1. **Appendix**

Appendix figure 1

|  |  |  |  |
| --- | --- | --- | --- |
| 2D-LC | routinely quantfiable | 1D-LC (offline desalt) | 1D-LC (online desalt) |
| TKQTAR | \* | TKQTAR | KSTGGKAPR |
| TKme1QTAR | \* | TKme1QTAR | Kme1STGGKAPR |
| KSTGGKAPR |  | KSTGGKAPR | Kme2STGGKAPR |
| Kme1STGGKAPR |  | Kme1STGGKAPR | Kme3STGGKAPR |
| Kme2STGGKAPR |  | Kme2STGGKAPR | KacSTGGKAPR |
| Kme3STGGKAPR |  | Kme3STGGKAPR | KSTGGKacAPR |
| KacSTGGKAPR |  | KacSTGGKAPR | Kme1STGGKacAPR |
| KSTGGKacAPR |  | KSTGGKacAPR | Kme2STGGKacAPR |
| Kme1STGGKacAPR |  | Kme1STGGKacAPR | Kme3STGGKacAPR |
| Kme2STGGKacAPR |  | Kme2STGGKacAPR | KQLATKAAR |
| Kme3STGGKacAPR |  | Kme3STGGKacAPR | Kme1QLATKAAR |
| KacSTGGKacAPR | \*\* | KQLATKAAR | KacQLATKAAR |
| KSphTGGKAPR | \* | Kme1QLATKAAR | KQLATKacAAR |
| Kme2SphTGGKAPR | \* | KacQLATKAAR | KacQLATKacAAR |
| KQLATKAAR |  | KQLATKacAAR | Kme2SAPATGGVKKPHR |
| KQLATphKAAR | \* | KacQLATKacAAR | Kme3SAPATGGVKKPHR |
| Kme1QLATKAAR |  | KSAPATGGVKKPHR | Kme2SAPATGGVKme1KPHR |
| KacQLATKAAR |  | Kme2SAPATGGVKKPHR | Kme2SAPATGGVKme2KPHR |
| KQLATKacAAR |  | Kme3SAPATGGVKKPHR | YRPGTVALR |
| KacQLATKacAAR |  | Kme2SAPATGGVKme1KPHR | KLFPQR |
| KSAPATGGVKKPHR |  | Kme2SAPATGGVKme2KPHR | GKGGKGLGKGGAKR |
| Kme1SAPATGGVKKPHR |  | Kme3SAPATGGVKme1KPHR | GKGGKGLGKGGAKacR |
| Kme2SAPATGGVKKPHR |  | YRPGTVALR | GKGGKacGLGKGGAKacR |
| Kme3SAPATGGVKKPHR |  | YQKSTELLIR | GKGGKacGLGKacGGAKR |
| KSAPATGGVKme2KPHR | \*\* | EIAQDFKTDLR | DNIQGITKPAIR |
| Kme1SAPATGGVKme1KPHR |  | GKGGKGLGKGGAKR | GVLKVFLENVIR |
| Kme1SAPATGGVKme2KPHR |  | GKGGKGLGKGGAKacR | AKAKTR |
| Kme1SAPATGGVKme3KPHR |  | GKGGKacGLGKGGAKacR | HLQLAIR |
| Kme2SAPATGGVKme1KPHR |  | GKGGKGLGKacGGAKacR | KGHYAER |
