Supplemental Text1.

Following contains 5 scripts and parameter tables for

<VCF_to_HRMprimer_step1.pl>, <TD_table.txt>, <VCF_to_HRMprimer_step2_exp1.pl>, <VCF_to_HRMprimer_noNNNs.pl>, < VCF_to_HRMprimer_NNNsHRM.pl> and <VCF_to_HRM_blast.pl>

Search command will help you to find the each scripts with their names. Relationship for each scripts is summarized in the flow chart in Supplemental figure 1.

```
#VCF_to_HRMprimer_step2_exp1.pl
#
###############################################################################################################
# Insall of Bioperl is needed.
# How to use.
# perl VCF_to_HRMprimer_step1.pl path_to_sequence_data input_vcf_file > output_file
# perl VCF_to_HRMprimer_step1.pl C:¥Glyma1¥Glyma1.fa 170912TF_HC_SNP.vcf > 180316conf.txt
##################################################################################################################


use Bio::DB::Fasta;


my $lengh = 200;
my $half_lengh = $lengh * 0.5;

my $fastafile = $ARGV[0];#
my $varientfile = $ARGV[1];#


my $db = Bio::DB::Fasta->new($fastafile);

open(my $fh, "<", $varientfile)
  or die "Cannot open $varientfile: $!";


while(my $line = readline $fh)
{
    chomp $line;
if($line=~/[A-Z]+,[A-Z]+/){next}
    my $readline = $line;
    my @readline = split(/¥s+/, $readline);

    my $CHROM = $readline[0];
    my $POS = $readline[1];
    my $ID = $readline[2];
    my $REF = $readline[3];
    my $ALT = $readline[4];
    my $QUAL = $readline[5];
    my $FILTER = $readline[6];
    my $INFO = $readline[7];
    my $FORMAT = $readline[8];
    my $BAM = $readline[9];
        #my $ID = $readline[10];

    my $varient;
    if($line=~/^#/){
    $INDEX = $BAM;
    next}
        $val = length("$REF")-length("$ALT");
    if($val>0){$varient = "DEL"}     #DELETION
    if($val==0){$varient = "SNP"}   #SNP
    if($val<0){$varient = "INS"}        #INSERT


    my $start = $POS - $half_lengh;
    my $end = $POS + $half_lengh;
    my $seq = $db->seq($CHROM,$start => $end);# (ID, first nucleotide＝＞end nucleotide)
    my $alt_seq = "not RFLP";


            my $startenz = $POS - 5;
    my $endenz = $POS + 5;

    my $seqenz = $db->seq($CHROM,$startenz => $endenz);# (ID, first nucleotide＝＞end nucleotide)
```

```
    my $seq1enz = substr($seqenz, 0 , 5 );
    my $seq2enz = substr($seqenz, 5 + 1);
    my $alt_seqenz = "$seq1enz$ALT$seq2enz";
    my $ref_seqenz = "$seq1enz$REF$seq2enz";


    if($varient=~/SNP/)
    {


}
    print "$CHROM¥t$POS¥t$seq¥t$varient¥t$REF¥t$ALT¥t$INDEX¥t$alt_seq¥t$ref_seqenz¥t$alt_seqenz¥n";

}

#end of VCF_to_HRMprimer_step1.pl

################################################
#TD_table.txt

AA  -1.2    -1  -7.9    -22.2
TT  -1.2    -1  -7.9    -22.2
AT  -0.9    -0.88   -7.2    -20.4
TA  -0.9    -0.58   -7.2    -21.3
CA  -1.7    -1.45   -8.5    -22.7
TG  -1.7    -1.45   -8.5    -22.7
GT  -1.5    -1.44   -8.4    -22.4
AC  -1.5    -1.44   -8.4    -22.4
CT  -1.5    -1.28   -7.8    -21
AG  -1.5    -1.28   -7.8    -21
GA  -1.5    -1.3    -8.2    -22.2
TC  -1.5    -1.3    -8.2    -22.2
CG  -2.8    -2.17   -10.6   -27.2
GC  -2.3    -2.24   -9.8    -24.4
GG  -2.1    -1.84   -8  -19.9
CC  -2.1    -1.84   -8  -19.9
C   1.7 0.98    0.1 -2.8
G   1.7 0.98    0.1 -2.8
A   1.7 1.03    2.3 4.1
T   1.7 1.03    2.3 4.1

#end of TD_table.txt


####################

#VCF_to_HRMprimer_step2_exp1.pl

########################################################################################################################
