat $moneytype tax included\n";
$finalreverse6 = "\n\nThis primer couple will amplify a $amplification bp long region";
$finalreverse = $finalreverse1.$finalreverse2.$finalreverse3.$finalreverse4.$finalreverse5.$finalreverse6;

$reportcomplete = $header.$finaldirect."\n\n".$finalreverse;
$reportwindow = $main->Toplevel(-title => 'Primer Wizard Report');

```

```

$reporttext = $reportwindow->Text(-width=>80, -height=>30)->pack;
$reporttext->Contents($reportcomplete);
$reportwindow->Button (-text=>'Save',
                    -command=>\&save)->pack;

sub save {
    $savefile = $reportwindow->getSaveFile;
    if(!$savefile){return 'false';}
    open (SAVEFILE, "> $savefile");
    print {SAVEFILE} "$reportcomplete";
    close (SAVEFILE);
}

sub degenerate {
    @who = @_ ;
    if($degeneration) {
        undef($degeneration);
        undef($degeneration2view);
    }
    open (TABLE, "< degen_table");
    do {$char =getc(TABLE);
        $degeneration .= $char;
        if ($char eq "*") {$char = " ";}
        $degeneration2view .= $char;
    } until eof (TABLE);
    close(TABLE);
    if ($who[0] eq $directprimer) {
        $lenghttocompare = $#directbase;
        if (!$dirprimermod) {
            $stovisual = $who[0];
        }
        else {
            $stovisual = $dirprimermod;
        }
    }
    else {
        $lenghttocompare = $#reversebase;
        if (!$revprimermod) {
            $stovisual = $who[0];
        }
        else {
            $stovisual = $revprimermod;
        }
    }
}

```

```

    }
}
&viewtable;
$degewindow = $main->Toplevel(-title=>'Degeneration');
$degewindow->geometry("300x160");
$degewindow->Label(-text=>'Modify primer according to table')->place(-x=>5, -y=>10);
$degewindow->Label(-text=>"Old: $who[0]")->place(-x=>5, -y=>40);
$degewindow->Label(-text=>'Translation:')->place(-x=>5, -y=>60);
$degewindow->Label(-text=>'Changes test:')->place(-x=>5, -y=>80);
$degewindow->Label(-text=>'New:')->place(-x=>5, -y=>105);
$degeentry = $degewindow->Entry(-width=>40)->place(-x=>40, -y=>105);
$degewindow->Button(-text=>'Done',
    -command=>\&Degeneration)->place(-x=>230, -y=>130);
$degewindow->Button(-text=>'Auto degenerate',
    -command=>\&Auto)->place(-x=>5, -y=>130);
$degewindow->Button(-text=>'Test changes',
    -command=>\&Testing)->place(-x=>100, -y=>130);

$degeentry->insert(0,$tovisual);
@usetable = split (/\\n/, "$degeneration");

@result = test();
$degewindow->Label(-text=>"$result[0]")->place(-x=>80, -y=>60);

sub test {
if($trad) {
    undef($trad);
    undef($autotrad);
    undef($ending);
    undef(@primthree);
}
if($changtest) {$changtest->placeForget;}
$totest = $degeentry->get;
@analizprimer = split (//, "$totest");
$m = 0;
$q = 1;
for ($i=0;$i<=#analizprimer;$i++) {
    if ($q<4) {
        $primthree[$m].= $analizprimer[$i];
        $q++;
    }
    else {

```

```

        $m++;
        $i--;
        $q=1;
    }
}
@lasttripl = split (//, "$primthree[$m]");
if($#lasttripl != 2) {
    $ending = $primthree[$m];
    $m--;
}
$count = 0;
for($i=0;$i<=$m;$i++) {
    for($j=2;$j<=$#usetable;$j++) {
        @temp = split (/ /, "$usetable[$j]");
        if ($temp[3] =~ $primthree[$i] || $temp[5] =~ $primthree[$i] || $temp[6] =~ $primthree[$i]) {
            $trad.=$temp[1]." ";
            $autotrad.=$temp[5];
            $count++;
        }
    }
}
$count--;
if($ending) {
    $autotrad = $autotrad.$ending;
}

if ($count != $m) {
    $main->Dialog(-title=>'Primer Wizard Error',
                -text =>"Revise degenerations...use the table")->Show;
    return 'false';
}
return ($trad,$autotrad)
}

sub viewtable {
    $degenwin = $main->Toplevel(-title=>'Degeneration table');
    $degenwintext = $degenwin->Text(-width=>40, -height=>27)->pack;
    $degenwintext->Contents($degeneration2view);
}
sub Auto {
    $degeentry->delete(0, 'end');
    $degeentry->insert(0, $result[1]);
}

```