| Kme2SAPATGGVKme2KPHR |  | GKGGKacGLGKacGGAKR | AGLQFPVGR |
| Kme2SAPATGGVKme3KPHR |  | DNIQGITKPAIR |  |
| Kme3SAPATGGVKme1KPHR |  | AKAKSR |  |
| KSphAPATGGVKKPHR | \* | AKAKTR |  |
| Kme2SphAPATGGVKKPHR | \* | HLQLAIR |  |
| YRPGTVALR |  | DNKKTR |  |
| YQKSTELLIR |  | GKTGGKAR |  |
| YEKSTELLIR |  | KGHYAER |  |
| KLFPQR |  | GKacQGGKAR |  |
| EIAQDFKTDLR |  | AGLQFPVGR |  |
| EIAQDFKme1TDLR | \* | GKQGGKAR |  |
| EIAQDFKme2TDLR |  | KSAGAAKR |  |
| KSAPSTGGVKKPHR | \*\* | KmeSAGAAKR |  |
| Kme1APSTGGVKKPHR |  |  |  |
| Kme2SAPSTGGVKKPHR |  |  |  |
| Kme3SAPSTGGVKKPHR |  |  |  |
| Kme1SAPSTGGVKme1KPHR |  |  |  |
| Kme1SAPSTGGVKme2KPHR |  |  |  |
| Kme1SAPSTGGVKme3KPHR |  |  |  |
| Kme2SAPSTGGVKme1KPHR |  |  |  |
| Kme2SAPSTGGVKme2KPHR |  |  |  |
| Kme2SAPSTGGVKme3KPHR |  |  |  |
| Kme3SAPSTGGVKme1KPHR |  |  |  |
| GKGGKGLGKGGAKR |  |  |  |
| GKGGKacGLGKGGAKR |  |  |  |
| GKGGKGLGKacGGAKR |  |  |  |
| GKGGKGLGKGGAKacR |  |  |  |
| GKGGKacGLGKGGAKacR |  |  |  |
| GKGGKGLGKacGGAKacR |  |  |  |
| GKacGGKacGLGKacGGAKacR |  |  |  |
| Kme2VLRDNIQGITKPAIR |  |  |  |
| DNIQGITKPAIR |  |  |  |
| DNIQGITphKPAIR | \* |  |  |
| DAVTYTEHAKR |  |  |  |
| AKAKSR |  |  |  |
| AKAKTR |  |  |  |
| HLQLAIR |  |  |  |
| DNKKTR |  |  |  |
| GKTGGKAR |  |  |  |
| KGHYAER |  |  |  |
| GKmeQGGKAR |  |  |  |
| GKacQGGKAR |  |  |  |
| AGLQFPVGR |  |  |  |
| NDEELNKLLGR |  |  |  |
| AGGKAGKDSGKAKTKAVSR |  |  |  |
| AGGKAGKDSGKAKAKAVSR |  |  |  |
| GKQGGKAR |  |  |  |
| HLQLAVR |  |  |  |
| GGKKKSTKTSR |  |  |  |
| KSAGAAKR |  |  |  |
| KmeSAGAAKR |  |  |  |
| SETAPAAPAAPAPAEKTPVKKKAR |  |  |  |
| SETAPAETATPAPVEKSPAKKKATKKAAGAGAAKR |  |  |  |
| GAPAAATAPAPTAHKAKKAAPGAAGSR |  |  |  |
| STITSR |  |  |  |
| EIQTAVR |  |  |  |
| LAHYNKR |  |  |  |

