

User Manual for EasyKASP: KASP primer design

Updated on October 15, 2025

Disclaimer: Although our lab at Beijing Vegetable Research Center (BVRC) has conducted extensive testing and KASP primers designed by EasyKASP are generally reliable. However, not all primers are guaranteed to produce successful genotyping results. For critical SNP or InDel variations where EasyKASP fails to design primers, we recommend users exploring other available software.

If you have any questions or suggestions, please email us at BVRC_SNP@126.com.

1 Introduction

As a third-generation molecular marker technology, single nucleotide polymorphisms (SNPs) are ideal for biological selective breeding due to their wide distribution, high throughput, genetic stability, and ease of standardization. The most commonly used method for SNP genotyping is Kompetitive Allele-Specific PCR (KASP), which offers high throughput, low cost, and high accuracy. KASP technology can also be applied to genotype certain InDel variations.

The Vegetable Research Center of the Beijing Academy of Agriculture and Forestry Sciences was the first in China to adopt an LGC Genomics system for high-throughput SNP genotyping based on KASP technology. Currently, our lab designs at least 10,000 KASP primers annually and successfully genotypes over one million SNP data points. We have established a robust KASP primer design pipeline, with an experimentally verified success rate exceeding 95%.

Based on this experience, we developed EasyKASP—an Excel VBA-based tool for designing KASP primers for both SNP and InDel variations.

2 Installing EasyKASP

2.1 System Requirements

EasyKASP was developed on Excel-VBA platform, an Office application development language, and runs directly in Microsoft Office installed without requiring an internet connection.

Hardware: CPU with at least one core, RAM $\geq$ 1GB.

Software: Microsoft Office Excel (version 2007 or later recommended).

2.2 Installation steps

Note: *If you have already have an Excel file containing the EasyKASP macros, you may skip this section.*

Step 1: Create a New Excel File and Enable the Developer Tab

- By default, VBA developer tools are hidden. To enable them:
- Go to **File** → **Options** → **Customize Ribbon**
- In the right-hand panel, check the box for Developer
- Click **OK**. The **Developer** tab will now appear in the Excel ribbon.

Step 2: Open the VBA Editor and Insert the Code

- Go to the **Developer** tab and click **Visual Basic** (Figure 1A).
- In the VBA window, go to **Insert** → **Module**. A new module named "Module1" will appear.
- Double-click the module to open the code window and paste the EasyKASP source code into it (Figure 1B).

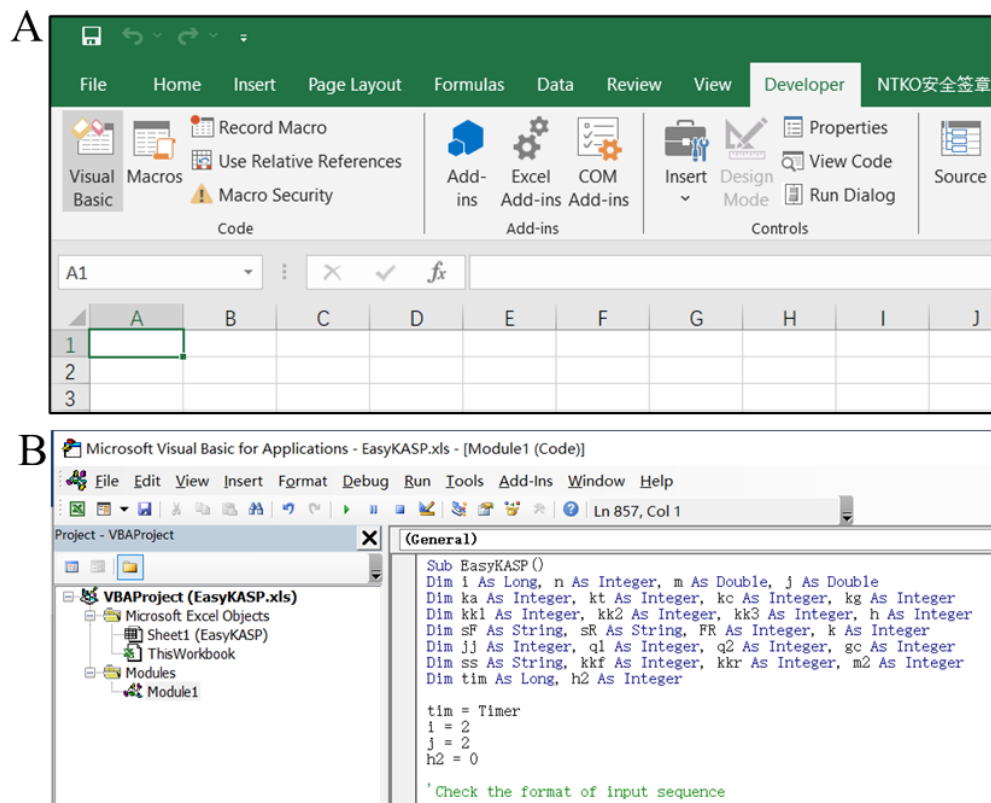


Fig. 1 Installation of EasyKASP in Excel VBA

Step 3: Assign a Macro Button

- Enter the following column headers in the first four columns: **SNP ID**, **Allele 1**, **Allele 2**, and **Flanking Sequence**.
- Go to the **Home** tab, insert a shape, and label it (e.g., "Start") (**Figure 2A**).
- Right-click the shape, select **Assign Macro**, and choose the **EasyKASP** macro (**Figure 2B**). Save the file in **.xlsm** format.
- Congratulations—EasyKASP is now installed and ready to use.

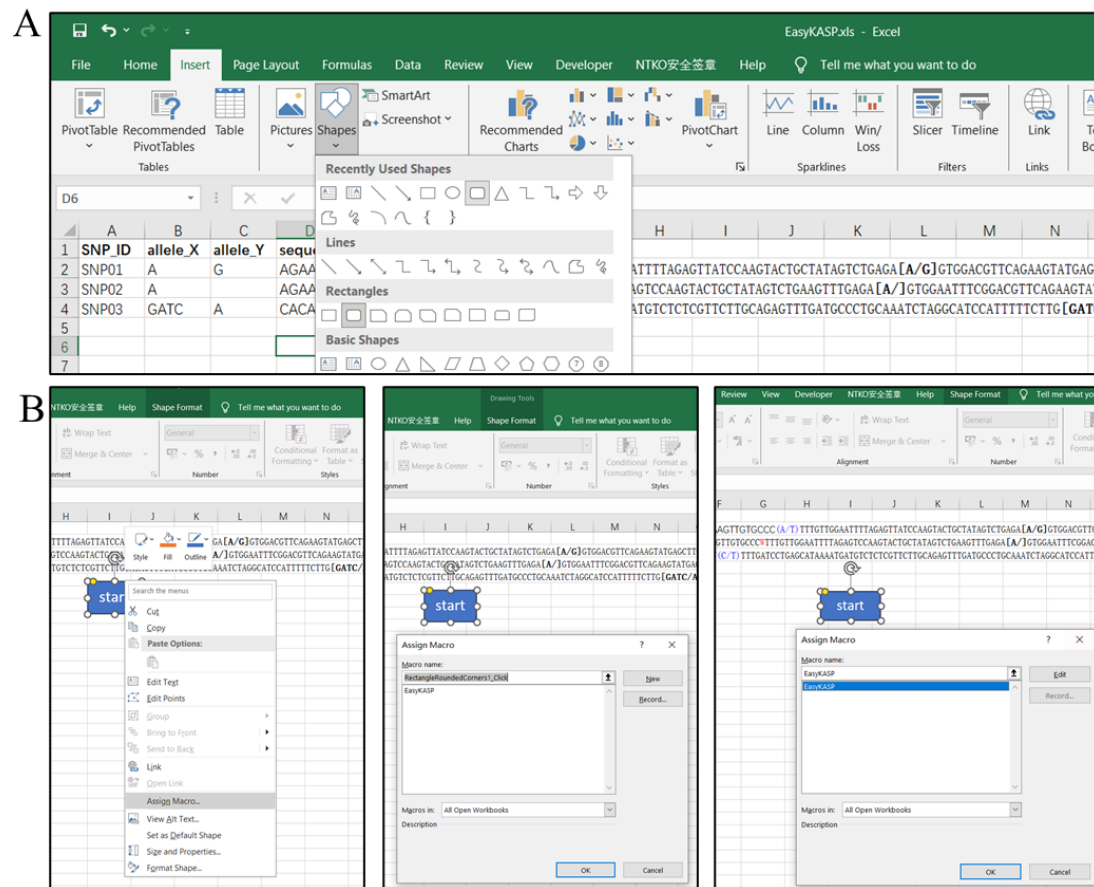


Fig. 2 Assign a start button to EasyKASP

3. Designing KASP Primers

3.1 Input Variation and Flanking Sequence Information

Prepare your data in four columns as shown in **Figure 3**:

- Column A: Names of SNP or InDel variation sites
- Column B: Nucleotide of the first allele
- Column C: Nucleotide of the second allele
- Column D: Flanking sequence of the variation site

Important Notes: The nucleotide variations in Columns B and C must exactly match the bracketed base in the flanking sequence. Provide at least 60 bp of flanking sequence on each side of the variation. Enclose the target SNP or InDel in **brackets** []. Use **parentheses** () or **degenerate bases** to indicate other variations. Do not use spaces or hyphens for missing nucleotides.