```

}
sub Testing {
    @testresult = test();
    $changtest = $degewindow->Label(-text=>"$testresult[0]")->place(-x=>80, -y=>80);
}
sub Degeneration {
    $modprimer = $degeentry->get;
    @temp = split (//,$modprimer);
    if ($#temp ne $lenghttocompare) {
        $main->Dialog(-title=>'Primer Wizard Warning',
                    -text =>'Warning: primer lenght has been modified, please revise degeneration')->Show;
        return 'false';
    }
    for ($i=0;$i<=$#temp;$i++) {
        if ($temp[$i] !~ /[ AaTtCcGgYyNnIiDdCcHhRr]/) {
            $main->Dialog(-title=>'Primer Wizard Error',
                        -text =>"Warning: unexpected base ($temp[$i]) in primer, please revise degeneration")->Show;
            return 'false';
        }
    }
    if ($who[0] ne $modprimer) {
        if ($who[0] eq $directprimer) {
            $dirprimermod = $modprimer;
            $directprimermodif = $dirprimermod;
            if ($dirsiteseq) {
                $directprimermodif = $flag.$dirsiteseq.$dirprimermod;
            }
            $dirlabel1->placeForget;
            $dirlabel3->placeForget;
            $dirlabel4->placeForget;
            $dirlabel1 = $main->Label(-text=>"$directprimermodif")->place(-y=>70, -x=>380);
            $dirlabel3 = $main->Label(-text=>"about $dirGCpercent")->place(-y=>110, -x=>420);
            $dirlabel4 = $main->Label(-text=>"about $dirTm")->place(-y=>130, -x=>420);
            $dirdegen = "yes";
        }
        else {
            if($moddegprimer) {undef($moddegprimer);}
            @tempmod = split (//, "$modprimer");
            for ($i=0;$i<=$#tempmod;$i++) {
                if (uc($tempmod[$i]) eq "C") {
                    $degrevbase[$i] = "G";
                }
            }
        }
    }
}

```

```

    if (uc($tempmod[$i]) eq "G") {
        $degrevbase[$i] = "C";
    }
    if (uc($tempmod[$i]) eq "A") {
        $degrevbase[$i] = "T";
    }
    if (uc($tempmod[$i]) eq "T") {
        $degrevbase[$i] = "A";
    }
    if (uc($tempmod[$i]) eq "D") {
        $degrevbase[$i] = "H";
    }
    if (uc($tempmod[$i]) eq "Y") {
        $degrevbase[$i] = "R";
    }
    if (uc($tempmod[$i]) eq "H") {
        $degrevbase[$i] = "D";
    }
    if (uc($tempmod[$i]) eq "N") {
        $degrevbase[$i] = "N";
    }
    if (uc($tempmod[$i]) eq "R") {
        $degrevbase[$i] = "Y";
    }
}
for ($i=$#tempmod;$i>=0;$i--) {
    $moddegprimer .= $degrevbase[$i];
}
$modprimer = $moddegprimer;
$revprimermod = $modprimer;
$reverseprimermodif = $revprimermod;
if ($revsiteseq) {
    $reverseprimermodif = $flag.$revsiteseq.$revprimermod;
}
$revlabel1->placeForget;
$revlabel3->placeForget;
$revlabel4->placeForget;
$revlabel1 = $main->Label(-text=>"$reverseprimermodif")->place(-y=>310, -x=>380);
$revlabel3 = $main->Label(-text=>"about $revGCpercent")->place(-y=>350, -x=>420);
$revlabel4 = $main->Label(-text=>"about $revTm")->place(-y=>370, -x=>420);
$revdegen = "yes";
}

```

```

    }
    $degewindow->destroy;
    $degenwin->destroy;
}

sub Start {
    $startfrom = "1.0";
    $newindex[1]=0;
    &finderstart("ATG");
    &finderstart("atg");
    &finderstart("Met");
    &finderstart("M");
    sub finderstart {
        @what = @_ ;
        while (!$foundat) {
            $foundat = $textwindow->search($what[0], $startfrom, 'end');
            if ($foundat) {
                @newindex = split (/\./, "$foundat");
                if ($what[0] eq "M") {
                    $nextchar = $newindex[1]+1;
                }
                else {
                    $nextchar = $newindex[1]+3;
                }
                $startnew = $newindex[0].".$nextchar;
                $startfrom = $startnew;
                $textwindow->tagAdd('ATG', $foundat, $startnew);
                undef ($foundat);
            }
            else {return 'false';}
        }
    }

    $textwindow->tagConfigure('ATG', -background=>'green');
}

sub Stop {
    $endtofind[0] = "TAA";
    $endtofind[1] = "taa";
}

```

```

$endtofind[2] = "TGA";
$endtofind[3] = "tga";
$endtofind[4] = "TAG";
$endtofind[5] = "tag";
$endtofind[6] = "End";
$endtofind[7] = "*";
$l = 0;
while ($l<8) {
$startfrom = "1.0";
&finderend;
sub finderend {
    while (!$foundat) {
        $foundat = $textwindow->search("$endtofind[$l]", $startfrom, 'end');
        if ($foundat) {
            @newindex = split (/\./, "$foundat");
            if($endtofind[$l] eq "*") {
                $nextchar = $newindex[1]+1;
            }
            else {
                $nextchar = $newindex[1]+3;
            }
            $startnew = $newindex[0].".$nextchar;
            $startfrom = $startnew;
            $textwindow->tagAdd('END', $foundat, $startnew);
            undef ($foundat);
        }
        else {return 'false';}
    }
}
$l++;
}
$textwindow->tagConfigure('END', -background=>'lightblue');
}

```