**Supplementary Table 1. Corresponding peptides identified by 2D-LC and 1D-LC methods.** Histone peptides quantified from both T-cells and monocyte derived macrophages presented in this thesis utilising the 2D-LC methodology  reliably identified in both T-cells and macrophages, \* reliably quantified in MDMs that had been stimulated with LPS, \*\* Reliably quantified in T-cells not in MDMs,\* Reliably quantified in MDMs not in T-cells. This is compared against the histone peptides quantified by the two 1D-LC methodologies (online desalt, and offline desalt respectively. The coloured background represents a different histone protein. Blue = H3.1 H3.2 peptides, yellow= H3.3 peptides, orange H4=peptides, purple= H2A peptides, grey = H1 peptides, green=H2B peptides. In addition to this information regarding the presence.

\* reliably quantified in MDMs that had been stimulated with LPS, \*\* Reliably quantified in T-cells not in MDMs,\* Reliably quantified in MDMs not in T-cells.

* 1. **Histomatic script**

'To automate the DataAnalysis software you need to add a reference to the DataAnalysis library

'to do this go tool references and locate BDal DataAnalysis Object Library

Public MyApp As DataAnalysis.Application 'define the top level application object

Public MyAnalysis As DataAnalysis.Analysis 'define the analysis object

Public chrom As DataAnalysis.Chromatogram 'define the chrom object

Public MyEIC As DataAnalysis.Chromatogram 'define the eic chrom object

Public ExportFileWB As Excel.Workbook 'define a workbook for importing that data

Public ExportFileWS As Excel.Worksheet 'define the worksheet of the active sheet

Public MasterWB As Excel.Workbook 'define a workbook for the master workbook

Public AnalysisCell As Excel.Range 'df 22.12.2014

Public AreaFound

Public RangeFound

Public IntensityFound

Public SNFound

Public Sub MineAllRTs()

' loop all rows until RT is not populated

Set MasterWB = Excel.Application.ActiveWorkbook 'set this master spreadsheet to the master spreadsheet

'

Do Until (Excel.ActiveCell.Value = "") 'look until current RT cell is empty

'

If Excel.ActiveCell.Value = 0 Then 'if the current RT = 0 then skip it

'do nothing as its 0

Else

Call ImportExportData 'lookup the RT

End If

'

Excel.ActiveCell.Offset(1, 0).Select ' move to the next row'

'If Excel.ActiveCell.Offset(1, 0).Select = <1 Then Excel.ActiveCell.Value = False

'above code skips RT values of 0 when data mining'

Loop ' loop

'df 22.12.14 remove the msg box MsgBox "No More RTs to process"'

Excel.ActiveCell.Offset(1, -4).Select 'df 22.12.14 'now we are on the last row move down one more row the back to the file name column (1,-4)'

'this could allow for many analysis files on one spreadsheet with one blank row between each filename / data'

End Sub

Public Sub ImportExportData()

Set MasterWB = Excel.Application.ActiveWorkbook 'set this master spreadsheet to the master spreadsheet

Dim inRT

Dim inMoleculeSize

' df 18.12.14 set RT from array now element

inRT = ActiveCell.Value ' store the RT from the MAster Spreadsheet

inMoleculeSize = ActiveCell.Offset(0, -1).Value ' get the offset one cell back get the sample molecule size

Dim MyDataMoleculesArr() As String

MyDataMoleculesArr = Split(inMoleculeSize, "/", , vbTextCompare) 'create an array of the molecule size 1 element for every "/" char

inMoleculeSize = MyDataMoleculesArr(0) ' get the first molecule

'new code 18.12.14 split the RT values by slash

Dim MyRTArr() As String

MyRTArr = Split(inRT, "/", , vbTextCompare) 'create an array of the RT 1 element for every "/" char

'Dim WorkbookPath As String

Dim AreaFoundArr() As String

Dim RangeFoundArr() As String

Dim IntensityFoundArr() As String

Dim SNFoundArr() As String

ReDim AreaFoundArr(UBound(MyRTArr)) 'make the area array same size as rt array

ReDim RangeFoundArr(UBound(MyRTArr))

ReDim IntensityFoundArr(UBound(MyRTArr))

ReDim SNFoundArr(UBound(MyRTArr))

Dim WorkbookName As String

'WorkbookPath = "C:\\_TOM\"

'WorkbookName = "B23 TM SUR CONT EXP1\_BC2\_01\_6682.dCompoundList.csv"

'WorkbookName = "C:\\_TOM\CompoundList.csv"

'DF 18.12.14 the value of each export file per analysis in on every row and opened every time.

'it could be quicker to open the export once instead of every single row.

WorkbookName = ActiveCell.Offset(0, 6).Value 'get the export filename to look up values in

Set ExportFileWB = Excel.Application.Workbooks.Open(WorkbookName, False, True, 2)

ExportFileWB.Activate

Set ExportFileWS = ExportFileWB.Sheets(1) ' set the export worksheet to the first worksheet in the export file

'MsgBox "export file opened"

ExportFileWB.ActiveSheet.Columns("A:Z").AutoFit 'resize all the column to autofit