##
##  PATH to primer3_core is nesessary.
##
##  perl VCF_to_HRMprimer_step2_exp1.pl 170628out1.txt  > 170926AF_HRMout_12tr_Final.txt
########################################################################################################################

######################################
%SUGI=();
%UNIF=();

$file1="TD_table.txt";

open FILE1, $file1 or die "not found your $file1 containing the data for thermodyamics of nuclotides.¥n";

while($line=<FILE1>){
    chomp $line;
    ($nuc,$sugi,$unif)=split(/¥t/,$line);
    $SUGI{$nuc}=$sugi;
    $UNIF{$nuc}=$unif;
}

close(FILE1);
####################################
```

```
$listfile=shift @ARGV;
open(FILE,$listfile) or die "not found your file.¥n";

while($line=<FILE>){
    chomp $line;
    if($line=~/[ATGCNX],[ATGCNX]/){next}
    ($a,$x,$b,$c,$d,$e,$f,$g,$h,$i,$j,$k,$l,$m,$n,$o,$dcap_enz_seq,$dcap_enz_name,$dcap)=split(/¥t/,$line);
    push @name, $a;
    push @position, $x;
    push @seq,$b;
    push @motif, $c;
    push @ref, $d;
    push @alt, $e;
    }


$number=@name;#
$seq_num=0;#
$range =("50-100");#
$target=101;#

print "CHR","¥t","POS","¥t","SEQ","¥t","REF","¥t","ALT","¥t","TYPE","¥t","PRODUCT_SIZE","¥t","FW_NAME","¥t","FW_SEQ","
¥t","RV_NAME","¥t","RV_SEQ","¥t","DELTA_G_REF_PRODUCT","¥t","DELTA_G_ALT_PRODUCT","¥n";

for($i;$i<$number;++$i){#

    $seq_num++;#

    $val = length($ref[$i]) - length($alt[$i]);#
    $length = length($seq[i]);#
    $subs=$ref[$i].$alt[$i];#

    if($val==0 && $subs=~m/AC|AG|TG|TC|GA|GT|CT|CA|GC|CG|AT|TA/i){#Transversion型の


        $seq[$i]=~s/¥x0D¥x0A$|¥x0D$|¥x0A$//;#

        $file=sprintf "test.tmp";#
        $fileout=sprintf "$name[$i]_$position[$i]_out.tmp";#

        open FILE1,">$file";#

        print FILE1 "SEQUENCE_ID¥=$name[$i]_$position[$i]¥n";
        print FILE1 "SEQUENCE_TEMPLATE¥=$seq[$i]¥n";
        print FILE1 "SEQUENCE_TARGET=","$target",",1","¥n";

        print FILE1 "PRIMER_TASK=generic¥n";
        print FILE1 "PRIMER_PICK_LEFT_PRIMER=1¥n";
        print FILE1 "PRIMER_PICK_INTERNAL_OLIGO=0¥n";
        print FILE1 "PRIMER_PICK_RIGHT_PRIMER=1¥n";
        print FILE1 "PRIMER_OPT_SIZE=25¥n";
        print FILE1 "PRIMER_MIN_SIZE=18¥n";
        print FILE1 "PRIMER_MAX_SIZE=30¥n";
        print FILE1 "PRIMER_PRODUCT_SIZE_RANGE=$range¥n";
        print FILE1 "PRIMER_NUM_NS_ACCEPTED=1¥n";
        print FILE1 "PRIMER_NUM_RETURN=1¥n";
        print FILE1 "PRIMER_FILE_FLAG=1¥n";
        print FILE1 "PRIMER_EXPLAIN_FLAG=1¥n";
        print FILE1 "PRIMER_MAX_POLY_X=3¥n";
        print FILE1 "¥=¥n";
        close(FILE1);

        system "primer3_core <$file> $fileout";

        $dcount=0;

        open FILE2,$fileout;

        while($line=<FILE2>){

        $line=~s/¥x0D¥x0A$|¥x0D$|¥x0A$//;#改行コードの削除
```

```
        if($line=~m/PRIMER_LEFT_¥d*_SEQUENCE¥=/){$fw=$';++$dcount;}
        if($line=~m/PRIMER_RIGHT_¥d*_SEQUENCE¥=/){$rv=$';++$dcount;}
        if($line=~m/PRIMER_LEFT_¥d*=/){($m,$n)=split(/¥,/,$');$fw_name="$name[$i]"."_"."$position[$i]"."_"."$m"."_fw";++$dcount;}#
        if($line=~m/PRIMER_RIGHT_¥d*=/){($m,$n)=split(/¥,/,$');$rv_name="$name[$i]"."_"."$position[$i]"."_"."$m"."_rv";++$dcount;}
        if($line=~m/PRIMER_PAIR_¥d*_PRODUCT_SIZE=/){$p_size=$',++$dcount;}

        }

       if($dcount==5){


       ####
       (undef,undef,$fw_m,undef)=split(/¥_/,$fw_name);


       $ref_target_seq=$seq[$i];

       $alt_target_seq=$seq[$i];
       substr($alt_target_seq,100,1,$alt);


       $ref_amp_seq=$ref_target_seq;
       $ref_amp_seq1=substr($ref_amp_seq,$fw_m,$p_size);

       $alt_amp_seq=$alt_target_seq;
       $alt_amp_seq1=substr($alt_amp_seq,$fw_m,$p_size);

       #print $ref_amp_seq1,"¥n";
       #print $alt_amp_seq1,"¥n";

       #$ref_dg_sug=&DELTA_G($ref_amp_seq1,¥%SUGI);
       #$alt_dg_sug=&DELTA_G($alt_amp_seq1,¥%SUGI);
       $ref_dg_unif=&DELTA_G($ref_amp_seq1,¥%UNIF);
       $alt_dg_unif=&DELTA_G($alt_amp_seq1,¥%UNIF);

       print $ref_dg_unif,"¥t",$alt_dg_unif;



       ####