	A	B	C	D	E	F	G
1	SNP_ID	allele_X	allele_Y	sequence	primer_X	primer_Y	primer_C
2	SNP_01	C	T	AAAAATTAATAAATCTCGGTCAATTTTGGTCAATTTGACCGGCACAAACCAAGGTGACAACTAAA[C/T]	GGGAAGGAGGGAGTAAAAATATAACATTACCAGATCAGCAATCTGTTATAA	CTT	
3	SNP_02	T	C	TTCTTTCTAGTCTGTATAGGACTTAACCTAGCCTGAGAAATACCTTATAATGATTGGTGATTATGCA[T/C]	GACTAATGACTAATAGGTAGGAGGTAACTCTACTCAAGGCTAGCCGATTTAAATTTCT		
4	InDel_03		A	GACTCGCACATCGCTTCGGCTTCGGACTTCTCTTTGGGCGGCGATTGGTAGCAATCGCGTGGCTGTAT[A]	TGTTCTGTCGAAAGACTACCACTCAGTGGCGAATTCGATTAATCGCGTAGTTTGGAT		
5	SNP_04	T	C	TAAAGGTATATTAATGATTAGGTGTAAGACATGAACCTGGAAAAATATAAAGCTGGTGTAGTGAT[T/C]	ATGAATGAGTCTAGGAAATAAATGGTAGAATAAGCGGCCATACCACCGATATG		
6	SNP_05	A	C	AAAAATGTTTTAAATAGGTGTTTGAGACTTGAATTAGTCAAAATTCACAAAGAAAGTAAAAATAAA[A/C]	GAGGGTCGTAAATGATTATCTGTTTTCATTTCATTGACTCGAAGTCTCCCATG		
7	SNP_06	G	A	AGGCATCCTAGGACATGTTGGATCAAGTTTTTCATCAATTGCCCTAAATATAAATAGGACATGAT[G/A]	TAGGGCATCTCTCTGTTGATGCAGGACAAGTTGAGAATGATGGATATAGAATG		
8	InDel_07	T		CATCCAGTTT TGATGACCGTGGTGAATGTTGCCCAAGGAGTCACCGACGTGTTGGGAACCATCAAGACCAAGCGTAC[T/C]	CATCCAGTTTGGTGAATGTTGGTGGTCTCTGTTGATGATGCGGTATCACTACCGACC		
9	SNP_08	A	G	TCCCGCTAATTTGTTGTTATGCACAAACGCTTACTCAAGCTGATTTTGGTTTATTTCCC[A/G]	TTTTCGGGAAAAAGATTTTGGTTGATTTTACTTACATTTTATAGGTATT		
10	SNP_09	T	A	ATGGGTCTCTTACCCACTTTTCTTCCCTTTTACACTACTACCTACTCAATTTATTTTCTT[T/A]	TATACCTTAAAGTTAAGTGAATGATCTACCAATTAACCCACTTTTCTCTATTTCAGT		
11	InDel_10	A		GACTGTAGCG TTGTACCCCTATTAAATATGTTTCTGTATCGGCCATAGTCTTATGGTAAGCTTTGACGCTGAGAT[A/G]	GACTGTAGCGTACGCTGCTGAAGAGATTGATGGCTCCATCAAACTCAACGCGTTGT		
12	SNP_11	T	C	AAAAATAGTCTCTGAATTTTCTATAGCTTTTTCACCTTTTCGACACGCAATTCATCTCTTA[T/C]	AAAAATAGTCTCTGAATTTTTCAGATTTTTCATTCTGAATAAAAAATAAAAAC		
13	SNP_12	T	C	TTTGACATTTTGGTCACTTCTAAATTTACAGAATTATAGAAGTGCAACGAGAAATTTGAAACATT[T/C]	AAGTGTTTTCCAAAAATTTGCTGATGCCCGAGGAAATTCCTCTAGATCGCGCC		
14	SNP_13	T	C	CGGAAAGATAATCCATAACCGGCAAGCAGCTTTGGTAAATGGTTTATGATTTTATCTT[T/C]	TTTATTAATTTTATCGGTTTCTAAATTTTAAATAATATAATATCTTTTATTTG		
15	SNP_14	C	A	CCGCAATTGGCGTATTTCATGGATATGTGAACAAATCTGTCTGCAATGTTTGAATGAGTGAAGACATA[C/A]	CTCTAAAGAGATTTTACTCCAGATATCATTTATTTGTCACAGAAATTAACCAAT		
16	SNP_15	G	T	GTGTGTATATCGAAGGAGGAGGGTTGGACTGGTGGTGGTGGCGCCGATCTGTTAGATGGCCA[G/T]	CCTACAATCTGTGCTGGTGTATGCAAGAGTCTGAGATGGAGATGAGAAATGAGGA		
17	SNP_16	C	T	ACCAAACTTTTCACTGTCTTCCAGTAGGTAGCTGGGCGACGTTTCAAGGATCATCTCAATCA[C/T]	GGTGTATATTTATCAGAAGTCAATTTTCAATTTGATTTGATTTAGCAATTTTGAAGCAAT		
18	SNP_17	T	A	TAAACCATTTTCAATTTCTAATCTCAGATTGAAATTCATTAATTTTGTGTGCAAGACGATGAAAGACAT[A/G]	TGGTTAGTCCCATCACAACATCATATCCAGTCCATCCGCTACATTTACCT		
19	SNP_18	A	G	ACCATACCGGCACATGGATTGGGGTTGACGAATGAAAGAAATTTAGTCACTATTACAGACAAAAG[A/G]	CTATCATCTATACAAGAAATTTGCTCAACCAATTTAGAGAGAAAGAGATGCGC		
20	SNP_19	C	T	GGCCCATGTATCAATATTTTACAATTTGATAGACACAGGATGACAGACTATCCGGTATCTCAG[C/T]	TGATATGAGTTCATCAATTTGATCAATCTTTTGGTCTATCATTTGGATGATAGC		
21	SNP_20	G	A	TGTACCACTCAATAAATTTTGTGTCATGGTCCAGGAATATCAACGAGAAACGGATTTTATGATTATG[G/A]	CTAAAAATTTGACGTAATTTCTACGCGGATTTATCATATAGTCAAGTTTATATCTCCC		

Fig. 3 Prepare DNA sequence before designing KASP primers

3.2 Running EasyKASP

Once the sequences are prepared, click the **"Start"** button. The tool will automatically design primers. A confirmation message will appear upon completion—click **"OK"**.

The output includes: **Primer_X** and **Primer_Y**: Allele-specific primers, each with a 21 bp adapter sequence. **Primer_C**: Universal primer sequence. Sites that fail to design will be labeled **"Failed"** (Figure 4).