Dim k As Integer ' define the for loop counter

' now do for each element in the array

For k = 0 To UBound(MyRTArr()) ' loop on each RT value in the array (0 based)

inRT = MyRTArr(k) ' DF 18.12.14 store the RT from the array now

ExportFileWS.Range("B2").Select 'select the first cell in the list

Dim foundmatch As Boolean

foundmatch = False 'init as false

AreaFound = "" 'reset the values

RangeFound = ""

SNFound = ""

IntensityFound = ""

Do Until Excel.ActiveCell.Value = ""

Dim inRTUpper

Dim inRTLower

inRTUpper = inRT + 0.2 ' set the tolerance upper level

inRTLower = inRT - 0.2 ' set the tolerance lower level

'MsgBox (ActiveCell.Value >= inRTLower)

'MsgBox (ActiveCell.Value = inTRLower)

'MsgBox (Val(ActiveCell.Value))

'MsgBox (Val(inRTLower))

'MsgBox (Val(ActiveCell.Value) = Val(inRTLower))

'df 08.01.2015 change to values to ensure strings are converted to numbers

If ((Val(ActiveCell.Value) >= Val(inRTLower)) And (Val(ActiveCell.Value) <= Val(inRTUpper))) Then 'if the value in the lookup sheet is within + or - 0.02

Dim ChromStartPos

Dim ChromEndPos

Dim ChromValue

If InStr(1, ActiveCell.Offset(0, 2).Value, "EIC ") > 0 Then

ChromStartPos = InStr(1, ActiveCell.Offset(0, 2).Value, "EIC ")

If InStr(ChromStartPos, ActiveCell.Offset(0, 2).Value, "±") > 0 Then

ChromEndPos = InStr(ChromStartPos, ActiveCell.Offset(0, 2).Value, "±")

ChromValue = VBA.Mid(ActiveCell.Offset(0, 2).Value, ChromStartPos + 4, ChromEndPos - (ChromStartPos + 4))

'MsgBox ChromValue

End If

End If ' end if start pos found

'MsgBox "found a match molecule match"

If InStr(1, ActiveCell.Offset(0, 2).Value, "EIC " & inMoleculeSize) > 0 Then ' if the molecule size is found eg(EIC 415.730000)

If (Val(ChromValue) = Val(inMoleculeSize)) Then

'MsgBox "found a match of RT and molecule"

foundmatch = True 'set that we have found a match

'pick up the values and put them back on the master spreadsheet

'df 18.12.14 remove vars and add to array elements for each rt loop

'AreaFound = ActiveCell.Offset(0, 3).Value

'RangeFound = ActiveCell.Offset(0, 4).Value

'SNFound = ActiveCell.Offset(0, 5).Value

'IntensityFound = ActiveCell.Offset(0, 6).Value

AreaFoundArr(k) = ActiveCell.Offset(0, 3).Value

RangeFoundArr(k) = ActiveCell.Offset(0, 4).Value

SNFoundArr(k) = ActiveCell.Offset(0, 5).Value

IntensityFoundArr(k) = ActiveCell.Offset(0, 6).Value

End If 'end if EIC molecule size matches

End If 'df 05.01.2015 end if EIC {MoleculeSize} was found in the cell

End If 'end RT match found if

If foundmatch = True Then

Exit Do 'exit loop when found the first match

End If 'end if found match

ActiveCell.Offset(1, 0).Select 'move to the next row in the export spreadsheet

Loop 'end do until active cell is emty

If foundmatch = False Then

'MsgBox "no match found"

'update the spreadsheet with (not found)

'df dont add values yet now 18.12.14

' MasterWB.Activate 'found a match update the values in the master spreadsheet

'

' Excel.ActiveCell.Offset(0, 1).Value = "N/A"

' Excel.ActiveCell.Offset(0, 2).Value = "N/A"

' Excel.ActiveCell.Offset(0, 3).Value = "N/A"

' Excel.ActiveCell.Offset(0, 4).Value = "N/A"

AreaFoundArr(k) = "N/A"

RangeFoundArr(k) = "N/A"

SNFoundArr(k) = "N/A"

IntensityFoundArr(k) = "N/A"