       print $name[$i],"¥t",$position[$i],"¥t",$seq[$i],"¥t",$ref[$i],"¥t",$alt[$i],"¥t",$motif[$i],"¥t",$p_size,"¥t",$fw_name,"¥t",$fw,"¥t",$rv_name,"¥t",$rv,"¥t",$ref_dg_unif,"¥t",$alt_dg_unif,"¥n";#"¥t",$blast_fw,"¥t",$blast_rv,

       close(FILE2);

       unlink($fileout);
       unlink("test.tmp");
       }else{

       #print $name[$i],"¥t",$position[$i],"¥t",$ref[$i],"¥t",$alt[$i],"¥t",$motif[$i],"¥n";
              close(FILE2);

       unlink($fileout);
       unlink("test.tmp");
       }


    }
}

############
sub BLAST{
   my $l_count=0;
   my $p_name=$_[0];
   my $p_seq=$_[1];
   my $out="blastinput.txt";
   my $bout="blastout.txt";
   my $db="C:¥¥Glyma1¥¥Glyma1.fa";

   open(NEW,"> $out");
```

```
   print NEW ">","$p_name","¥n$p_seq¥n";
   close(NEW);

   system "blastn.exe -query $out -db C:¥¥Glyma1¥¥Glyma1.fa -outfmt 6 -max_target_seqs 10 -out $bout -evalue 0.01 -task blastn-short";

   open (FILE1,$bout) or die "not found your file $bout.¥n";
        while($line2=<FILE1>){
         ++$l_count;
         }

   close(FILE1);
   unlink($out);
   unlink($bout);

   return ($l_count);
}




###########################################################
sub DELTA_G{
my ($t_seq,$table) = @_;########
my %TABLE=%$table;
my @t_seq=split(//,$t_seq);
my $init_f=0;
my $init_r=0;
my $t_length=length($t_seq);
my $nn=();
my $delta_g=0;
my $i=0;

$init_f=$TABLE{$t_seq[0]};
$init_r=$TABLE{$t_seq[$t_length-1]};

for($i=0;$i<$t_length-1;++$i){
    $nn=$t_seq[$i].$t_seq[$i+1];

    $delta_g+=$TABLE{$nn};

    }

$delta_g=$delta_g+$init_f+$init_r;

$delta_g;

}
#end of VCF_to_HRMprimer_step2_exp1.pl
#######################################

#VCF_to_HRMprimer_noNNNs.pl

####################################################################################################################################
##
##   primer3_core is nesessary.
##
##  perl test4d.pl 170628out1.txt > 170818HRMout_reall_nosub.txt
##  perl VCF_to_HRMprimer2.pl 170628out1.txt >170628outHRM2.txt
####################################################################################################################################


$listfile=shift @ARGV;
open(FILE,$listfile) or die "not found your file.¥n";

while($line=<FILE>){
    chomp $line;
    if($line=~/[ATGCNX],[ATGCNX]/){next}#
    ($a,$x,$b,$c,$d,$e)=split(/¥t/,$line);
    push @name, $a;
    push @position, $x;
```

```
    push @seq,$b;
    push @motif, $c;
    push @ref, $d;
    push @alt, $e;
    }


$number=@name;#
$seq_num=0;#
$range =("50-100");#
$target=101;#
%case1=();
%case2=();

######################################
%UNIF=();
%UNIF=();

$file1="TD_table.txt";

open FILE1, $file1 or die "not found your $file1.¥n";

while($line=<FILE1>){
    chomp $line;
    ($nuc,$UNIF,$unif)=split(/¥t/,$line);
    $UNIF{$nuc}=$UNIF;
    $UNIF{$nuc}=$unif;
}

close(FILE1);
####################################



for($i;$i<$number;++$i){#

    $seq_num++;#

    $val = length($ref[$i]) - length($alt[$i]);#
    $length = length($seq[i]);#
    $subs=$ref[$i].$alt[$i];#

    #if($val==0 && $subs=~m/AC|AG|TG|TC|GA|GT|CT|CA|CG/i){#Transversion Type
    if($val==0 && $subs=~m/AC|AG|TG|TC|GA|GT|CT|CA|GC|CG|AT|TA/i){#


        $seq[$i]=~s/¥x0D¥x0A$|¥x0D$|¥x0A$//;#

########

#########CASE1
        @seq_array=split(//,$seq[$i]);

        $ref_triplet=$seq_array[99].$ref[$i].$seq_array[101];
        $alt_triplet=$seq_array[99].$alt[$i].$seq_array[101];

        $ori_ref_seq=$seq[$i];
        $c_ori_ref_seq=&compseq($seq[$i]);
        $ori_alt_seq=$seq[$i];

        substr($ori_alt_seq,100,1,$alt[$i]);
        $ori_d_ref=&DELTA_G($ori_ref_seq,¥%UNIF);
        $ori_d_alt=&DELTA_G($ori_alt_seq,¥%UNIF);
        $ori_ddg=abs($ori_d_ref - $ori_d_alt);
        $out_ori_ddg=sprintf ("%.3f", $ori_ddg);

        @nuc_array=("A","T","G","C");
        foreach $j (@nuc_array){

            $ref_seq=$seq[$i];
            $sub1=$j.$ref[$i];
            substr($ref_seq,99,2,$sub1);
```

```
            $alt_seq=$seq[$i];
