Sub eCPX\_Sequencing()

'

' Macro for Translating and Analyzing Peptide Sequences from the eCPX 3.0 Bacterial Display Library

'This macro inputs all sequence files in a folder and searches each sequence for the random peptide based on the codons known to immediately precede and follow it.

'It compiles these peptides in a table on the open workbook for further analysis.

'The translation between the genetic code and the amino acids is adapted from methods developed at:http://www.bioc.uzh.ch/plueckthun/antibody/Macros/index.html

'

'Justin Jahnke & Deborah Sarkes, Jan. 2017

'This section puts in the default search sequences

Dim beg(100) As String

Dim fin(100) As String

Dim seqInput As String

Dim inCounter As Integer

inCounter = 91

'inputs default beginning sequence(s) at end (“GGCCAGTCTGGCCAG" or "ADD YOUR SEQUENCE")

beg(inCounter) = "GGCCAGTCTGGCCAG"

inCounter = inCounter + 1

beg(inCounter) = "GGCCAGTCTGGCCAG"

inCounter = inCounter + 1

beg(inCounter) = "GGCCAGTCTGGCCAG"

inCounter = 91

'inputs default final sequence(s) at end ("GGCTCGAGC" or "ADD YOUR SEQUENCE").

fin(inCounter) = "GGCTCGAGC"

inCounter = inCounter + 1

'number of codons taken by default, including beginning and end

Dim ndef As Integer

ndef = 150

'puts in genetic code

Dim Codons(64) As String

Dim Amino(64) As String

Codons(1) = "TTT"

Codons(2) = "TTC"

Codons(3) = "TTA"

Codons(4) = "TTG"

Codons(5) = "CTT"

Codons(6) = "CTC"

Codons(7) = "CTA"

Codons(8) = "CTG"

Codons(9) = "ATT"

Codons(10) = "ATC"

Codons(11) = "ATA"

Codons(12) = "ATG"

Codons(13) = "GTT"

Codons(14) = "GTC"

Codons(15) = "GTA"

Codons(16) = "GTG"

Codons(17) = "TCT"

Codons(18) = "TCC"

Codons(19) = "TCA"

Codons(20) = "TCG"

Codons(21) = "CCT"

Codons(22) = "CCC"

Codons(23) = "CCA"

Codons(24) = "CCG"

Codons(25) = "ACT"

Codons(26) = "ACC"

Codons(27) = "ACA"

Codons(28) = "ACG"

Codons(29) = "GCT"

Codons(30) = "GCC"

Codons(31) = "GCA"

Codons(32) = "GCG"

Codons(33) = "TAT"

Codons(34) = "TAC"

Codons(35) = "TAA"

Codons(36) = "TAG"

Codons(37) = "CAT"

Codons(38) = "CAC"

Codons(39) = "CAA"

Codons(40) = "CAG"

Codons(41) = "AAT"

Codons(42) = "AAC"

Codons(43) = "AAA"

Codons(44) = "AAG"

Codons(45) = "GAT"

Codons(46) = "GAC"

Codons(47) = "GAA"

Codons(48) = "GAG"

Codons(49) = "TGT"

Codons(50) = "TGC"

Codons(51) = "TGA"

Codons(52) = "TGG"

Codons(53) = "CGT"

Codons(54) = "CGC"

Codons(55) = "CGA"

Codons(56) = "CGG"

Codons(57) = "AGT"

Codons(58) = "AGC"

Codons(59) = "AGA"

Codons(60) = "AGG"

Codons(61) = "GGT"

Codons(62) = "GGC"

Codons(63) = "GGA"

Codons(64) = "GGG"

Amino(1) = "F"

Amino(2) = "F"

Amino(3) = "L"

Amino(4) = "L"

Amino(5) = "L"

Amino(6) = "L"

Amino(7) = "L"

Amino(8) = "L"

Amino(9) = "I"

Amino(10) = "I"

Amino(11) = "I"

Amino(12) = "M"

Amino(13) = "V"

Amino(14) = "V"