Else

' df 18.12.14 values already added to the array removed update excel code

' MasterWB.Activate 'found a match update the values in the master spreadsheet

' Excel.ActiveCell.Offset(0, 1).Value = AreaFound

' Excel.ActiveCell.Offset(0, 2).Value = RangeFound

' Excel.ActiveCell.Offset(0, 3).Value = IntensityFound

' Excel.ActiveCell.Offset(0, 4).Value = SNFound

End If 'end if match found

Next k 'new split '/' rt value in the array

'now loop on the RT array again to tally the values found

For k = 0 To UBound(MyRTArr()) ' loop on each RT value in the array (0 based)

If (AreaFoundArr(k) = "N/A") Then ' if not found

'dont add it

Else

AreaFound = Val(AreaFound) + Val(AreaFoundArr(k)) 'tally the found area

End If

If (k = UBound(MyRTArr)) Then 'if this is the last element

'no separaotr for the last value

RangeFound = RangeFound & RangeFoundArr(k) 'add the vlaue

Else

'value plus separator

RangeFound = RangeFound & RangeFoundArr(k) & "," 'separate each value with a comma

End If

If (SNFoundArr(k) = "N/A") Then ' if not found

'dont add it

Else

SNFound = Val(SNFound) + Val(SNFoundArr(k)) 'tally the SN

End If

If (IntensityFoundArr(k) = "N/A") Then ' if not found

'dont add it

Else

IntensityFound = Val(IntensityFound) + Val(IntensityFoundArr(k)) 'tally the SN

End If

Next k 'new split '/' rt value in the array

'now output tally to spreadsheet

MasterWB.Activate 'found a match update the values in the master spreadsheet

If (AreaFound = "") Then 'if all were blank

AreaFound = "N/A" 'set to n/a

End If

If (SNFound = "") Then 'if all were blank

SNFound = "N/A" 'set to n/a

End If

If (IntensityFound = "") Then 'if all were blank

IntensityFound = "N/A" 'set to n/a

End If

Excel.ActiveCell.Offset(0, 1).Value = AreaFound

Excel.ActiveCell.Offset(0, 2).Value = RangeFound

Excel.ActiveCell.Offset(0, 3).Value = IntensityFound

Excel.ActiveCell.Offset(0, 4).Value = SNFound

ExportFileWB.Close (False) ' close the export file without saving changes

MasterWB.Activate ' activate the master spreadsheet again.

End Sub

Public Sub TestSoftware()

Set MyApp = New DataAnalysis.Application

MsgBox "Application Name is:" & MyApp.Name

MsgBox "Application Versions is:" & MyApp.Version

End Sub

Public Sub DoALL()

'method to do everything from 1 excelspreadsheet

Set MasterWB = Excel.Application.ActiveWorkbook 'set this master spreadsheet to the master spreadsheet

'

Do Until (Excel.ActiveCell.Value = "") 'loop until current cell is empty

Call OpenDatafile 'open the data file and store the current cells range

Call AddEICByDefinition 'scroll down the EIC column adding EICs until we get to a blank row

Call SmoothEICs 'smooth all EICs in the analysis file (no moving excel cells)

Call IntergrateEICs 'intergrateEICs in the analysis file (no moving excel cells)

Call ExportData 'the exports the data for the current analysis only needed once (doesn't need to move excel rows as it just exports data -filename was already added each row during addEics)

AnalysisCell.Activate 'df 22.12.14 active the cell of the analysis file

ActiveCell.Offset(0, 4).Select 'df 22.12.14 move the cell over to the RT column ready for data mine

Call MineAllRTs 'df 22.12.14 scrolls down the RT column and looks up export file and popualtes the data fields with the matches

'finished one row after the last RT value then moves down one more row past the blank row and 4 columns to the left (next analysis filename cell).