            $sub2=$j.$alt[$i];
            substr($alt_seq,99,2,$sub2);

            $combi="$sub1"."_"."$sub2";

        $d_ref=&DELTA_G($ref_seq,¥%UNIF);
        $d_alt=&DELTA_G($alt_seq,¥%UNIF);

        $ddg=abs($d_ref-$d_alt);

        $case1{$combi}=$ddg;
        $ave_dfdt=($d_ref+$d_alt)*0.5;
        $case1_dfdt{$combi}=$ave_dfdt;

        $case1_eval{$combi}=$ave_dfdt/-100+$ddg;

        #print $name[$i],"¥t",$position[$i],"¥t",$combi,$case1{$combi},"¥t".$d_ref,"¥t",$d_alt,"¥t","Case1","¥n";
        }

        @array1=(sort {$case1_eval{$b} <=> $case1_eval{$a}} keys %case1_eval);
        #print $name[$i],"¥t",$position[$i],"¥t",$array1[0],"¥t",$case1{$array1[0]},"¥t",$case1_eval{$array1[0]},"¥t","
Case1","¥n";


######CASE2

        $c_seq=&compseq($seq[$i]);
        $c_ref=&compseq($ref[$i]);
        $c_alt=&compseq($alt[$i]);

        @c_seq_array=split(//,$c_seq);
        $c_ref_triplet=$c_seq_array[99].$c_ref.$seq_array[101];
        $c_alt_triplet=$c_seq_array[99].$c_alt.$seq_array[101];

        @nuc_array=("A","T","G","C");
        foreach $j (@nuc_array){

            $c_ref_seq=$c_seq;
            $c_sub1=$j.$c_ref;
            substr($c_ref_seq,99,2,$c_sub1);

            $c_alt_seq=$c_seq;
            $c_sub2=$j.$c_alt;
            substr($c_alt_seq,99,2,$c_sub2);

            $c_combi="$c_sub1"."_"."$c_sub2";

        $c_d_ref=&DELTA_G($c_ref_seq,¥%UNIF);
        $c_d_alt=&DELTA_G($c_alt_seq,¥%UNIF);

        $c_ddg=abs($c_d_ref-$c_d_alt);

        $case2{$c_combi}=$c_ddg;
        $c_ave_dfdt=($c_d_ref+$c_d_alt)*0.5;
        $case2_dfdt{$c_combi}=$c_ave_dfdt;

        $case2_eval{$c_combi}=$c_ave_dfdt/-100+$c_ddg;

        #print $name[$i],"¥t",$position[$i],"¥t",$c_combi,"¥t",$case2{$c_combi},"¥t".$c_d_ref,"¥t",$c_d_alt,"¥t","Case2"
,"¥n";
        }

        @array2=(sort {$case2_eval{$b} <=> $case2_eval{$a}} keys %case2_eval);
        #print $name[$i],"¥t",$position[$i],"¥t",$array2[0],"¥t",$case2{$array2[0]},"¥t",$case2_eval{$array2[0]},"¥t","
Case2","¥n";

        if($case1_eval{$array1[0]}>$case2_eval{$array2[0]}){

            ($f_nuc,$damy)=split(/¥_/,$array1[0]);

            #substr($ori_ref_seq,99,2,$f_nuc);
            $f_seq=$ori_ref_seq;
```

```
            $direction="FOR";
            $out_dfdt=$case1_dfdt{$array1[0]};
            $out_ddg=sprintf ("%.3f", $case1{$array1[0]});
            $out_nuc=$array1[0];
            }else{

            ($f_nuc,$damy)=split(/¥_/,$array2[0]);

            #substr($c_ori_ref_seq,99,2,$f_nuc);
            $f_seq=$c_ori_ref_seq;
            $direction="REV";
            $out_dfdt=$case2_dfdt{$array2[0]};
            $out_ddg=sprintf ("%.3f", $case2{$array2[0]});
            $out_nuc=$array2[0];

            }

        %case1=();
        %case1_dfdt=();
        %case1_eval=();
########
        %case2=();
        %case2_dfdt=();
        %case2_eval=();

#=test
        $file=sprintf "test.tmp";#
        $fileout=sprintf "$name[$i]_$position[$i]_out.tmp";#

        open FILE1,">$file";#

        print FILE1 "SEQUENCE_ID¥=$name[$i]_$position[$i]¥n";
        print FILE1 "SEQUENCE_TEMPLATE¥=$f_seq¥n";
        print FILE1 "SEQUENCE_TARGET="."$target".",1","¥n";
        print FILE1 "SEQUENCE_FORCE_LEFT_END=99¥n";
        print FILE1 "PRIMER_TASK=generic¥n";
        print FILE1 "PRIMER_PICK_LEFT_PRIMER=1¥n";
        print FILE1 "PRIMER_PICK_INTERNAL_OLIGO=0¥n";
        print FILE1 "PRIMER_PICK_RIGHT_PRIMER=1¥n";
        print FILE1 "PRIMER_OPT_SIZE=25¥n";
        print FILE1 "PRIMER_MIN_SIZE=18¥n";
        print FILE1 "PRIMER_MAX_SIZE=30¥n";
        print FILE1 "PRIMER_PRODUCT_SIZE_RANGE=$range¥n";
        print FILE1 "PRIMER_NUM_NS_ACCEPTED=1¥n";
        print FILE1 "PRIMER_NUM_RETURN=1¥n";
        print FILE1 "PRIMER_FILE_FLAG=1¥n";
        print FILE1 "PRIMER_EXPLAIN_FLAG=1¥n";
        print FILE1 "¥=¥n";
        close(FILE1);

        system "primer3_core <$file> $fileout";
        #print "$f_seq","¥n";

        $dcount=0;

        open FILE2,$fileout;