	A	B	C	D	E	F	G
1	SNP_ID	allele_X	allele_Y	sequence	primer_X	primer_Y	primer_C
2	SNP_01	C	T	AAAAATTTAA GAAGGTGACCAAGTTCATGCTTTATAATTTTACTCCTCTCTCCCG	GAAGGTGCGAGTCAACGGATTGTTATAATTTTACTCCTCTCTCCCA	TTTGACCGGCACAAACCAAGG	
3	SNP_02	T	C	TTCTTTCTAG GAAGGTGACCAAGTTCATGCTACCTCTACCTATTAGTCATTAGTCA	GAAGGTGCGAGTCAACGGATTCTCCTCTACCTATTAGTCATTAGTTCG	AGCCTGAGAAATCTTTATAATG	
4	InDel_03		A	GACTCGCACA GAAGGTGACCAAGTTCATGCTGAGTGGTAGCTTCCAGCAACAA	GAAGGTGCGAGTCAACGGATTGAGTGGTAGCTTCTGAGCAACAT	TCCTTTGGCGCGATTTGGTAGC	
5	SNP_04	T	C	TAAAGGTATA GAAGGTGACCAAGTTCATGCTGGAATAAATAAAGCTGGTGTAGTGA	GAAGGTGCGAGTCAACGGATTGAAAAATAAATAAGCTGGTGTAGTGA	TGCTTTATCTACCAATTTATTTT	
6	SNP_05	A	C	AAAAATGTTT GAAGGTGACCAAGTTCATGCTGAACAAAGA		TACGACCTCG	CCCTGAATTAGTCAAAATTCAT
7	SNP_06	G	A	AGGCATCACT GAAGGTGACCAAGTTCATGCTCATCAACA	EasyKASP: Easy for KASP primer design	ATGCCCTAT	TCATCAATTTGCCCTAAATTTAT
8	InDel_07	T		CATCCAGTTT TGATGTACCG GAAGGTGACCAAGTTCATGCTGAACCATG		CCGTACC	CTGGTAGTTGATACCGCACTTG
9	SNP_08	A	G	TCCCGCTAA GAAGGTGACCAAGTTCAT		CTTTATTTCCCG	GGAGAAAAATACCTATTAAAAAT
10	SNP_09	T	A	ATGGGTCTCT GAAGGTGACCAAGTTCAT		CTTTTAAGGTATA	CTTCCTTTTCAGTTTATTAAC
11	InDel_10	A		TGTTACCCCG GAAGGTGACCAAGTTCATGCTGGTAAGCG		CATATGATCT	CTACCTACAGCAACTTAACAG
12	SNP_11	T	C	AAAAATAGT GAAGGTGACCAAGTTCATGCTTTTCGACACGCTTTCATCTCTTAT	GAAGGTGCGAGTCAACGGATTTCGACACGCTTTCATCTCTTAC	TCAGATGAAAAATCTCAAAAT	
13	SNP_12	T	C	TTTGACATT GAAGGTGACCAAGTTCATGCTGAAGTGAACGAAAAATTTGAAACAT	GAAGGTGCGAGTCAACGGATTAGTGCACACGAAAAATTTGAAACATCT	CTCGGGCGATCAACGAAATTTT	
14	SNP_13	T	C	CGGAAAGAT Failed			
15	SNP_14	C	A	CCGCAATTGGC GAAGGTGACCAAGTTCATGCTATATCGGAGTAAATCTCTTTAGAG	GAAGGTGCGAGTCAACGGATTGATATCTGGAGTAAATCTCTTTAGAGT	CAAACTGTCTGCATAGTTT	
16	SNP_15	G	T	GTGTGTATA GAAGGTGACCAAGTTCATGCTATACACCGACACAGTTGATAGG	GAAGGTGCGAGTCAACGGATTATACACCGACACAGTTGATAGGA	TGCGCCCGCGATCTGTTAGAT	
17	SNP_16	C	T	ACCAAACTT GAAGGTGACCAAGTTCATGCTGGTTTCAAGGATCATCTCAATCAC	GAAGGTGCGAGTCAACGGATTACGTTTCAAGGATCATCTCAATCAC	CCAATATGAACATGACTCTG	
18	SNP_17	T	A	TAAACCATTT GAAGGTGACCAAGTTCATGCTGATGTTTGTGTGGGACTAAACCA	GAAGGTGCGAGTCAACGGATTGATGTTTGTGTGGGACTAAACCAT	TTGAATTCATTAATTTGTTG	
19	SNP_18	A	G	ACCATACCG GAAGGTGACCAAGTTCATGCTGAAATTTTATGACACTATTACAGACAAAA	GAAGGTGCGAGTCAACGGATTAAATTTTATGACACTATTACAGACAAAA	TCTCATATTTGTTGAGCAAA	
20	SNP_19	C	T	GGCCCATGTA GAAGGTGACCAAGTTCATGCTCAGACTCATCGGTATCTCAG	GAAGGTGCGAGTCAACGGATTACAGACTCATCGGTATCTCAGT	AGACCAAAAGGATGATCAAA	
21	SNP_20	G	A	TGTACCACTC GAAGGTGACCAAGTTCATGCTCCGTGAGAATTTAGTCAAAATTTTAGC	GAAGGTGCGAGTCAACGGATTCCGTGAGAATTTAGTCAAAATTTTAGT	GAAGTATCCACGACGAAACG	

Fig. 4 Result of designed KASP primer by EasyKASP

4. Additional Notes

KASP genotyping relies on fluorescence signals rather than amplicon length, which limits the flexibility of primer design. Allele-specific primers must be located within 20–30 bp upstream or downstream of the target SNP/InDel, and amplicon lengths are typically between 50–100 bp.

Not all genetic variations are suitable for KASP assays, and not all designed primers will yield successful genotyping results. To ensure specificity:

- Primers must be single-copy in the genome to avoid nonspecific amplification.
- Flanking sequence alignment or experimental validation is recommended to confirm product uniqueness.
- Non-target variations in flanking regions should be considered, as they may affect primer binding.
- The quality of the reference genome and potential sequencing errors may influence sequence accuracy.
- Undiscovered variations near the target site may interfere with KASP genotyping.

For critical genetic variations, we recommend verifying the target and flanking regions via **Sanger sequencing** before designing KASP primers.

Revised by Jian Zhang

Contact: BVRC_SNP@126.com

The source code of EasyKASP in Excel VBA

```
Sub EasyKASP()  
Dim i As Long, n As Integer, m As Double, j As Double  
Dim ka As Integer, kt As Integer, kc As Integer, kg As Integer  
Dim kk1 As Integer, kk2 As Integer, kk3 As Integer, h As Integer  
Dim sF As String, sR As String, FR As Integer, k As Integer  
Dim jj As Integer, q1 As Integer, q2 As Integer, gc As Integer  
Dim ss As String, kkf As Integer, kkr As Integer, m2 As Integer  
Dim tim As Long, h2 As Integer  
tim = Timer  
i = 2  
j = 2  
h2 = 0  
'Check the format of input sequence  
Cells(1, 1) = "SNP_ID"  
Cells(1, 2) = "allele_X"  
Cells(1, 3) = "allele_Y"  
Cells(1, 4) = "Sequence"  
Cells(1, 5) = "primer_X"  
Cells(1, 6) = "primer_Y"  
Cells(1, 7) = "primer_C"  
Do While Len(Cells(i, 1)) <> 0  
Application.ScreenUpdating = False  
Cells(i, 4) = UCase(WorksheetFunction.Substitute(WorksheetFunction.Clean(Trim(Cells(i, 4))),  
" ", ""))  
Cells(i, 4) = WorksheetFunction.Substitute(Cells(i, 4), "-", "")  
Cells(i, 4) = WorksheetFunction.Substitute(Cells(i, 4), "_", "")  
FR = 0  
sR = ""  
sF = ""  
m = 0  
ss = ""  
gc = 0  
h = 0  
If Len(Cells(i, 4)) - Len(WorksheetFunction.Substitute(Cells(i, 4), "[", "")) > 1 Then Cells(i, 1) =  
Cells(i, 1) & " Variation mistake"  
If Len(Cells(i, 4)) - Len(WorksheetFunction.Substitute(Cells(i, 4), "[", "")) > 1 Then Cells(i, 5) =  
"Failed"  
  
Do While FR = 0  
If InStr(1, Cells(i, 4), "[") = 0 Or InStr(1, Cells(i, 4), "]") = 0 Or InStr(1, Cells(i, 4), "/") = 0 Then  
Cells(i, 5) = "Failed"  
If InStr(1, Cells(i, 4), ",") <> 0 Or InStr(1, Cells(i, 4), "\") <> 0 Or InStr(1, Cells(i, 4), "**") <> 0
```

```

Then Cells(i, 5) = "Failed"
If Cells(i, 5) = "Failed" Then Exit Do
kk1 = 0
kk2 = 0
kk3 = 0
q1 = 0
q2 = 0
ka = 0
kt = 0
kc = 0
kg = 0
k = 0

```

Determine the location of the variant site

```

kk1 = WorksheetFunction.Find("[", Cells(i, 4), 1)
kk2 = WorksheetFunction.Find("/", Mid(Cells(i, 4), kk1 + 1, Len(Cells(i, 4)) - kk1), 1)
kk2 = kk2 + kk1
kk3 = WorksheetFunction.Find("]", Cells(i, 4), 1)

If UCase(Cells(i, 2)) <> Mid(Cells(i, 4), kk1 + 1, kk2 - kk1 - 1) Or UCase(Cells(i, 3)) <>
Mid(Cells(i, 4), kk2 + 1, kk3 - kk2 - 1) Then Cells(i, 1) = Cells(i, 1) & " Allele mistake"
Cells(i, 2) = Mid(Cells(i, 4), kk1 + 1, kk2 - kk1 - 1)
Cells(i, 3) = Mid(Cells(i, 4), kk2 + 1, kk3 - kk2 - 1)

```

```

Do While Right(Cells(i, 1), 7) = "mistake"
    Cells(i, 1).Select
    With Selection.Interior
        .PatternColorIndex = xlAutomatic
        .Color = 255
    End With
Exit Do
Loop

```

Determine the direction of primer design by evaluating the sequences on both sides

```

If kk1 >= 27 Then sF = Mid(Cells(i, 4), kk1 - 26, 26) & Cells(i, 2)
If kk1 < 27 Then sF = Mid(Cells(i, 4), 1, kk1 - 1) & Cells(i, 2)
sR = Cells(i, 2) & Mid(Cells(i, 4), kk3 + 1, 26)
sF = Right(sF, 26)
sR = Left(sR, 26)

n = Len(sF)
Do While n > 0

```

If Mid(sF, n, 1) = "A" Then ka = ka + 1

If Mid(sF, n, 1) = "T" Then kt = kt + 1

If Mid(sF, n, 1) = "C" Then kc = kc + 1

If Mid(sF, n, 1) = "G" Then kg = kg + 1

n = n - 1

Loop

Check for repetitive sequences

If InStr(1, sF, "AAAAAA") <> 0 Or InStr(1, sF, "TTTTTT") <> 0 Or InStr(1, sF, "CCCCCC") <> 0 Or InStr(1, sF, "GGGGGG") <> 0 Or Len(sF) <> ka + kt + kg + kc Then FR = -1

m = (kg + kc) / (kg + kc + kt + ka) * 100

ka = 0

kt = 0

kc = 0

kg = 0

n = Len(sR)