Amino(15) = "V"

Amino(16) = "V"

Amino(17) = "S"

Amino(18) = "S"

Amino(19) = "S"

Amino(20) = "S"

Amino(21) = "P"

Amino(22) = "P"

Amino(23) = "P"

Amino(24) = "P"

Amino(25) = "T"

Amino(26) = "T"

Amino(27) = "T"

Amino(28) = "T"

Amino(29) = "A"

Amino(30) = "A"

Amino(31) = "A"

Amino(32) = "A"

Amino(33) = "Y"

Amino(34) = "Y"

Amino(35) = "\*"

Amino(36) = "\*"

Amino(37) = "H"

Amino(38) = "H"

Amino(39) = "Q"

Amino(40) = "Q"

Amino(41) = "N"

Amino(42) = "N"

Amino(43) = "K"

Amino(44) = "K"

Amino(45) = "D"

Amino(46) = "D"

Amino(47) = "E"

Amino(48) = "E"

Amino(49) = "C"

Amino(50) = "C"

Amino(51) = "\*"

Amino(52) = "W"

Amino(53) = "R"

Amino(54) = "R"

Amino(55) = "R"

Amino(56) = "R"

Amino(57) = "S"

Amino(58) = "S"

Amino(59) = "R"

Amino(60) = "R"

Amino(61) = "G"

Amino(62) = "G"

Amino(63) = "G"

Amino(64) = "G"

Dim AAcids(21) As String

AAcids(1) = "A"

AAcids(2) = "C"

AAcids(3) = "D"

AAcids(4) = "E"

AAcids(5) = "F"

AAcids(6) = "G"

AAcids(7) = "H"

AAcids(8) = "I"

AAcids(9) = "K"

AAcids(10) = "L"

AAcids(11) = "M"

AAcids(12) = "N"

AAcids(13) = "P"

AAcids(14) = "Q"

AAcids(15) = "R"

AAcids(16) = "S"

AAcids(17) = "T"

AAcids(18) = "V"

AAcids(19) = "W"

AAcids(20) = "Y"

AAcids(21) = "X"

Dim B As Integer

B = 91

'This section inputs any sequences to be searched for just once (taken from column A and B of the open worksheet)

inCounter = 1

seqInput = Cells(inCounter, 1)

Do While seqInput <> "" And inCounter < 90

 beg(inCounter) = seqInput

 inCounter = inCounter + 1

 seqInput = Cells(inCounter, 1)

Loop

Do While B < 100

beg(inCounter) = beg(B)

inCounter = inCounter + 1

B = B + 1

Loop

inCounter = 1

seqInput = Cells(inCounter, 2)

Do While seqInput <> "" And inCounter < 90

 fin(inCounter) = seqInput

 inCounter = inCounter + 1

 seqInput = Cells(inCounter, 2)

Loop

B = 91

Do While B < 100

fin(inCounter) = fin(B)

inCounter = inCounter + 1

B = B + 1

Loop

B = 0

'This section gathers all the file names in the desired folder to be sorted.

 Dim fold As String 'fold is the string for the file directory to be used with the program

 fold = InputBox("enter directory") 'asks for the file directory to for the program to run with

 Dim fName(1000) As String

 Dim seqList(1000) As String 'seqList is an array of file names from the fold directory. The array length is currently limited to 1000, but this can be increased

 ' The next section is based on code found online and r

 ' Sets up the variable "MyFile" to be each file in the directory

 ' Test for Windows or Macintosh platform. Make the directory request.

 Sep = Application.PathSeparator

 If Sep = "\" Then

 ' Windows platform search syntax. Takes all the files names in the 'fold' directory and puts them in 'MyFile'

 MyFile = Dir(fold & Sep & "\*.\*")

 Else

 ' Macintosh platform search syntax. Never tested

 MyFile = Dir(fold, MacID("seq"))

 End If

 ' Starts the loop, which will continue until there are no more files found.