        while($line=<FILE2>){

        $line=~s/¥x0D¥x0A$|¥x0D$|¥x0A$//;#

        if($line=~m/PRIMER_LEFT_¥d*_SEQUENCE¥=/){$fw=$';++$dcount;}
        if($line=~m/PRIMER_RIGHT_¥d*_SEQUENCE¥=/){$rv=$';++$dcount;}
        if($line=~m/PRIMER_LEFT_¥d*=/){($m,$n)=split(/¥,/,$');$fw_name="$name[$i]"."_"."$position[$i]"."_"."$m"."_fw";++
$dcount;}#
        if($line=~m/PRIMER_RIGHT_¥d*=/){($m,$n)=split(/¥,/,$');$rv_name="$name[$i]"."_"."$position[$i]"."_"."$m"."_rv";+
+$dcount;}
        if($line=~m/PRIMER_PAIR_¥d*_PRODUCT_SIZE=/){$p_size=$',++$dcount;}

        }

       if($dcount==5){
       print $name[$i],"¥t",$position[$i],"¥t",$ref[$i],"¥t",$alt[$i],"¥t",$motif[$i],"¥t",$p_size,"¥t",$fw_name,"¥t",
```

```
$fw,"¥t",$rv_name,"¥t",$rv,"¥t",$direction,"¥t",$out_nuc,"¥t",$out_dfdt,"¥t",$out_ddg,"¥t",$ref_triplet,"¥t",
$alt_triplet,"¥t",$ori_d_ref,"¥t",$ori_d_alt,"¥t",$out_ori_ddg,"¥n";

       close(FILE2);

       unlink($fileout);
       unlink("test.tmp");
       }else{

       #print $name[$i],"¥t",$position[$i],"¥t",$ref[$i],"¥t",$alt[$i],"¥t",$motif[$i],"¥n";
              close(FILE2);

       unlink($fileout);
       unlink("test.tmp");
       }

#=cut
    }
}



###########################################################
sub DELTA_G{
my ($t_seq,$table) = @_;########
my %TABLE=%$table;
my @t_seq=split(//,$t_seq);
my $init_f=0;
my $init_r=0;
my $t_length=length($t_seq);
my $nn=();
my $delta_g=0;
my $i=0;

$init_f=$TABLE{$t_seq[0]};
$init_r=$TABLE{$t_seq[$t_length-1]};

for($i=0;$i<$t_length-1;++$i){
    $nn=$t_seq[$i].$t_seq[$i+1];

    $delta_g+=$TABLE{$nn};

    }

$delta_g=$delta_g+$init_f+$init_r;

$delta_g;

}
#######################################


sub compseq{
#Complementary sequence
  my ($seq) = @_;
  $seq=~s/¥x0D¥x0A$|¥x0D$|¥x0A$//;
        my $seq_1 = $seq =~ tr/AaTtGgCc/TtAaCcGg/r;

        my $seq_minus = reverse $seq_1;
        return "$seq_minus";
}
#####################
#end of VCF_to_HRMprimer_noNNNs.pl



#VCF_to_HRMprimer_NNNsHRM.pl

########################################################################################################################
############
##
##   primer3_core is nesessary.
##
```

```
##  perl VCF_to_HRMprimer_NNNsHRM.pl 180223test.txt > 180223out.txt
########################################################################################################################
############

$CONC=1;#1uM. 0.5->500nM

######################################
%SUGI=();
%UNIF=();

$file1="TD_table.txt";

open FILE1, $file1 or die "not found your $file1 containing the data for thermodyamics of nuclotides.¥n";

while($line=<FILE1>){
    chomp $line;
    ($nuc,$sugi,$unif,$delta_H,$delta_S)=split(/¥t/,$line);
    $SUGI{$nuc}=$sugi;
    $UNIF{$nuc}=$unif;
    $DH{$nuc}=$delta_H;
    $DS{$nuc}=$delta_S;
}

close(FILE1);
####################################



$listfile=shift @ARGV;
open(FILE,$listfile) or die "not found your file.¥n";

while($line=<FILE>){
    chomp $line;
    if($line=~/[ATGCNX],[ATGCNX]/){next}#
    ($a,$x,$b,$c,$d,$e)=split(/¥t/,$line);
    push @name, $a;
    push @position, $x;
    push @seq,$b;
    push @motif, $c;
    push @ref, $d;
    push @alt, $e;
    }


$number=@name;#
$seq_num=0;#
$range =("50-100");#
$target=101;#
%case1=();
%case2=();

######################################
%UNIF=();
%UNIF=();

$file1="TD_table.txt";

open FILE1, $file1 or die "not found your $file1.¥n";

while($line=<FILE1>){
    chomp $line;
    ($nuc,$UNIF,$unif)=split(/¥t/,$line);
    $UNIF{$nuc}=$UNIF;
    $UNIF{$nuc}=$unif;
}

close(FILE1);
####################################

print "CHR","¥t","POS","¥t","SEQ","¥t","REF","¥t","ALT","¥t","TYPE","¥t","PRODUCT_SIZE","¥t","FW_NAME","¥t","FW_SEQ","
¥t","RV_NAME","¥t","RV_SEQ","¥t","DELTA_G_REF_PRODUCT","¥t","DELTA_G_ALT_PRODUCT","¥t","REF_Tm","¥t","ALT_Tm","¥t","DIR"
,"¥t","COMBI_NUC","¥t","ddG_withSUB","¥t","ddG_NoSUB","¥t","REF_triplets","¥t","ALT_triplets","¥n";
```

```
for($i;$i<$number;++$i){#

    $seq_num++;#Sereal ID.