Do While n > 0

If UCase(Mid(sR, n, 1)) = "A" Then ka = ka + 1

If UCase(Mid(sR, n, 1)) = "T" Then kt = kt + 1

If UCase(Mid(sR, n, 1)) = "C" Then kc = kc + 1

If UCase(Mid(sR, n, 1)) = "G" Then kg = kg + 1

n = n - 1

Loop

If InStr(1, sR, "AAAAAA") <> 0 Or InStr(1, sR, "TTTTTT") <> 0 Or InStr(1, sR, "CCCCCC") <> 0 Or InStr(1, sR, "GGGGGG") <> 0 Or Len(sR) <> ka + kg + kc + kt Then FR = FR + 100

If FR = 99 Then Cells(i, 5) = "Failed"

If FR = 99 Then Exit Do

If FR = 100 Then FR = 1

kkf = 0

Do While Len(Cells(i, 2)) <> 1 Or Len(Cells(i, 3)) <> 1

kkf = 1

kk3 = 1

h = 999

Do While Len(Cells(i, 2)) >= Len(Cells(i, 3))

sF = Left(Cells(i, 2), 6)

sR = Cells(i, 3) & Mid(Cells(i, 4), kk3 + 1, 6)

Do While kkf < 6

If Left(sF, kkf) <> Left(sR, kkf) Then Exit Do

If Left(sF, kkf) = Left(sR, kkf) Then kkf = kkf + 1

Loop

```

sF = Right(Cells(i, 2), 6)
sR = Mid(Cells(i, 4), kk1 - 6, 6) & Cells(i, 3)
Do While kkr < 6
    If Right(sF, kkr) <> Right(sR, kkr) Then Exit Do
    If Right(sF, kkr) = Right(sR, kkr) Then kkr = kkr + 1
Loop

```

```

If WorksheetFunction.Min(kkf, kkr) = 5 Then Cells(i, 5) = "Failed"
If FR = 1 And kkf = 5 Then Cells(i, 5) = "Failed"
If FR = -1 And kkr = 5 Then Cells(i, 5) = "Failed"
If kkf < kkr And FR = 0 Then FR = 1
If kkf > kkr And FR = 0 Then FR = -1

```

```

Exit Do
Loop

```

```

Do While Len(Cells(i, 2)) < Len(Cells(i, 3))
    sF = Left(Cells(i, 3), 6)
    sR = Cells(i, 2) & Mid(Cells(i, 4), kk3 + 1, 6)

```

```

Do While kkf < 6
    If Left(sF, kkf) <> Left(sR, kkf) Then Exit Do
    If Left(sF, kkf) = Left(sR, kkf) Then kkf = kkf + 1
Loop

```

```

sF = Right(Cells(i, 3), 6)
sR = Mid(Cells(i, 4), kk1 - 6, 6) & Cells(i, 2)

```

```

Do While kkr < 6
    If Right(sF, kkr) <> Right(sR, kkr) Then Exit Do
    If Right(sF, kkr) = Right(sR, kkr) Then kkr = kkr + 1

```

```

Loop

```

```

If WorksheetFunction.Min(kkf, kkr) = 6 Then Cells(i, 5) = "Failed"
If FR = 1 And kkf = 5 Then Cells(i, 5) = "Failed"
If FR = -1 And kkr = 5 Then Cells(i, 5) = "Failed"
If kkf < kkr And FR = 0 Then FR = 1
If kkf > kkr And FR = 0 Then FR = -1

```

```

Exit Do
Loop

```

```

Exit Do
Loop

```

If Cells(i, 2) = Cells(i, 3) Then Cells(i, 5) = "Failed"

If Cells(i, 5) = "Failed" Then Exit Do

If Abs(m - 45) <= Abs(((kg + kc) / (kg + kc + kt + ka) * 100) - 45) And FR = 0 Then gc = 1

If Abs(m - 45) <= Abs(((kg + kc) / (kg + kc + kt + ka) * 100) - 45) And FR = 0 Then FR = 1

If Abs(m - 45) > Abs(((kg + kc) / (kg + kc + kt + ka) * 100) - 45) And FR = 0 Then gc = 2

If Abs(m - 45) > Abs(((kg + kc) / (kg + kc + kt + ka) * 100) - 45) And FR = 0 Then FR = -1

Do While gc = 1 Or gc = 2

If InStr(Mid(Cells(i, 4), kk1 - 3, 3), "AAA") <> 0 Or InStr(Mid(Cells(i, 4), kk1 - 3, 3), "TTT") <> 0 Then FR = -1

If InStr(Mid(Cells(i, 4), kk3 + 1, 4), "AAAA") <> 0 Or InStr(Mid(Cells(i, 4), kk3 + 1, 4), "TTTT") <> 0 Then FR = 1

If InStr(Mid(Cells(i, 4), kk3 + 1, 3), "AAA") <> 0 Or InStr(Mid(Cells(i, 4), kk3 + 1, 3), "TTT") <> 0 Then FR = 1

If InStr(Mid(Cells(i, 4), kk1 - 3, 4), "AAAA") <> 0 Or InStr(Mid(Cells(i, 4), kk1 - 3, 4), "TTTT") <> 0 Then FR = -1

Exit Do

Loop

n = 0

m = 0

If Len(Cells(i, 4)) - kk3 < 70 Then q2 = 70 - (Len(Cells(i, 4)) - kk3)

If q2 > 0 Then n = q2

Do While n > 0

Cells(i, 4) = Cells(i, 4) & "Q"

n = n - 1

Loop

n = 0

Addition of the flanking sequence which was less than 70bp

If kk1 < 70 Then q1 = 70 - kk1

If q1 > 0 Then n = q1

Do While n > 0

Cells(i, 4) = "Q" & Cells(i, 4)

n = n - 1

Loop

n = 0

kk1 = WorksheetFunction.Find("[", Cells(i, 4), 1)

kk2 = WorksheetFunction.Find("]", Mid(Cells(i, 4), kk1 + 1, Len(Cells(i, 4)) - kk1), 1)

kk2 = kk2 + kk1

```
kk3 = WorksheetFunction.Find("]", Cells(i, 4), 1)
```

```
sR = ""
```

```
sF = ""
```

```
m2 = 1
```

```
ss = ""
```

```
Do While Len(Cells(i, 2)) = 1 And Len(Cells(i, 3)) = 1
```

```
m2 = 3
```

```
ss = Cells(i, 2) & Cells(i, 3)
```

```
If ss = "AT" Or ss = "GC" Or ss = "TA" Or ss = "CG" Then m2 = 0
```

```
ss = ""
```

```
Exit Do
```

```
Loop
```

Forward designing primers

Designing the first specific primer

```
Do While FR = 1 Or FR = 0
```

```
k = 38
```

```
ss = Mid(Cells(i, 4), kk1 - 40, 40) & Cells(i, 2)
```

```
If h = 999 And Len(Cells(i, 2)) <= 38 And Len(Cells(i, 2)) >= kkf Then ss = Mid(Cells(i, 4), kk1 - 40, 40) & Left(Cells(i, 2), kkf)
```

```
If h = 999 And Len(Cells(i, 2)) < kkf Then ss = Mid(Cells(i, 4), kk1 - 40, 40) & Cells(i, 2) & Mid(Cells(i, 4), kk3 + 1, kkf - Len(Cells(i, 2)))
```

```
Do While Len(Cells(i, 8)) = 0 Or Cells(i, 8) >= 56.6
```

```
If Cells(i, 8) >= 56.6 Then k = k - 1
```

```
sF = Right(ss, k)
```

```
n = Len(sF)
```

```
If n < 18 And Cells(i, 8) <= 59 And Cells(i, 8) > 0 Then sF = Right(ss, k + 1)
```

```
If k < 18 Then Exit Do
```

```
ka = 0
```

```
kt = 0
```

```
kg = 0
```

```
kc = 0
```

```
Do While n > 0 Calculate the length and GC content of candidate primer
```

```
If UCase(Mid(sF, n, 1)) = "A" Then ka = ka + 1
```

```
If UCase(Mid(sF, n, 1)) = "T" Then kt = kt + 1
```

```
If UCase(Mid(sF, n, 1)) = "C" Then kc = kc + 1
```

```
If UCase(Mid(sF, n, 1)) = "G" Then kg = kg + 1
```

```
n = n - 1
```

```
Loop
```

```
j = 0
```

```
If ka + kt + kc + kg < 18 Then Exit Do
```

```
Select the optimal primer sequence
```

If Len(sF) = ka + kt + kg + kc And (kg + kc) / (ka + kt + kg + kc) <= 0.72 And (kg + kc) / (ka + kt + kg + kc) >= 0.16 Then j = 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc)