 Dim d As Integer 'counter variable used in loop below and elsewhere in the code

 d = 1

 Do While MyFile <> "" 'Loop ends when MyFile is empty

 'These next lines write the file names from MyFile to seqList

 seqList(d) = fold & Sep & MyFile

 fName(d) = MyFile

 MyFile = Dir()

 d = d + 1

 Loop

 ActiveSheet.Name = "AA sequences" 'Creates a new worksheet 'AA sequences' that will have the sequence table

 ActiveWorkbook.Worksheets.Add

 ActiveSheet.Name = "test"

na = 1 'counter for big for loop

ro = 2 'ro is a counter for the row location

co = 1 'co is a counter for the column location

'This for loop is the heart of the program. It takes each seq file one at a time and does the following with it:

'imports it into excel,

'processes the dna sequence to find the random peptide part,

'converts the dna sequence into an amino acid sequence

For na = 1 To 1000 Step 1

 'Exits loop if next file name in sequence list is empty

 If seqList(na) = "" Then

 Exit For

 Else

 Range("A1").Select

 'Imports next sequence file; developed by recording a macro of importing a text file into excel

 With ActiveSheet.QueryTables.Add(Connection:= \_

 "TEXT;" & seqList(na) \_

 , Destination:=Range("$A$1"))

 .Name = "DNA"

 .FieldNames = True

 .RowNumbers = False

 .FillAdjacentFormulas = False

 .PreserveFormatting = True

 .RefreshOnFileOpen = False

 .RefreshStyle = xlInsertDeleteCells

 .SavePassword = False

 .SaveData = True

 .AdjustColumnWidth = True

 .RefreshPeriod = 0

 .TextFilePromptOnRefresh = False

 .TextFilePlatform = 437

 .TextFileStartRow = 1

 .TextFileParseType = xlDelimited

 .TextFileTextQualifier = xlTextQualifierDoubleQuote

 .TextFileConsecutiveDelimiter = False

 .TextFileTabDelimiter = True

 .TextFileSemicolonDelimiter = False

 .TextFileCommaDelimiter = False

 .TextFileSpaceDelimiter = False

 .TextFileColumnDataTypes = Array(1)

 .TextFileTrailingMinusNumbers = True

 .Refresh BackgroundQuery:=False

 End With

 ActiveSheet.Name = "DNA" 'name dna used to keep track of active sheet

 'These next lines process the dna sequence to find the random peptide portion.

 'concatenates entire dna sequence to a single cell. Only the first 18 cells are concatenated.

 Range("N2").Select

 ActiveCell.FormulaR1C1 = \_

 "=CONCATENATE(RC[-13],R[1]C[-13],R[2]C[-13],R[3]C[-13],R[4]C[-13],R[5]C[-13],R[6]C[-13],R[7]C[-13],R[8]C[-13],R[9]C[-13],R[10]C[-13],R[11]C[-13],R[12]C[-13],R[13]C[-13],R[14]C[-13],R[15]C[-13],R[16]C[-13],R[17]C[-13],R[18]C[-13])"

 'searches for the sequence immediately preceeding the random peptide

 inCounter = 1

 Range("N3").Select

 ActiveCell.FormulaR1C1 = "=SEARCH(""" & ZZZ & """,R[-1]C,1)" 'Ensures error message is displayed if no search sequences are inputed

 'searches for beginning sequences

 Do

 If beg(inCounter) <> "" Then

 ActiveCell.FormulaR1C1 = "=SEARCH(""" & beg(inCounter) & """,R[-1]C,1)"

 End If

 inCounter = inCounter + 1

 Loop While (IsError(ActiveCell) = True) And (inCounter < 100)

 Selection.Copy

 Range("n4").Select

 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

 :=False, Transpose:=False

 'If beginning sequence has not been found, the final sequence is used to search instead.

 inCounter = 1

 Range("N3").Select

 If IsError(ActiveCell) = True Then

 Do While (IsError(ActiveCell) = True) And (inCounter < 100)

 If fin(inCounter) <> "" Then

 ActiveCell.FormulaR1C1 = "=SEARCH(""" & fin(inCounter) & """,R[-1]C,1)"