Loop ' loop

MsgBox "No More files to process"

End Sub

Public Sub OpenDatafile()

' sub routine to open a datafile

'get the file name from the active cell in excel

Set AnalysisCell = Excel.ActiveCell 'store the starting active cell

Dim MyFile As String

MyFile = Excel.ActiveCell.Value 'get the filename from the currently selected excel cell

'MsgBox ("attempting to open file:" + MyFile)

Set MyApp = New DataAnalysis.Application

MyApp.Analyses.Open (MyFile) 'open analysis file

Set MyAnalysis = MyApp.ActiveAnalysis 'set the newly opened analysis to the current one

'MsgBox "Name of opened analysis file is:" & MyAnalysis.Name 'get the name of the opened file

End Sub

Public Sub AddEICByDefinition()

'starts of the first row in the filename column, then moves over 3 columns to the EIC colum

'then moves down a row at a time adding eics and also the export files name 7 column across from the eic column

'until there is a blank row

Set MyApp = New DataAnalysis.Application

Set MyAnalysis = MyApp.ActiveAnalysis 'set the newly opened analysis to the current one

Dim MyMoleculeSize 'input size of molecule

'Dim MyMoleculesArr(0) As String 'create an array of molecule sizes to input

Excel.ActiveCell.Offset(0, 3).Select 'move excel 3 column to the right Molecule size column (EIC COLUMN)

MyMoleculeSize = Excel.ActiveCell.Value

Dim i As Integer

Dim CompoundFilename As String

'new code to check values are -+ 0.02 before added

Dim aEICsAdded() As Double 'dimension array type as a double

ReDim aEICsAdded(0) 'set the array size to 1 element (0 based index)

aEICsAdded(0) = -9999 'set an impossible low value for the first element value

Dim CurrentEICvalue As Double

'CurrentEICvalue = ActiveCell.Value 'store the current cells eic value

Dim useEIC As Boolean

useEIC = True 'initialse as true

Dim j As Integer ' define the for loop counter

Dim anyValuesExported As Boolean ' define variable to know if any value was used/exported in the cell

'end new code

'df 22.12.14 changed to empty cell Do Until (ActiveCell.Value = "END") 'keep going until we see END in the cell

Do Until (ActiveCell.Value = "") 'keep going until we see a blank cell in the cell

anyValuesExported = False 'initailse each cell as nothing exported yet

If (ActiveCell.Offset(0, 1).Value = 0) Then ' if the RT column is 0

' then we do not add this EIC (do nothing)

Else 'RT column was not 0 so we will add it

MyMoleculeSize = Excel.ActiveCell.Value

Dim MyMoleculesArr() As String

MyMoleculesArr = Split(MyMoleculeSize, "/", , vbTextCompare) 'create an array of the molecule size 1 element for every "/" char

For i = 0 To UBound(MyMoleculesArr) ' loop on each element in the array

CurrentEICvalue = Val(MyMoleculesArr(i)) 'convert the string into a number so we can range check

useEIC = True ' set use flag to true initailly

'new code check if molecules is within - = 0.02 of already added value before readding.

For j = LBound(aEICsAdded) To UBound(aEICsAdded)

'code update with smaller duplicate window 25 01 15

If ((CurrentEICvalue + 0.01) >= aEICsAdded(j)) Then 'if the current value + 0.02 is greater or = to the array element

If ((CurrentEICvalue - 0.01) <= aEICsAdded(j)) Then 'if the current value -0.02 is less than or = to the array element

useEIC = False ' set use flag to false as it is within + - 0.02 of already added values

Exit For 'exit the loop as there is no need to check other values now

End If

End If

Next j 'loop to next value in the added values array

If (useEIC = True) Then 'now the loop has finished comparing added value check if we will use the eic

anyValuesExported = True 'set flag that we created an export file

If (UBound(aEICsAdded) = 0) Then 'if this is the first element

If (aEICsAdded(0) <> -9999) Then 'if a value has been added (its not -9999)

ReDim Preserve aEICsAdded(UBound(aEICsAdded) + 1) 'add another element to the array

End If

Else

ReDim Preserve aEICsAdded(UBound(aEICsAdded) + 1) 'add another element to the array

End If

aEICsAdded(UBound(aEICsAdded)) = CurrentEICvalue 'add the current cell value to the highest element in the array