If j >= 56.6 And j <= 59 Then Cells(i, 8) = j

Do While Cells(i, 8) > 0

If InStr(1, sF, "AAAAAA") <> 0 Or InStr(1, sF, "TTTTTT") <> 0 Or InStr(1, sF, "CCCCCC") <> 0 Or InStr(1, sF, "GGGGGG") <> 0 Then Cells(i, 8) = ""

Exit Do

Loop

If Cells(i, 8) > 0 And Cells(i, 8) <= 57.8 Then Exit Do

If Cells(i, 8) > 0 And Len(sF) = ka + kt + kg + kc And Len(sF) = ka + kt + kg + kc And 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc - 1) < 56.1 Then sF = Right(ss, k + 1)

If Cells(i, 8) > 0 And 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc - 1) < 56.1 Then Exit Do

If Len(Cells(i, 8)) = 0 Then k = k - 1

Loop

If Cells(i, 8) > 0 And Cells(i, 8) <= 59 Then Cells(i, 5) = sF

If Len(Cells(i, 8)) = 0 Or Cells(i, 8) > 64.6 Then Cells(i, 5) = "Failed"

Cells(i, 8) = ""

sR = ""

sF = ""

ss = ""

Designing the seconde specific primer

Do While Cells(i, 5) <> "Failed"

k = 38

If m2 = 0 Or m2 = 3 Then k = Len(Cells(i, 5)) + m2

ss = Mid(Cells(i, 4), kk1 - 40, 40) & Cells(i, 3)

If h = 999 And Len(Cells(i, 3)) <= 38 And Len(Cells(i, 3)) >= kkf Then ss = Mid(Cells(i, 4), kk1 - 40, 40) & Left(Cells(i, 3), kkf)

If h = 999 And Len(Cells(i, 3)) < kkf Then ss = Mid(Cells(i, 4), kk1 - 40, 40) & Cells(i, 3) & Mid(Cells(i, 4), kk3 + 1, kkf - Len(Cells(i, 3)))

Do While Len(Cells(i, 9)) = 0 Or Cells(i, 9) >= 56.6

If Cells(i, 9) >= 56.6 Then k = k - 1

sF = Right(ss, k)

n = Len(sF)

If n < 18 And Cells(i, 9) <= 59 And Cells(i, 9) > 0 Then sF = Right(ss, k + 1)

If k < 18 Then Exit Do

ka = 0

kt = 0

kg = 0

```

kc = 0
Do While n > 0
  If UCase(Mid(sF, n, 1)) = "A" Then ka = ka + 1
  If UCase(Mid(sF, n, 1)) = "T" Then kt = kt + 1
  If UCase(Mid(sF, n, 1)) = "C" Then kc = kc + 1
  If UCase(Mid(sF, n, 1)) = "G" Then kg = kg + 1
  n = n - 1
Loop
If ka + kt + kc + kg < 18 Then Exit Do
j = 0
  If Len(sF) = ka + kt + kg + kc And (kg + kc) / (ka + kt + kg + kc) <= 0.72 And (kg + kc) / (ka +
kt + kg + kc) >= 0.16 Then j = 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc)
  If j >= 56.6 And j <= 59 Then Cells(i, 9) = j

  Do While Cells(i, 9) > 0
    If InStr(1, sF, "AAAAAA") <> 0 Or InStr(1, sF, "TTTTTT") <> 0 Or InStr(1, sF, "CCCCCC")
<> 0 Or InStr(1, sF, "GGGGGG") <> 0 Then Cells(i, 9) = ""
    Exit Do
  Loop

  If Cells(i, 9) > 0 And Cells(i, 9) <= 63.3 Then Exit Do
  If Cells(i, 9) > 0 And Len(sF) = ka + kt + kg + kc And 72 + (37.4 * (kg + kc) - 747) / (ka + kt +
kg + kc - 1) < 56.1 Then sF = Right(ss, k + 1)
  If Cells(i, 9) > 0 And 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc - 1) < 56.1 Then Exit Do

  If Len(Cells(i, 9)) = 0 Then k = k - 1

Loop
If Cells(i, 9) > 0 And Left(sF, 4) = "TTTT" Then sF = Mid(sF, 2, 38)
If Cells(i, 9) > 0 And Cells(i, 9) <= 59 Then Cells(i, 6) = sF
If Len(Cells(i, 9)) = 0 Or Cells(i, 9) > 59 Then Cells(i, 5) = "Failed"

Exit Do
Loop
  Cells(i, 9) = ""
  sR = ""
  sF = ""
  ss = ""

```

Designing common primer

```

jj = 4
Do While Cells(i, 5) <> "Failed"

```

```

Do While jj <= 30
    k = 40
    sF = Mid(Cells(i, 4), kk3 + jj, k)
    j = 2
Do While Len(Cells(i, 10)) = 0 Or Cells(i, 10) >= 59.9
    If Cells(i, 10) >= 59.9 Then k = k - 1
    sF = Left(sF, k)
    n = Len(sF)
    If n < 20 And Cells(i, 10) >= 59.9 Then sF = Mid(Cells(i, 4), kk3 + jj, k + 1)
    If n < 20 Then Exit Do
    ka = 0
    kt = 0
    kg = 0
    kc = 0
Do While n > 0
    If UCase(Mid(sF, n, 1)) = "A" Then ka = ka + 1
    If UCase(Mid(sF, n, 1)) = "T" Then kt = kt + 1
    If UCase(Mid(sF, n, 1)) = "C" Then kc = kc + 1
    If UCase(Mid(sF, n, 1)) = "G" Then kg = kg + 1
    n = n - 1
Loop
    If ka + kt + kc + kg < 20 Then Exit Do
    j = 0
    If Len(sF) = ka + kt + kg + kc And (kg + kc) / (ka + kt + kg + kc) <= 0.7 And (kg + kc) / (ka +
kt + kg + kc) >= 0.21 Then j = 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc)
    If j >= 59.9 And j <= 62.9 Then Cells(i, 10) = j

    Do While Abs(Cells(i, 10) - 60.4) <= 0.57
        If InStr(1, sF, "AAAAAA") <> 0 Or InStr(1, sF, "TTTTTT") <> 0 Or InStr(1, sF, "CCCCCC")
<> 0 Or InStr(1, sF, "GGGGGG") <> 0 Then Cells(i, 10) = ""
        If Left(sF, 4) = "AAAA" Or Left(sF, 4) = "TTTT" <> 0 Or Left(sF, 4) = "CCCC" <> 0 Or
Left(sF, 4) = "GGGG" <> 0 Then Cells(i, 10) = ""
        If Right(sF, 4) = "AAAA" Or Right(sF, 4) = "TTTT" <> 0 Or Right(sF, 4) = "CCCC" <> 0 Or
Right(sF, 4) = "GGGG" <> 0 Then Cells(i, 10) = ""
        Exit Do
    Loop

    If Cells(i, 10) >= 59.9 And Len(sF) = ka + kt + kg + kc And 72 + (37.4 * (kg + kc) - 747) / (ka +
kt + kg + kc - 1) < 59.34 Then sF = Mid(Cells(i, 4), kk3 + jj, k + 1)
    If Cells(i, 10) >= 59.9 And 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc - 1) < 59.34 Then
Exit Do

    If Len(Cells(i, 10)) = 0 Then k = k - 1

```

Loop

```
If Len(Cells(i, 7)) = 0 And Abs(Cells(i, 10) - 60.4) <= 0.57 Then Cells(i, 7) = sF
Do While Abs(Cells(i, 10) - 60.4) <= 0.57 And Len(Cells(i, 7)) <> 0
  If Left(sF, 1) <> "A" And Left(sF, 1) <> "T" And Abs(Len(sF) - 29.5) <= 4.5 Then Cells(i, 7) =
sF
  If Left(Cells(i, 7), 1) <> "G" And Left(Cells(i, 7), 1) <> "C" And Abs(Len(sF) - 29.5) <= 4.5
Then Cells(i, 7) = sF
  If Cells(i, 7) <> sF And Abs(Len(Cells(i, 7)) - 29) > 5 And Abs(Len(sF) - 29) <= Abs(Len(Cells(i,
7)) - 29) Then Cells(i, 7) = sF
Exit Do
```

Loop

```
jj = jj + 1
If jj >= 10 And Left(Cells(i, 7), 1) <> "A" And Left(Cells(i, 7), 1) <> "T" And Len(Cells(i, 7)) <>
0 Then Exit Do
If jj >= 20 And Len(Cells(i, 7)) <> 0 Then Exit Do
Cells(i, 10) = ""
```

Loop

```
If Len(Cells(i, 7)) = 0 Then Cells(i, 5) = "Failed"
If Cells(i, 5) = "Failed" Then Cells(i, 10) = ""
```

```
If Len(Cells(i, 7)) <> 0 Then sF = Cells(i, 7)
If Len(Cells(i, 7)) = 0 Then sF = ""
If Len(Cells(i, 7)) <> 0 Then Cells(i, 7) = ""
n = Len(sF)
Do While n > 0 'Sequence reverse complementation
  If UCase(Mid(sF, n, 1)) = "A" Then Cells(i, 7) = Cells(i, 7) & "T"
  If UCase(Mid(sF, n, 1)) = "T" Then Cells(i, 7) = Cells(i, 7) & "A"
  If UCase(Mid(sF, n, 1)) = "C" Then Cells(i, 7) = Cells(i, 7) & "G"
  If UCase(Mid(sF, n, 1)) = "G" Then Cells(i, 7) = Cells(i, 7) & "C"
  n = n - 1
Loop
```