    $val = length($ref[$i]) - length($alt[$i]);#SNP check. 0 is SNP, not 0 is indel.
    $length = length($seq[i]);#length of sequence.
    $subs=$ref[$i].$alt[$i];#record of SNP

    if($val==0 && $subs=~m/AC|AG|TG|TC|GA|GT|CT|CA|CG|GC|AT|TA/i){#
        #print $name[$i],"¥t",$position[$i],"¥t",$val,"¥t",$subs,"¥t",$motif[$i],"¥t",$ref[$i],"¥t",$alt[$i],"¥n";#for
test

        $seq[$i]=~s/¥x0D¥x0A$|¥x0D$|¥x0A$//;#Deletion of new line code.

########

#########CASE1
        @seq_array=split(//,$seq[$i]);

        $ref_triplet=$seq_array[99].$ref[$i].$seq_array[101];
        $alt_triplet=$seq_array[99].$alt[$i].$seq_array[101];

        $ori_ref_seq=$seq[$i];
        $c_ori_ref_seq=&compseq($seq[$i]);
        $ori_alt_seq=$seq[$i];

        substr($ori_alt_seq,100,1,$alt[$i]);
        $ori_d_ref=&DELTA_G($ori_ref_seq,¥%UNIF);
        $ori_d_alt=&DELTA_G($ori_alt_seq,¥%UNIF);
        $ori_ddg=abs($ori_d_ref - $ori_d_alt);
        $out_ori_ddg=sprintf ("%.3f", $ori_ddg);

        @nuc_array=("A","T","G","C");
        foreach $j (@nuc_array){

            $ref_seq=$seq[$i];
            $sub1=$j.$ref[$i];
            substr($ref_seq,99,2,$sub1);

            $alt_seq=$seq[$i];
            $sub2=$j.$alt[$i];
            substr($alt_seq,99,2,$sub2);

            $combi="$sub1"."_"."$sub2";

        $d_ref=&DELTA_G($ref_seq,¥%UNIF);
        $d_alt=&DELTA_G($alt_seq,¥%UNIF);

        $ddg=abs($d_ref-$d_alt);

        $case1{$combi}=$ddg;
        $ave_dfdt=($d_ref+$d_alt)*0.5;
        $case1_dfdt{$combi}=$ave_dfdt;

        $case1_eval{$combi}=$ave_dfdt/-100+$ddg;

        #print $name[$i],"¥t",$position[$i],"¥t",$combi,$case1{$combi},"¥t".$d_ref,"¥t",$d_alt,"¥t","Case1","¥n";
        }

        @array1=(sort {$case1_eval{$b} <=> $case1_eval{$a}} keys %case1_eval);
        #print $name[$i],"¥t",$position[$i],"¥t",$array1[0],"¥t",$case1{$array1[0]},"¥t",$case1_eval{$array1[0]},"¥t","
Case1","¥n";


######CASE2

        $c_seq=&compseq($seq[$i]);
        $c_ref=&compseq($ref[$i]);
        $c_alt=&compseq($alt[$i]);

        @c_seq_array=split(//,$c_seq);
        $c_ref_triplet=$c_seq_array[99].$c_ref.$seq_array[101];
        $c_alt_triplet=$c_seq_array[99].$c_alt.$seq_array[101];
```

```
        @nuc_array=("A","T","G","C");
        foreach $j (@nuc_array){

            $c_ref_seq=$c_seq;
            $c_sub1=$j.$c_ref;
            substr($c_ref_seq,99,2,$c_sub1);

            $c_alt_seq=$c_seq;
            $c_sub2=$j.$c_alt;
            substr($c_alt_seq,99,2,$c_sub2);

            $c_combi="$c_sub1"."_"."$c_sub2";

        $c_d_ref=&DELTA_G($c_ref_seq,¥%UNIF);
        $c_d_alt=&DELTA_G($c_alt_seq,¥%UNIF);

        $c_ddg=abs($c_d_ref-$c_d_alt);

        $case2{$c_combi}=$c_ddg;
        $c_ave_dfdt=($c_d_ref+$c_d_alt)*0.5;
        $case2_dfdt{$c_combi}=$c_ave_dfdt;

        $case2_eval{$c_combi}=$c_ave_dfdt/-100+$c_ddg;

        #print $name[$i],"¥t",$position[$i],"¥t",$c_combi,"¥t",$case2{$c_combi},"¥t".$c_d_ref,"¥t",$c_d_alt,"¥t","Case2"
,"¥n";
        }

        @array2=(sort {$case2_eval{$b} <=> $case2_eval{$a}} keys %case2_eval);
        #print $name[$i],"¥t",$position[$i],"¥t",$array2[0],"¥t",$case2{$array2[0]},"¥t",$case2_eval{$array2[0]},"¥t","
Case2","¥n";

        if($case1_eval{$array1[0]}>$case2_eval{$array2[0]}){

            ($f_nuc,$g_nuc)=split(/¥_/,$array1[0]);

            substr($ori_ref_seq,99,2,$f_nuc);
            $f_seq=$ori_ref_seq;

            substr($ori_ref_seq,99,2,$g_nuc);
            $g_seq=$ori_ref_seq;