'do thie adding of the eic here

'code updated to give larger EIC window 25 01 15

Dim MyEICDefinition As New DataAnalysis.EICChromatogramDefinition

MyEICDefinition.BackgroundType = daBgrdTypeNone

'MyEICDefinition.MSFilter.FragmentationMode = daAny

'MyEICDefinition.MSFilter.FragmentationPath = ""

MyEICDefinition.MSFilter.Type = daMSFilterMS 'filter to al ms

MyEICDefinition.Polarity = daBoth

'MyEICDefinition.Range = "408.730620" 'combined range, e.g., “290 – 350; 400” or single value

MyEICDefinition.Range = MyMoleculesArr(i) 'add the size of current array element 'combined range, e.g., “290 – 350; 400” or single value

MyEICDefinition.ScanMode = daScanModeAll

MyEICDefinition.WidthLeft = 0.025 'Extends the range if it is defined with single values to the left and to the right of a ChromatogramDefinition object.

MyEICDefinition.WidthRight = 0.025 'Extends the range if it is defined with single values to the left and to the right of a ChromatogramDefinition object.

'add analysis with the definition we have just created

MyAnalysis.Chromatograms.AddChromatogram MyEICDefinition 'add the definition to the analysis

Else 'we will not add this eic

ActiveCell.Font.Color = vbRed 'set the font as strike through as we didn't use this value

End If

Next i 'move to the next element in the molecule array

End If 'end if RT column was 0

If (anyValuesExported = True) Then

'put the export filename in the end column

'CompoundFilename = "GOOD"

'

'put the file name back anyway as we still look up values or eics even if they werenot added on this row due to dupes df 18.12.2014

CompoundFilename = "C:\zzHistomatic\" & MyAnalysis.Name & "CompoundList.csv" 'set up the variable for the export compound filename

Excel.ActiveCell.Offset(0, 7).Value = CompoundFilename 'output the export file name to the output filename cell

'

Else

'didn't export anything create a not exported filename entry

'put the export filename in the end column

'CompoundFilename = "Not Exported" 'set up the variable for the export compound filename

'

CompoundFilename = "C:\zzHistomatic\" & MyAnalysis.Name & "CompoundList.csv" 'set up the variable for the export compound filename

Excel.ActiveCell.Offset(0, 7).Value = CompoundFilename 'output the export file name to the output filename cell

'

End If

Excel.ActiveCell.Offset(1, 0).Select 'move the current cell 1 row down

Loop ' keep going until we reach END

'Excel.ActiveCell.Offset(1, -3).Select 'move the current cell 1 row down and back one column (filename)

'MsgBox "EICs added"

'by the end we will be on the first blank row after the last eic in the eic column

End Sub

Public Sub SmoothEICs()

'smooth all EIC chroms found in the analysis

Set MyApp = New DataAnalysis.Application

Set MyAnalysis = MyApp.ActiveAnalysis 'set the newly opened analysis to the current one

'MyAnalysis.MethodDAPart.ProcessParameters.ChromSmoothingAlgorithm = 1 ' set to 1 for degauss

'MyAnalysis.MethodDAPart.ProcessParameters.ChromSmoothingCycles = 1

'MyAnalysis.MethodDAPart.ProcessParameters.ChromSmoothingWidth = 1.1

Dim chrom As Chromatogram

For Each chrom In MyAnalysis.Chromatograms

pos = InStr(chrom.Name, "EIC") 'look for EIC in the chrom name

If pos = 1 Then

chrom.Smooth 'smooth any found EIC chromatograms

End If

Next chrom

End Sub

Public Sub IntergrateEICs()

'run the intergrate EIC method on all EIC chroms

Set MyApp = New DataAnalysis.Application

Set MyAnalysis = MyApp.ActiveAnalysis 'set the newly opened analysis to the current one

'intergate ALL chroms??

'MsgBox MyAnalysis.Chromatograms.Count

'MyAnalysis.Chromatograms.Enable 'this will intergrate all chroms in the anaylsis

'or only intergrate the EIC smoothed chroms

Dim chrom As Chromatogram

For Each chrom In MyAnalysis.Chromatograms

'determin if chrom is smoothed?