Exit Do

Loop

Exit Do

Loop

'Reverse designing primers

```

If gc = 1 And Cells(i, 5) = "Failed" Then FR = -1
If kkf <= kkr And kkf > 0 And Cells(i, 5) = "Failed" Then FR = -1
If gc = 1 And Cells(i, 5) = "Failed" Then Cells(i, 5) = ""
If gc = 1 And Cells(i, 5) = "Failed" Then Cells(i, 6) = ""
If gc = 1 And Cells(i, 5) = "Failed" Then Cells(i, 7) = ""

```

```

sR = ""
sF = ""
ss = ""
j = 2

```

Designing the first specific primer

```

Do While FR = -1
    k = 38
    ss = Cells(i, 2) & Mid(Cells(i, 4), kk3 + 1, 40)

    If h = 999 And Len(Cells(i, 2)) <= 38 And Len(Cells(i, 2)) >= kkr Then ss = Right(Cells(i, 2),
kkr) & Mid(Cells(i, 4), kk3 + 1, 40)
    If h = 999 And Len(Cells(i, 2)) < kkr Then ss = Mid(Cells(i, 4), kk1 - (kkr - Len(Cells(i, 2))),
kkr - Len(Cells(i, 2))) & Cells(i, 2) & Mid(Cells(i, 4), kk3 + 1, 40)

```

```

Do While Len(Cells(i, 8)) = 0 Or Cells(i, 8) >= 56.6
    If Cells(i, 8) >= 56.6 Then k = k - 1
    sR = Left(ss, k)
    n = Len(sR)
    If n < 18 And Cells(i, 8) <= 59 And Cells(i, 8) > 0 Then sR = Left(ss, k + 1)
    If k < 18 Then Exit Do
    ka = 0
    kt = 0
    kg = 0
    kc = 0
    Do While n > 0
        If UCase(Mid(sR, n, 1)) = "A" Then ka = ka + 1
        If UCase(Mid(sR, n, 1)) = "T" Then kt = kt + 1
        If UCase(Mid(sR, n, 1)) = "C" Then kc = kc + 1
        If UCase(Mid(sR, n, 1)) = "G" Then kg = kg + 1
        n = n - 1
    Loop
    If ka + kt + kc + kg < 18 Then Exit Do
    j = 0
    If Len(sR) = ka + kt + kg + kc And (kg + kc) / (ka + kt + kg + kc) <= 0.72 And (kg + kc) / (ka +
kt + kg + kc) >= 0.16 Then j = 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc)
    If j >= 56.6 And j <= 59 Then Cells(i, 8) = j
    Do While Cells(i, 8) > 0

```

```
If InStr(1, sR, "AAAAAA") <> 0 Or InStr(1, sR, "TTTTTT") <> 0 Or InStr(1, sR, "CCCCC")
<> 0 Or InStr(1, sR, "GGGGGG") <> 0 Then Cells(i, 8) = ""
```

```
Exit Do
```

```
Loop
```

```
If Cells(i, 8) > 0 And Cells(i, 8) <= 57.8 Then Exit Do
```

```
If Cells(i, 8) > 0 And Len(sR) = ka + kt + kg + kc And 72 + (37.4 * (kg + kc) - 747) / (ka + kt +
kg + kc - 1) < 56.1 Then sR = Left(ss, k + 1)
```

```
If Cells(i, 8) > 0 And 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc - 1) < 56.1 Then Exit Do
```

```
If Len(Cells(i, 8)) = 0 Then k = k - 1
```

```
Loop
```

```
n = Len(sR)
```

```
Cells(i, 5) = ""
```

```
Do While n > 0 And Cells(i, 8) > 0 And Cells(i, 8) <= 59
```

Sequence reverse complementation

```
If UCase(Mid(sR, n, 1)) = "A" Then Cells(i, 5) = Cells(i, 5) & "T"
```

```
If UCase(Mid(sR, n, 1)) = "T" Then Cells(i, 5) = Cells(i, 5) & "A"
```

```
If UCase(Mid(sR, n, 1)) = "C" Then Cells(i, 5) = Cells(i, 5) & "G"
```

```
If UCase(Mid(sR, n, 1)) = "G" Then Cells(i, 5) = Cells(i, 5) & "C"
```

```
n = n - 1
```

```
Loop
```

```
If Len(Cells(i, 8)) = 0 Or Cells(i, 8) > 64.6 Then Cells(i, 5) = "Failed"
```

```
Cells(i, 8) = ""
```

```
sR = ""
```

```
sF = ""
```

```
ss = ""
```

Designing the seconde specific primer

```
Do While Cells(i, 5) <> "Failed"
```

```
k = 38
```

```
If m2 = 0 Or m2 = 3 Then k = Len(Cells(i, 5)) + m2
```

```
ss = Cells(i, 3) & Mid(Cells(i, 4), kk3 + 1, 40)
```

```
If h = 999 And Len(Cells(i, 3)) >= kkr And Len(Cells(i, 3)) <= 38 Then ss = Right(Cells(i, 3), kkr)
& Mid(Cells(i, 4), kk3 + 1, 40)
```

```
If h = 999 And Len(Cells(i, 3)) < kkr Then ss = Mid(Cells(i, 4), kk1 - (kkr - Len(Cells(i, 3))), kkr -
Len(Cells(i, 3))) & Cells(i, 3) & Mid(Cells(i, 4), kk3 + 1, 40)
```

```
Do While Len(Cells(i, 9)) = 0 Or Cells(i, 9) >= 56.6
```

```
If Cells(i, 9) >= 56.6 Then k = k - 1
```

```
sR = Left(ss, k)
```

```
n = Len(sR)
```

```

If n < 18 And Cells(i, 9) <= 59 And Cells(i, 9) > 0 Then sR = Left(ss, k + 1)
If k < 18 Then Exit Do
ka = 0
kt = 0
kg = 0
kc = 0
Do While n > 0
    If UCase(Mid(sR, n, 1)) = "A" Then ka = ka + 1
    If UCase(Mid(sR, n, 1)) = "T" Then kt = kt + 1
    If UCase(Mid(sR, n, 1)) = "C" Then kc = kc + 1
    If UCase(Mid(sR, n, 1)) = "G" Then kg = kg + 1
    n = n - 1
Loop
    If ka + kt + kc + kg < 18 Then Exit Do
    j = 0
    If Len(sR) = ka + kt + kg + kc And (kg + kc) / (ka + kt + kg + kc) <= 0.72 And (kg + kc) / (ka +
kt + kg + kc) >= 0.16 Then j = 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc)
    If j >= 56.6 And j <= 59 Then Cells(i, 9) = j
    Do While Cells(i, 9) > 0
        If InStr(1, sR, "AAAAAA") <> 0 Or InStr(1, sR, "TTTTTT") <> 0 Or InStr(1, sR, "CCCCCC")
<> 0 Or InStr(1, sR, "GGGGGG") <> 0 Then Cells(i, 9) = ""
        Exit Do
    Loop

    If Cells(i, 9) > 0 And Cells(i, 9) <= 63.3 Then Exit Do
    If Cells(i, 9) > 0 And Len(sR) = ka + kt + kg + kc And 72 + (37.4 * (kg + kc) - 747) / (ka + kt +
kg + kc - 1) < 56.1 Then sR = Left(ss, k + 1)
    If Cells(i, 9) > 0 And 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc - 1) < 56.1 Then Exit Do

    If Len(Cells(i, 9)) = 0 Then k = k - 1

Loop

If Cells(i, 9) > 0 Then n = Len(sR)
Cells(i, 6) = ""
Do While n > 0 And Cells(i, 9) > 0 And Cells(i, 9) <= 59
Sequence reverse complementation
    If UCase(Mid(sR, n, 1)) = "A" Then Cells(i, 6) = Cells(i, 6) & "T"
    If UCase(Mid(sR, n, 1)) = "T" Then Cells(i, 6) = Cells(i, 6) & "A"
    If UCase(Mid(sR, n, 1)) = "C" Then Cells(i, 6) = Cells(i, 6) & "G"
    If UCase(Mid(sR, n, 1)) = "G" Then Cells(i, 6) = Cells(i, 6) & "C"
    n = n - 1
Loop

```

If Len(Cells(i, 9)) = 0 Or Cells(i, 9) > 59 Then Cells(i, 5) = "Failed"

Exit Do

Loop

Cells(i, 9) = ""

sR = ""

sF = ""

ss = ""

jj = 4

Do While Cells(i, 5) <> "Failed"