            $direction="FOR";
            $out_dfdt=$case1_dfdt{$array1[0]};
            $out_ddg=sprintf ("%.3f", $case1{$array1[0]});
            $out_nuc=$array1[0];
            }else{

            ($f_nuc,$g_nuc)=split(/¥_/,$array2[0]);

            substr($c_ori_ref_seq,99,2,$f_nuc);
            $f_seq=$c_ori_ref_seq;

            substr($c_ori_ref_seq,99,2,$g_nuc);
            $g_seq=$c_ori_ref_seq;

            $direction="REV";
            $out_dfdt=$case2_dfdt{$array2[0]};
            $out_ddg=sprintf ("%.3f", $case2{$array2[0]});
            $out_nuc=$array2[0];

            }

        %case1=();
        %case1_dfdt=();
        %case1_eval=();
########
        %case2=();
        %case2_dfdt=();
        %case2_eval=();

#=test
```

```
        $file=sprintf "test.tmp";#Making input file for Primer3.
        $fileout=sprintf "$name[$i]_$position[$i]_out.tmp";#Output file for Primer3

        open FILE1,">$file";

        print FILE1 "SEQUENCE_ID¥=$name[$i]_$position[$i]¥n";
        print FILE1 "SEQUENCE_TEMPLATE¥=$f_seq¥n";
        print FILE1 "SEQUENCE_TARGET=". "$target".",1","¥n";
        print FILE1 "SEQUENCE_FORCE_LEFT_END=99¥n";
        print FILE1 "PRIMER_TASK=generic¥n";
        print FILE1 "PRIMER_PICK_LEFT_PRIMER=1¥n";
        print FILE1 "PRIMER_PICK_INTERNAL_OLIGO=0¥n";
        print FILE1 "PRIMER_PICK_RIGHT_PRIMER=1¥n";
        print FILE1 "PRIMER_OPT_SIZE=25¥n";
        print FILE1 "PRIMER_MIN_SIZE=18¥n";
        print FILE1 "PRIMER_MAX_SIZE=30¥n";
        print FILE1 "PRIMER_PRODUCT_SIZE_RANGE=$range¥n";
        print FILE1 "PRIMER_NUM_NS_ACCEPTED=1¥n";
        print FILE1 "PRIMER_NUM_RETURN=1¥n";
        print FILE1 "PRIMER_FILE_FLAG=1¥n";
        print FILE1 "PRIMER_EXPLAIN_FLAG=1¥n";
        print FILE1 "¥=¥n";
        close(FILE1);

        system "primer3_core <$file> $fileout";
        #print "$f_seq","¥n";

        $dcount=0;

        open FILE2,$fileout;

        while($line=<FILE2>){

        $line=~s/¥x0D¥x0A$|¥x0D$|¥x0A$//;#Deletion of new line code.

        if($line=~m/PRIMER_LEFT_¥d*_SEQUENCE¥=/){$fw=$';++$dcount;}
        if($line=~m/PRIMER_RIGHT_¥d*_SEQUENCE¥=/){$rv=$';++$dcount;}
        if($line=~m/PRIMER_LEFT_¥d*=/){($m,$n)=split(/¥,/,$');$fw_name="$name[$i]"."_"."$position[$i]"."_"."$m"."_fw";++
$dcount;}#
        if($line=~m/PRIMER_RIGHT_¥d*=/){($m,$n)=split(/¥,/,$');$rv_name="$name[$i]"."_"."$position[$i]"."_"."$m"."_rv";+
+$dcount;}
        if($line=~m/PRIMER_PAIR_¥d*_PRODUCT_SIZE=/){$p_size=$',++$dcount;}

        }

       if($dcount==5){

       (undef,undef,$fw_m,undef)=split(/¥_/,$fw_name);

       $ref_target_seq=$f_seq;

       $alt_target_seq=$g_seq;

       $ref_amp_seq1=substr($ref_target_seq,$fw_m,$p_size);

       $alt_amp_seq1=substr($alt_target_seq,$fw_m,$p_size);

       $ref_pp_dg_unif=&DELTA_G($ref_amp_seq1,¥%UNIF);
       $alt_pp_dg_unif=&DELTA_G($alt_amp_seq1,¥%UNIF);

       $ref_dh_unif=&DELTA_G($ref_amp_seq1,¥%DH);
       $alt_dh_unif=&DELTA_G($alt_amp_seq1,¥%DH);

       $ref_ds_unif=&DELTA_G($ref_amp_seq1,¥%DS);
       $alt_ds_unif=&DELTA_G($alt_amp_seq1,¥%DS);

       #print $ref_dh_unif,"¥t",$alt_dh_unif,"¥t",$ref_ds_unif,"¥t",$alt_ds_unif,"¥n";

       $ref_tm=&TMCALC($ref_dh_unif,$ref_ds_unif,$CONC);
       $alt_tm=&TMCALC($alt_dh_unif,$alt_ds_unif,$CONC);

       print $name[$i],"¥t",$position[$i],"¥t",$f_seq,"¥t",$ref[$i],"¥t",$alt[$i],"¥t",$motif[$i],"¥t",$p_size,"¥t",
$fw_name,"¥t",$fw,"¥t",$rv_name,"¥t",$rv,"¥t",$ref_pp_dg_unif,"¥t",$alt_pp_dg_unif,"¥t",$ref_tm,"¥t",$alt_tm,"¥t",