 End If

 If IsError(ActiveCell) = True Then

 inCounter = inCounter + 1

 End If

 Loop

 Dim Dummy2 As Integer

 If IsError(ActiveCell) = False Then

 Dummy2 = ActiveCell

 Dummy2 = (Dummy2 - 60)

 ActiveCell.FormulaR1C1 = Dummy2

 Range("O3").Select

 ActiveCell.FormulaR1C1 = "5' ERROR"

 Else

 Range("O3").Select

 ActiveCell.FormulaR1C1 = "CHECK SEQ"

 End If

 End If

 'takes the random peptide sequence + fixed sequences preceding and following into from overall sequence

 Range("N4").Select

 ActiveCell.FormulaR1C1 = "=MID(R[-2]C,R[-1]C," & ndef & ")"

 'checks if dna sequence taken ends correctly BROKEN

 Range("N5").Select

 ActiveCell.FormulaR1C1 = "=SEARCH(""" & fin(inCounter) & """,R[-1]C,1)"

 If IsError(ActiveCell) = True Then

 inCounter = 1

 Do While (inCounter < 100)

 If fin(inCounter) <> "" Then

 ActiveCell.FormulaR1C1 = "=SEARCH(""" & fin(inCounter) & """,R[-1]C,1)"

 If IsError(ActiveCell) = False Then

 Range("N6").Select

 ActiveCell.FormulaR1C1 = "=MID(R[-2]C,16,(R[-1]C-16))"

 inCounter = 1000

 End If

 inCounter = inCounter + 1

 Else

 inCounter = 200

 End If

 Loop

 If inCounter <> 1001 Then

 Range("N6").Select

 ActiveCell.FormulaR1C1 = "=MID(R[-2]C,16,45)"

 Range("O6").Select

 ActiveCell.FormulaR1C1 = "3' ERROR"

 End If

 Else

 Range("N6").Select

 ActiveCell.FormulaR1C1 = "=MID(R[-2]C,16,(R[-1]C-16))"

 End If

 'this next section of the code copies the search data to a particular row of the sheet "AA sequences"

 Sheets("DNA").Select 'copies dna sequence

 Range("N2").Select

 Selection.Copy

 Sheets("AA sequences").Select

 Cells(ro, co).Select

 ActiveCell.FormulaR1C1 = na 'puts in overall sequence number

 co = co + 1

 Cells(ro, co).Select

 ActiveCell.FormulaR1C1 = fName(na) 'puts in file name

 co = co + 1

 'pastes dna sequence

 Cells(ro, co).Select

 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

 :=False, Transpose:=False

 co = co + 1

 'copies and paste random peptide dna sequence

 Sheets("DNA").Select

 Range("N6").Select

 Application.CutCopyMode = False

 Selection.Copy

 Sheets("AA sequences").Select

 Cells(ro, co).Select

 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

 :=False, Transpose:=False

 co = co + 1

 Sheets("DNA").Select

 Range("N5").Select

 Application.CutCopyMode = False

 Selection.Copy

 Sheets("AA sequences").Select

 Cells(ro, co).Select

 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

 :=False, Transpose:=False

 If ActiveCell.FormulaR1C1 <> "#VALUE!" Then

 Dim Dummy As Integer

 Dummy = ActiveCell

 Dummy = (Dummy - 16) / 3

 ActiveCell.FormulaR1C1 = Dummy

 End If

 Sheets("DNA").Select

 Range("O3").Select

 If ActiveCell.FormulaR1C1 = "5' ERROR" Then

 Sheets("AA sequences").Select

 Cells(ro, co).Select

 With Selection.Interior

 .ColorIndex = 6

 .Pattern = xlSolid

 End With

 ActiveCell.FormulaR1C1 = "5' ERROR"

 End If

 Sheets("DNA").Select

 Range("O6").Select

 If ActiveCell.FormulaR1C1 = "3' ERROR" Then

 Sheets("AA sequences").Select

 Cells(ro, co).Select

 With Selection.Interior

 .ColorIndex = 4

 .Pattern = xlSolid

 End With

 ActiveCell.FormulaR1C1 = "3' ERROR"