Designing common primer

Do While jj <= 30

k = 40

If kk1 < k + jj + 18 Then Exit Do

sF = Mid(Cells(i, 4), kk1 - k - jj, k)

j = 2

Do While Len(Cells(i, 10)) = 0 Or Cells(i, 10) >= 59.9

If Cells(i, 10) >= 59.9 Then k = k - 1

sF = Right(sF, k)

n = Len(sF)

If n < 20 And Cells(i, 10) >= 59.9 Then sF = Mid(Cells(i, 4), kk1 - k - 1 - jj, k + 1)

If n < 20 Then Exit Do

ka = 0

kt = 0

kg = 0

kc = 0

Do While n > 0

If UCase(Mid(sF, n, 1)) = "A" Then ka = ka + 1

If UCase(Mid(sF, n, 1)) = "T" Then kt = kt + 1

If UCase(Mid(sF, n, 1)) = "C" Then kc = kc + 1

If UCase(Mid(sF, n, 1)) = "G" Then kg = kg + 1

n = n - 1

Loop

If ka + kt + kc + kg < 20 Then Exit Do

j = 0

If Len(sF) = ka + kt + kg + kc And (kg + kc) / (ka + kt + kg + kc) <= 0.7 And (kg + kc) / (ka + kt + kg + kc) >= 0.21 Then j = 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc)

If j >= 59.9 And j <= 62.9 Then Cells(i, 10) = j

Do While Abs(Cells(i, 10) - 60.4) <= 0.57

If InStr(1, sF, "AAAAAA") <> 0 Or InStr(1, sF, "TTTTTT") <> 0 Or InStr(1, sF, "CCCCCC")

```

<> 0 Or InStr(1, sF, "GGGGGG") <> 0 Then Cells(i, 10) = ""
    If Left(sF, 4) = "AAAA" Or Left(sF, 4) = "TTTT" <> 0 Or Left(sF, 4) = "CCCC" <> 0 Or
Left(sF, 4) = "GGGG" <> 0 Then Cells(i, 10) = ""
    If Right(sF, 4) = "AAAA" Or Right(sF, 4) = "TTTT" <> 0 Or Right(sF, 4) = "CCCC" <> 0 Or
Right(sF, 4) = "GGGG" <> 0 Then Cells(i, 10) = ""
    Exit Do
    Loop

```

```

    If Cells(i, 10) >= 59.9 And Len(sF) = ka + kt + kg + kc And 72 + (37.4 * (kg + kc) - 747) / (ka +
kt + kg + kc - 1) < 59.34 Then sF = Mid(Cells(i, 4), kk1 - k - 1 - jj, k + 1)
    If Cells(i, 10) >= 59.9 And 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc - 1) < 59.34 Then
Exit Do

```

```

    If Len(Cells(i, 10)) = 0 Then k = k - 1

```

```

    Loop

```

```

    If Len(Cells(i, 7)) = 0 And Abs(Cells(i, 10) - 60.4) <= 0.57 Then Cells(i, 7) = sF

```

```

    Do While Abs(Cells(i, 10) - 60.4) <= 0.57 And Len(Cells(i, 7)) <> 0
        If Right(sF, 1) <> "A" And Right(sF, 1) <> "T" And Abs(Len(sF) - 29.5) <= 4.5 Then Cells(i, 7)
= sF
        If Right(Cells(i, 7), 1) <> "G" And Right(Cells(i, 7), 1) <> "C" And Abs(Len(sF) - 29.5) <= 4.5
Then Cells(i, 7) = sF
        If Cells(i, 7) <> sF And Abs(Len(Cells(i, 7)) - 29) > 5 And Abs(Len(sF) - 29) <= Abs(Len(Cells(i,
7)) - 29) Then Cells(i, 7) = sF
    Exit Do
    Loop

```

```

    jj = jj + 1
    If jj >= 10 And Right(Cells(i, 7), 1) <> "A" And Right(Cells(i, 7), 1) <> "T" And Len(Cells(i, 7))
<> 0 Then Exit Do
    If jj >= 20 And Len(Cells(i, 7)) <> 0 Then Exit Do
    Cells(i, 10) = ""

```

```

    Loop
    If Len(Cells(i, 7)) = 0 Then Cells(i, 5) = "Failed"
    If Cells(i, 5) = "Failed" Then Cells(i, 10) = ""

```

```

    Exit Do
    Loop

```

```

    Exit Do
    Loop

```

Exit Do

Loop

' If the reverse design of the primers was failed, it was needed to add one more forward primer design

If gc = 2 And Cells(i, 5) = "Failed" Then FR = 3

If kkf > kkr And FR = -1 And kkf < 4 And Cells(i, 5) = "Failed" Then FR = 3

If FR = 3 Then Cells(i, 5) = ""

If FR = 3 Then Cells(i, 6) = ""

If FR = 3 Then Cells(i, 7) = ""

sR = ""

sF = ""

ss = ""

Do While FR = 3

Cells(i, 8) = ""

Cells(i, 9) = ""

Cells(i, 10) = ""

'Designing first specific primer

k = 38

ss = Mid(Cells(i, 4), kk1 - 40, 40) & Cells(i, 2)

If h = 999 And Len(Cells(i, 2)) <= 38 And Len(Cells(i, 2)) >= kkf Then ss = Mid(Cells(i, 4), kk1 - 40, 40) & Left(Cells(i, 2), kkf)

If h = 999 And Len(Cells(i, 2)) < kkf Then ss = Mid(Cells(i, 4), kk1 - 40, 40) & Cells(i, 2) & Mid(Cells(i, 4), kk3 + 1, kkf - Len(Cells(i, 2)))

Do While Len(Cells(i, 8)) = 0 Or Cells(i, 8) >= 56.6

If Cells(i, 8) >= 56.6 Then k = k - 1

sF = Right(ss, k)

n = Len(sF)

If n < 18 And Cells(i, 8) <= 59 And Cells(i, 8) > 0 Then sF = Right(ss, k + 1)

If k < 18 Then Exit Do

ka = 0

kt = 0

kg = 0

kc = 0

Do While n > 0

If UCase(Mid(sF, n, 1)) = "A" Then ka = ka + 1

If UCase(Mid(sF, n, 1)) = "T" Then kt = kt + 1

If UCase(Mid(sF, n, 1)) = "C" Then kc = kc + 1

If UCase(Mid(sF, n, 1)) = "G" Then kg = kg + 1

n = n - 1

Loop

If ka + kt + kc + kg < 18 Then Exit Do

```

j = 0
If Len(sF) = ka + kt + kg + kc And (kg + kc) / (ka + kt + kg + kc) <= 0.72 And (kg + kc) / (ka +
kt + kg + kc) >= 0.16 Then j = 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc)
If j >= 56.6 And j <= 59 Then Cells(i, 8) = j

```

```

Do While Cells(i, 8) > 0
If InStr(1, sF, "AAAAAA") <> 0 Or InStr(1, sF, "TTTTTT") <> 0 Or InStr(1, sF, "CCCCCC")
<> 0 Or InStr(1, sF, "GGGGGG") <> 0 Then Cells(i, 8) = ""
Exit Do
Loop

```

```

If Cells(i, 8) > 0 And Cells(i, 8) <= 57.8 Then Exit Do
If Cells(i, 8) > 0 And Len(sF) = ka + kt + kg + kc And 72 + (37.4 * (kg + kc) - 747) / (ka + kt +
kg + kc - 1) < 56.1 Then sF = Right(ss, k + 1)
If Cells(i, 8) > 0 And 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc - 1) < 56.1 Then Exit Do

```

```

If Len(Cells(i, 8)) = 0 Then k = k - 1
Loop

```

```

If Cells(i, 8) > 0 And Cells(i, 8) <= 59 Then Cells(i, 5) = sF
If Len(Cells(i, 8)) = 0 Or Cells(i, 8) > 64.6 Then Cells(i, 5) = "Failed"
Cells(i, 8) = ""
sR = ""
sF = ""
ss = ""
j = 2

```

Designing second specific primer

```

Do While Cells(i, 5) <> "Failed"
k = 38
If m2 = 0 Or m2 = 3 Then k = Len(Cells(i, 5)) + m2
ss = Mid(Cells(i, 4), kk1 - 40, 40) & Cells(i, 3)
If h = 999 And Len(Cells(i, 3)) <= 38 And Len(Cells(i, 3)) >= kkf Then ss = Mid(Cells(i, 4), kk1 -
40, 40) & Left(Cells(i, 3), kkf)
If h = 999 And Len(Cells(i, 3)) < kkf Then ss = Mid(Cells(i, 4), kk1 - 40, 40) & Cells(i, 3) &
Mid(Cells(i, 4), kk3 + 1, kkf - Len(Cells(i, 3)))

```

```

Do While Len(Cells(i, 9)) = 0 Or Cells(i, 9) >= 56.6
If Cells(i, 9) >= 56.6 Then k = k - 1
sF = Right(ss, k)
n = Len(sF)
If n < 18 And Cells(i, 9) <= 59 And Cells(i, 9) > 0 Then sF = Right(ss, k + 1)
If k < 18 Then Exit Do