```

```
$direction,"¥t",$out_nuc,"¥t",$out_ddg,"¥t",$out_ori_ddg,"¥t",$ref_triplet,"¥t",$alt_triplet,"¥n";

        close(FILE2);

        unlink($fileout);
        unlink("test.tmp");
        }else{

        #print $name[$i],"¥t",$position[$i],"¥t",$ref[$i],"¥t",$alt[$i],"¥t",$motif[$i],"¥n";
                close(FILE2);

        unlink($fileout);
        unlink("test.tmp");
        }

#=cut
    }
}



###########################################################
sub DELTA_G{
my ($t_seq,$table) = @_;########
my %TABLE=%$table;
my @t_seq=split(//,$t_seq);
my $init_f=0;
my $init_r=0;
my $t_length=length($t_seq);
my $nn=();
my $delta_g=0;
my $i=0;

$init_f=$TABLE{$t_seq[0]};
$init_r=$TABLE{$t_seq[$t_length-1]};

for($i=0;$i<$t_length-1;++$i){
    $nn=$t_seq[$i].$t_seq[$i+1];

    $delta_g+=$TABLE{$nn};

    }

$delta_g=$delta_g+$init_f+$init_r;

$delta_g;

}
#######################################


sub compseq{
#Complementary sequence
  my ($seq) = @_;
  $seq=~s/¥x0D¥x0A$|¥x0D$|¥x0A$//;
        my $seq_1 = $seq =~ tr/AaTtGgCc/TtAaCcGg/r;

        my $seq_minus = reverse $seq_1;
        return "$seq_minus";
}
#####################

sub TMCALC{
my ($dh,$ds,$conc)=@_;
my $tm=0;
my $ct=$conc/1000000;#if $conc=1, it incdicate 1uM.

$tm=((1000*$dh)/($ds+1.987*log($ct/4)))-273;

$tm = sprintf("%.2f", $tm);
$tm;

}
```

```
#end of VCF_to_HRMprimer_NNNsHRM.pl
########################################################################################################################
###########

#VCF_to_HRM_blast.pl

#/usr/bin/perl

########################################################################################################################
###########
##
##  For executing this script, set the PATH to blast+.
##  Type "blastn.exe -h", you can confirm a PATH to blast programs on a command prompt.
##
##  perl VCF_to_HRM_blast.pl C:¥Glyma1¥Glyma1.fa 170914HRMout_8tr.txt > 170914HRMout_final.txt
##
########################################################################################################################
############



$db=shift @ARGV; #'C:¥Glyma1¥Glyma1.fa';

$out="tempout.txt";
$bout="blastout.txt";
$listfile=shift @ARGV;
open(FILE,$listfile) or die "not found your file.¥n";

$line=FILE;#

while($line=<FILE>){
    chomp $line;
    open(NEW,"> $out");
    ($a,$x,$seq,$ref,$alt,$mot,$psize,$fw_name,$b,$rv_name,$c)=split(/¥t/,$line);#Gm01  40354   GAATGCAGCAGACATGTGTACC
TCTTCTCTGGAAATGCTGCAGT  337 G   C   HindIII
    $ID="$a"."_"."$x";
    $ID{$ID}+=1;
    $fw=$ID."_fw";
    $rv=$ID."_rv";
    print NEW ">","$fw","¥n$b¥n",">","$rv","¥n$c¥n";
    close(NEW);

#=comment
    system "blastn.exe -query $out -db $db -outfmt 6 -max_target_seqs 10 -out $bout -evalue 0.01 -task blastn-short";

    open (FILE1,$bout) or die "not found your file $bout.¥n";
        while($line2=<FILE1>){
            chomp $line2;
            #print $line2,"¥n";
            ($primer,@array)=split(/¥t/,$line2);
            $primer{$primer}+=1;
            }

            print $line,"¥t";

            $k_num=keys %primer;

            if($k_num>1){
              foreach $j (sort keys %primer){

                  print $primer{$j},"¥t";

                  }
              }else{
              print "ND","¥t","ND","¥t";
              }



            #print $line2,"¥n";
    %primer=();
    close(FILE1);
```

```
    unlink($out);
    unlink($bout);


    print "¥n";
}

#=cut

#end of VCF_to_HRM_blast.pl
#######
```