 End If

 Sheets("DNA").Select

 Range("O3").Select

 If ActiveCell.FormulaR1C1 = "CHECK SEQ" Then

 Sheets("AA sequences").Select

 Cells(ro, co).Select

 With Selection.Interior

 .ColorIndex = 3

 .Pattern = xlSolid

 End With

 ActiveCell.FormulaR1C1 = "CHECK SEQ"

 End If

 Sheets("AA sequences").Select

 Cells(ro, co).Select

 co = co + 1

 'copies dna sequence into new cell to be converted to peptide sequence

 Cells(ro, co - 2).Select

 Selection.Copy

 Cells(ro, co + 1).Select

 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

 :=False, Transpose:=False

'Code here is adapted from: http://www.bioc.uzh.ch/plueckthun/antibody/Macros/index.html

 i1 = Selection.Row

 i2 = i1 + Selection.Rows.Count - 1

 j = Selection.Column

 N = 0 'length of longest sequence

 For i = i1 To i2 Step 1 'for all selected cells

 AAA = Cells(i, j)

 If TypeName(AAA) = "String" Then

 A = Len(AAA)

 If B < A / 3 Then

 B = A / 3

 End If

 If (A / 3) > N Then N = (A / 3) 'determine length of longest sequence

 For k = 1 To (A / 3) Step 1 'parse the sequence

 Cells(i, j + k) = Mid(AAA, (3 \* k) - 2, 3)

 Next k

 Else

 'Error, is not a character string

 End If

 Next i

 Selection.Delete Shift:=xlToLeft 'Delete original strings

 Range(Cells(i1, j), Cells(i2, j + N)).Select 'select sequence alignment

 Selection.ColumnWidth = 4 'set column width to 3

 'These three loops look up the amino acid sequences based on the dna sequences

 i1 = Selection.Row

 i2 = i1 + Selection.Rows.Count - 1

 j1 = Selection.Column

 j2 = j1 + Selection.Columns.Count - 1

 'These three loops look up the amino acid sequences based on the dna sequences

 For i = i1 To i2 Step 1 'outermost loop-each step moves onto another sequence

 For j = j1 To j2 - 1 Step 1 'outer loop-each steps moves onto another codon

 'this section makes cell red for an error; when the codon is successfully translated the cell is converted back to white

 Cells(i, j).Select

 With Selection.Interior

 .ColorIndex = 3

 .Pattern = xlSolid

 End With

 For k = 1 To 64 Step 1 'inner loop-scans through codon table to find codon and then substitutes correct amino acid

 If Cells(i, j) = Codons(k) Then 'this line checks to see if the codon matches the current line (k) in the table

 Cells(i, j) = Amino(k) 'if it matches, the cell is replaced with the correct amino acid

 Cells(i, j).Select 'cell converted back to white background

 With Selection.Interior

 .ColorIndex = 0

 .Pattern = xlSolid

 End With

 If Cells(i, j) = "\*" Then

 Cells(i, j).Select

 With Selection.Interior

 .ColorIndex = 3

 .Pattern = xlSolid

 End With

 End If

 k = -1

 Exit For 'exits innermost loop; moves onto next codon (only executed if codon matched correctly)

 End If

 Next k

 If k <> -1 Then 'Turns improperly decoded nucleotides into X's

 Cells(i, j) = "X"

 End If

 Next j

 Next i

 'last step for each sequence is to concatentate the sequence for easier parsing later

 Cells(ro, co).Select

 ActiveCell.FormulaR1C1 = \_

 "=CONCATENATE(RC[1],RC[2],RC[3],RC[4],RC[5],RC[6],RC[7],RC[8],RC[9],RC[10],RC[11],RC[12],RC[13],RC[14],RC[15],RC[16],RC[17],RC[18],RC[19],RC[20],RC[21],RC[22],RC[23],RC[24]," & \_