```

```

ka = 0
kt = 0
kg = 0
kc = 0
Do While n > 0
  If UCase(Mid(sF, n, 1)) = "A" Then ka = ka + 1
  If UCase(Mid(sF, n, 1)) = "T" Then kt = kt + 1
  If UCase(Mid(sF, n, 1)) = "C" Then kc = kc + 1
  If UCase(Mid(sF, n, 1)) = "G" Then kg = kg + 1
  n = n - 1
Loop
  If ka + kt + kc + kg < 18 Then Exit Do
  j = 0
  If Len(sF) = ka + kt + kg + kc And (kg + kc) / (ka + kt + kg + kc) <= 0.72 And (kg + kc) / (ka +
kt + kg + kc) >= 0.16 Then j = 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc)
  If j >= 56.6 And j <= 59 Then Cells(i, 9) = j

  Do While Cells(i, 9) > 0
    If InStr(1, sF, "AAAAAA") <> 0 Or InStr(1, sF, "TTTTTT") <> 0 Or InStr(1, sF, "CCCCCC")
<> 0 Or InStr(1, sF, "GGGGGG") <> 0 Then Cells(i, 9) = ""
    Exit Do
  Loop

  If Cells(i, 9) > 0 And Cells(i, 9) <= 63.3 Then Exit Do
  If Cells(i, 9) > 0 And Len(sF) = ka + kt + kg + kc And 72 + (37.4 * (kg + kc) - 747) / (ka + kt +
kg + kc - 1) < 56.1 Then sF = Right(ss, k + 1)
  If Cells(i, 9) > 0 And 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc - 1) < 56.1 Then Exit Do

  If Len(Cells(i, 9)) = 0 Then k = k - 1

Loop

If Cells(i, 9) > 0 And Cells(i, 9) <= 59 Then Cells(i, 6) = sF
If Len(Cells(i, 9)) = 0 Or Cells(i, 9) > 59 Then Cells(i, 5) = "Failed"

Exit Do
Loop
Cells(i, 9) = ""
sR = ""
sF = ""
ss = ""

jj = 4

```

Do While Cells(i, 5) <> "Failed"

Designing common primer

Do While jj <= 30

k = 40

sF = Mid(Cells(i, 4), kk3 + jj, k)

j = 2

Do While Len(Cells(i, 10)) = 0 Or Cells(i, 10) >= 59.9

If Cells(i, 10) >= 59.9 Then k = k - 1

sF = Left(sF, k)

n = Len(sF)

If n < 20 And Cells(i, 10) >= 59.9 Then sF = Mid(Cells(i, 4), kk3 + jj, k + 1)

If n < 20 Then Exit Do

ka = 0

kt = 0

kg = 0

kc = 0

Do While n > 0

If UCase(Mid(sF, n, 1)) = "A" Then ka = ka + 1

If UCase(Mid(sF, n, 1)) = "T" Then kt = kt + 1

If UCase(Mid(sF, n, 1)) = "C" Then kc = kc + 1

If UCase(Mid(sF, n, 1)) = "G" Then kg = kg + 1

n = n - 1

Loop

If ka + kt + kc + kg < 20 Then Exit Do

j = 0

If Len(sF) = ka + kt + kg + kc And (kg + kc) / (ka + kt + kg + kc) <= 0.7 And (kg + kc) / (ka + kt + kg + kc) >= 0.21 Then j = 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc)

If j >= 59.9 And j <= 62.9 Then Cells(i, 10) = j

Do While Abs(Cells(i, 10) - 60.4) <= 0.57

If InStr(1, sF, "AAAAAA") <> 0 Or InStr(1, sF, "TTTTTT") <> 0 Or InStr(1, sF, "CCCCCC") <> 0 Or InStr(1, sF, "GGGGGG") <> 0 Then Cells(i, 10) = ""

If Left(sF, 4) = "AAAA" Or Left(sF, 4) = "TTTT" <> 0 Or Left(sF, 4) = "CCCC" <> 0 Or Left(sF, 4) = "GGGG" <> 0 Then Cells(i, 10) = ""

If Right(sF, 4) = "AAAA" Or Right(sF, 4) = "TTTT" <> 0 Or Right(sF, 4) = "CCCC" <> 0 Or Right(sF, 4) = "GGGG" <> 0 Then Cells(i, 10) = ""

Exit Do

Loop

If Cells(i, 10) >= 59.9 And Len(sF) = ka + kt + kg + kc And 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc - 1) < 59.34 Then sF = Mid(Cells(i, 4), kk3 + jj, k + 1)

If Cells(i, 10) >= 59.9 And 72 + (37.4 * (kg + kc) - 747) / (ka + kt + kg + kc - 1) < 59.34 Then Exit Do

If Len(Cells(i, 10)) = 0 Then k = k - 1

Loop

If Len(Cells(i, 7)) = 0 And Abs(Cells(i, 10) - 60.4) <= 0.57 Then Cells(i, 7) = sF

Do While Abs(Cells(i, 10) - 60.4) <= 0.57 And Len(Cells(i, 7)) <> 0

If Left(sF, 1) <> "A" And Left(sF, 1) <> "T" And Abs(Len(sF) - 29.5) <= 4.5 Then Cells(i, 7) = sF

If Left(Cells(i, 7), 1) <> "G" And Left(Cells(i, 7), 1) <> "C" And Abs(Len(sF) - 29.5) <= 4.5 Then Cells(i, 7) = sF

If Cells(i, 7) <> sF And Abs(Len(Cells(i, 7)) - 29) > 5 And Abs(Len(sF) - 29) <= Abs(Len(Cells(i, 7)) - 29) Then Cells(i, 7) = sF

Exit Do

Loop

jj = jj + 1

If jj >= 10 And Left(Cells(i, 7), 1) <> "A" And Left(Cells(i, 7), 1) <> "T" And Len(Cells(i, 7)) <> 0 Then Exit Do

If jj >= 20 And Len(Cells(i, 7)) <> 0 Then Exit Do

Cells(i, 10) = ""

Loop

If Len(Cells(i, 7)) = 0 Then Cells(i, 5) = "Failed"

If Cells(i, 5) = "Failed" Then Cells(i, 10) = ""

If Len(Cells(i, 7)) <> 0 Then sF = Cells(i, 7)

If Len(Cells(i, 7)) = 0 Then sF = ""

If Len(Cells(i, 7)) <> 0 Then Cells(i, 7) = ""

n = Len(sF)

Do While n > 0

Sequence reverse complementation

If UCase(Mid(sF, n, 1)) = "A" Then Cells(i, 7) = Cells(i, 7) & "T"

If UCase(Mid(sF, n, 1)) = "T" Then Cells(i, 7) = Cells(i, 7) & "A"

If UCase(Mid(sF, n, 1)) = "C" Then Cells(i, 7) = Cells(i, 7) & "G"

If UCase(Mid(sF, n, 1)) = "G" Then Cells(i, 7) = Cells(i, 7) & "C"

n = n - 1

Loop

Exit Do

Loop

Exit Do
Loop

'Output primer sequence

Cells(i, 8) = ""
Cells(i, 9) = ""
Cells(i, 10) = ""

If q2 > 0 Then Cells(i, 4) = Left(Cells(i, 4), Len(Cells(i, 4)) - q2)
If q1 > 0 Then Cells(i, 4) = Right(Cells(i, 4), Len(Cells(i, 4)) - q1)
q1 = 0
q2 = 0
If Right(Cells(i, 5), 5) = Right(Cells(i, 6), 5) Then Cells(i, 5) = "Failed"
Do While Cells(i, 5) = "Failed" Or Len(Cells(i, 5)) = 0 Or Len(Cells(i, 6)) = 0 Or Cells(i, 7) = ""
 Cells(i, 5) = "Failed"
 Cells(i, 6) = ""
 Cells(i, 7) = ""
 h2 = h2 + 1
Exit Do
Loop

Do While Cells(i, 5) <> "Failed"
If Left(Cells(i, 5), 5) = "TTTTT" Then Cells(i, 5) = Mid(Cells(i, 5), 2, 38)
If Left(Cells(i, 6), 4) = "TTTT" Then Cells(i, 6) = Mid(Cells(i, 6), 2, 38)
If Left(Cells(i, 6), 4) = "TTTT" Then Cells(i, 6) = Mid(Cells(i, 6), 2, 38)
Cells(i, 5) = "GAAGGTGACCAAGTTCATGCT" & Cells(i, 5) 'Add 21bp adapter sequence
Cells(i, 6) = "GAAGGTCGGAGTCAACGGATT" & Cells(i, 6)
Cells(i, 5).Select
With Selection.Font
 .ThemeColor = xlThemeColorLight1

End With
With ActiveCell.Characters(Start:=1, Length:=21).Font
 .Color = -65536
End With
Exit Do
Loop

Do While Len(Cells(i, 6)) > 38
Cells(i, 6).Select
With Selection.Font
 .ThemeColor = xlThemeColorLight1

```
End With
With ActiveCell.Characters(Start:=1, Length:=21).Font
    .Color = -16776961
End With
Exit Do
Loop

i = i + 1
If i Mod 5 = 0 Then Application.ScreenUpdating = True
Loop
End Sub
```