 "RC[25],RC[26],RC[27],RC[28],RC[29],RC[30],RC[31],RC[32],RC[33],RC[34],RC[35],RC[36],RC[37],RC[38],RC[39],RC[40],RC[41],RC[42],RC[43],RC[44],RC[45],RC[46],RC[47],RC[48],RC[49],RC[50])"

 Cells(ro, (co - 1)).Select

 If ActiveCell.FormulaR1C1 = 0 Then

 Cells(ro, co).Select

 ActiveCell.FormulaR1C1 = "#EMPTY"

 End If

 co = 1

 ro = ro + 1

 Sheets("DNA").Select

End If 'This end if goes with If seqList(na) = "" Then at the beginning of the loop

Next na 'Moves onto the next sequence

'After all the sequences are parsed, labels are added to each of the columns and the widths of some columns increased to improve readability

Sheets("AA sequences").Select

 Range("A1").Select

 ActiveCell.FormulaR1C1 = "Seq #"

 Range("B1").Select

 ActiveCell.FormulaR1C1 = "File Name"

 Columns("C:C").Select

 Range("C1").Select

 Columns("B:B").EntireColumn.AutoFit

 Range("C1").Select

 ActiveCell.FormulaR1C1 = "Full Sequence"

 Range("D1").Select

 Columns("C:C").ColumnWidth = 12.71

 ActiveCell.FormulaR1C1 = "Random Peptide"

 Columns("E:E").Select

 Columns("D:D").EntireColumn.AutoFit

 Range("D1").Select

 ActiveCell.FormulaR1C1 = "Random Peptide DNA Sequence"

 Range("E1").Select

 ActiveCell.FormulaR1C1 = "Read OK?"

 Range("F1").Select

 ActiveCell.FormulaR1C1 = "AA Sequence"

 Range("G1").Select

 ActiveCell.FormulaR1C1 = ""

 Range("F1").Select

 Columns("F:F").EntireColumn.AutoFit

 Range("G1").Select

 ActiveCell.FormulaR1C1 = "AA Sequence Seperated"

 Range(Cells(1, 1), Cells(1, B + 6)).Select

 Range(Selection, Selection.End(xlDown)).Select

 Selection.Borders(xlDiagonalDown).LineStyle = xlNone

 Selection.Borders(xlDiagonalUp).LineStyle = xlNone

 With Selection.Borders(xlEdgeLeft)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 With Selection.Borders(xlEdgeTop)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 With Selection.Borders(xlEdgeBottom)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 With Selection.Borders(xlEdgeRight)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 With Selection.Borders(xlInsideVertical)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 With Selection.Borders(xlInsideHorizontal)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 Range("D6").Select

 Range(Cells(1, 7), Cells(1, B + 6)).Select

 With Selection

 .HorizontalAlignment = xlCenter

 .VerticalAlignment = xlBottom

 .WrapText = False

 .Orientation = 0

 .AddIndent = False

 .IndentLevel = 0

 .ShrinkToFit = False

 .ReadingOrder = xlContext

 .MergeCells = False

 End With

 Selection.Merge

 Range(Cells(1, 1), Cells(1, B + 6)).Select

 With Selection

 .HorizontalAlignment = xlCenter

 .VerticalAlignment = xlBottom

 .WrapText = False

 .Orientation = 0

 .AddIndent = False

 .IndentLevel = 0

 .ShrinkToFit = False

 .ReadingOrder = xlContext

 End With

 Range(Cells(1, 1), Cells(1, B + 6)).Select

 Selection.Font.Bold = True

 With Selection.Font

 .Name = "Calibri"

 .Size = 12

 .Strikethrough = False

 .Superscript = False

 .Subscript = False

 .OutlineFont = False

 .Shadow = False

 .Underline = xlUnderlineStyleNone

 .ThemeColor = xlThemeColorLight1

 .TintAndShade = 0

 .ThemeFont = xlThemeFontMinor

 End With

 Columns("C:C").ColumnWidth = 13.86

 Columns("E:E").EntireColumn.AutoFit

'Creates Summary Table on New Sheet

 Sheets("DNA").Select

 Range("A1:JJ1001").Select

 Selection.Delete

 ActiveSheet.Name = "Summary Table"

 Sheets("Summary Table").Select

 Sheets("AA sequences").Select

 ActiveWindow.SmallScroll Down:=-15

 Range("A1:AZ1001").Select

 Selection.Copy

 Sheets("Summary Table").Select

 Range("A2").Select

 Sheets("Summary Table").Select

 Range("A1").Select

 ActiveSheet.Paste

 Range("B1:C1001").Select

 Application.CutCopyMode = False

 Application.CutCopyMode = False

 Selection.EntireColumn.Delete

 Range("C1:C1001").Select

 Selection.Delete Shift:=xlToLeft

 Range("D1:R1001").Select

 Columns("B:B").EntireColumn.AutoFit

 Columns("C:C").EntireColumn.AutoFit

 Range("A2:AZ1001").Select

 With Selection

 .HorizontalAlignment = xlCenter

 .VerticalAlignment = xlBottom

 .WrapText = False

 .Orientation = 0

 .AddIndent = False

 .IndentLevel = 0

 .ShrinkToFit = False

 .ReadingOrder = xlContext

 .MergeCells = False

 End With

 Range("B10").Select

 Columns("A:AZ").EntireColumn.AutoFit

'Creates FASTA sheet

 Sheets("AA sequences").Select

 Sheets.Add After:=ActiveSheet

 ActiveSheet.Name = "FASTA"

 d = 1

 j = 100

 Do While (d < na)

 Sheets("FASTA").Select

 ro = d \* 2 - 1

 Cells(ro, 1).Select

 ActiveCell.FormulaR1C1 = ">" & d

 ro = ro + 1

 Sheets("AA sequences").Select

 Cells(d + 1, 6).Select

 Selection.Copy

 Sheets("FASTA").Select

 Cells(ro, 1).Select

 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

 :=False, Transpose:=False

 If (Cells(ro, 1) = "") Then

 Cells(ro, 1) = "#VALUE!"

 End If

 d = d + 1

 Loop

'Sorts Summary Table and copies sorted Table to FASTA

 Sheets("Summary Table").Select

 Range("A2:AZ1001").Select

 ActiveWorkbook.Worksheets("Summary Table").sort.SortFields.Clear

 ActiveWorkbook.Worksheets("Summary Table").sort.SortFields.Add Key:=Range( \_

 "C2:C1001"), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= \_

 xlSortNormal

 With ActiveWorkbook.Worksheets("Summary Table").sort

 .SetRange Range("A2:AZ1001")

 .Header = xlGuess

 .MatchCase = False

 .Orientation = xlTopToBottom

 .SortMethod = xlPinYin

 .Apply

 End With

 d = 1

 j = 100

 Do While (d < na)

 Sheets("Summary Table").Select

 j = Cells(d + 1, 1)

 Sheets("FASTA").Select

 ro = d \* 2 - 1

 Cells(ro, 2).Select

 ActiveCell.FormulaR1C1 = ">" & j

 ro = ro + 1

 Sheets("Summary Table").Select

 Cells(d + 1, 3).Select

 Selection.Copy

 Sheets("FASTA").Select

 Cells(ro, 2).Select

 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

 :=False, Transpose:=False

 If (Cells(ro, 2) = "") Then

 Cells(ro, 2) = "#VALUE!"

 End If

 d = d + 1

 Loop

 Sheets("FASTA").Select

 Columns("A:A").EntireColumn.AutoFit

 Columns("B:B").EntireColumn.AutoFit

'Creates Freq Table Sheet

 Sheets.Add After:=ActiveSheet

 ActiveSheet.Name = "Freq Table"

 Range("A1").Select

 ActiveCell.FormulaR1C1 = "Seq #"

 Range("B1").Select

 ActiveCell.FormulaR1C1 = "AA Sequence"

 Columns("B:B").EntireColumn.AutoFit

 Range("A1:W" & na).Select

 Selection.Borders(xlDiagonalDown).LineStyle = xlNone

 Selection.Borders(xlDiagonalUp).LineStyle = xlNone

 With Selection.Borders(xlEdgeLeft)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 With Selection.Borders(xlEdgeTop)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 With Selection.Borders(xlEdgeBottom)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 With Selection.Borders(xlEdgeRight)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 With Selection.Borders(xlInsideVertical)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 With Selection.Borders(xlInsideHorizontal)

 .LineStyle = xlContinuous

 .ColorIndex = 0

 .TintAndShade = 0

 .Weight = xlThin

 End With

 d = 3

 Do While (d < 24)

 Cells(1, d).Select

 ActiveCell.Formula = AAcids(d - 2)

 d = d + 1

 Loop

'Counts Frequency for different amino acids

 Dim Freq(21) As Integer

 d = 2

 i = 7

 j = 1

 Do While (d < na + 1)

 Sheets("AA sequences").Select

 Cells(d, 6).Copy

 Sheets("Freq Table").Select

 Cells(d, 2).Select

 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

 :=False, Transpose:=False

 Cells(d, 1) = d - 1

 j = 1

 Sheets("AA sequences").Select

 Do While (j < 22)

 Freq(j) = 0

 i = 7

 Do While (i < 107)

 If (Cells(d, i) = AAcids(j)) Then

 Freq(j) = Freq(j) + 1

 End If

 i = i + 1

 Loop

 j = j + 1

 Loop

 Sheets("Freq Table").Select

 j = 1

 Do While (j < 22)

 Cells(d, j + 2) = Freq(j)

 j = j + 1

 Loop

 d = d + 1

 Loop

 Columns("B:AZ").EntireColumn.AutoFit

 Range("a1:z1").Select

 Selection.Font.Bold = True

'Creates sheet with search parameters

 Sheets("Freq Table").Select

 Sheets.Add After:=ActiveSheet

 ActiveSheet.Name = "Search Parameters"

 Cells(1, 1) = "5'"

 Cells(1, 2) = "3'"

 d = 1

 Do While (d < 100)

 Cells(d + 1, 1) = beg(d)

 Cells(d + 1, 2) = fin(d)

 d = d + 1

 Loop

 Columns("A:A").EntireColumn.AutoFit

 Columns("B:B").EntireColumn.AutoFit

'Looks for repeats on Summary Table

 Sheets("Summary Table").Select

 d = 1

 j = 2

 Do While (d <= na)

 If Cells(d + 1, 3) <> "" Then

 If Cells(d + 1, 3) = Cells(d, 3) Then

 Range(Cells(d, 3), Cells(d + 1, 3)).Select

 With Selection.Borders(xlEdgeTop)

 .LineStyle = xlContinuous

 .ColorIndex = j

 .TintAndShade = 0

 .Weight = xlThick

 End With

 With Selection.Borders(xlEdgeBottom)

 .LineStyle = xlContinuous

 .ColorIndex = j

 .TintAndShade = 0

 .Weight = xlThick

 End With

 With Selection.Borders(xlEdgeRight)

 .LineStyle = xlContinuous

 .ColorIndex = j

 .TintAndShade = 0

 .Weight = xlThick

 End With

 With Selection.Borders(xlEdgeLeft)

 .LineStyle = xlContinuous

 .ColorIndex = j

 .TintAndShade = 0

 .Weight = xlThick

 End With

 With Selection.Borders(xlInsideVertical)

 .LineStyle = xlContinuous

 .ColorIndex = j

 .TintAndShade = 0

 .Weight = xlThick

 End With

 With Selection.Borders(xlInsideHorizontal)

 .LineStyle = xlContinuous

 .ColorIndex = j

 .TintAndShade = 0

 .Weight = xlThick

 End With

 Else

 j = j + 1

 If j > 72 Then

 j = 3

 End If

 End If

 End If

 d = d + 1

 Loop

Cells(1, 1).Select 'Selects A1 on Summary Table

End Sub