'chrom.IntegrateOnly

pos = InStr(chrom.Name, "EIC") 'look for EIC in the chrom name

possmooth = InStr(chrom.Name, "Smoothed") ' look for smoothed in the chrom name

'If (pos = 1) And (posssmoth > 1) Then ' if this is a smoothed EIC

If (pos = 1) Then ' if this is a EIC

'MsgBox "chrom has children count of" & chrom.ChildChromatograms.Count

'enable all children of the chrom (smoothed croms) to 1 level deep

For x = 1 To chrom.ChildChromatograms.Count

Dim childcrom As Chromatogram

Set childcrom = chrom.ChildChromatograms.Item(x)

childcrom.Enable 'make sure child (smoothed assumed) is enabled so it intergrates

childcrom.IntegrateOnly 'intergrate the child smoothed crom

Next x

chrom.IntegrateOnly 'intergrate the parent chrom?

End If

Next chrom

'MsgBox "intergrated complete"

End Sub

Public Sub ExportData()

' this goes down the EIC columna and outputs the export filename 7 columns to the right

Set MyApp = New DataAnalysis.Application

Set MyAnalysis = MyApp.ActiveAnalysis 'set the newly opened analysis to the current one

'MsgBox "getting compounds"

'MyAnalysis.FindAllCompounds 'find the compounds

'MsgBox "this analysis has compunds count of" & MyApp.ActiveAnalysis.Compounds.Count

Dim MyCompound As DataAnalysis.Compound

'we can either export the compound list as a csv or loop through them and output propeties of the compound directly to excel somewhere

'df 22.12.14 we have already put the filename in the export column and we added each eic value

' so remove the output filename to excel colum code

Dim CompoundFilename As String

CompoundFilename = "C:\zzHistomatic\" & MyAnalysis.Name & "CompoundList.csv" 'set up the variable for the export compound filename

'df 22.12.14 removed 'Excel.ActiveCell.Offset(0, 7).Value = CompoundFilename 'output the export file name to the output filename cell

'MyAnalysis.Compounds.Export "C:\zzTempDion\compoundexport.csv", daCSV 'hard coded filename output test

MyAnalysis.Compounds.Export CompoundFilename, daCSV 'output the compound file based on the name of the analysis

'MsgBox (MyAnalysis.Compounds.Count())

'MsgBox "compounds exported to:" & CompoundFilename

Excel.Worksheets(2).Activate 'active the page 2 worksheet in the current spreadsheet

Excel.ActiveCell.Value = MyAnalysis.Name & " ==== Data Exported to:" & CompoundFilename 'output the filename on the first row

For Each MyCompound In MyAnalysis.Compounds 'loop through the collection of compounds

'MsgBox MyCompound.Chromatogram 'the chromatogram for the compound

'MsgBox MyCompound.CompoundNumber

'other paramaters of compunds are available

ActiveCell.Offset(1, 0).Select 'select the next row

ActiveCell.Offset(0, 0).Value = MyCompound.CompoundNumber ' output the compound number to first column (no offset)

ActiveCell.Offset(0, 1).Value = MyCompound.RetentionTime 'output the retentiontime to col 2

ActiveCell.Offset(0, 2).Value = MyCompound.Chromatogram 'output the chromatogram to the column 3(0 row offset and 2 column offset)

ActiveCell.Offset(0, 3).Value = MyCompound.Area 'output area to column 4

'ActiveCell.Offset(0, 4).Value = MyCompound.Width 'output range to column 4

ActiveCell.Offset(0, 4).Value = MyCompound.SignalToNoise 'output s/n to column 5

ActiveCell.Offset(0, 5).Value = MyCompound.Intensity 'output intensity to column 6

Next MyCompound

ActiveCell.Offset(1, 0).Select 'select the next row of the spreadsheet (sheet 2)

'data mining ??

'Excel.Workbooks(2).Select

'Excel.ActiveSheet.Clear

Excel.Worksheets(1).Activate ' active page 1 of the excel workbook again

'Excel.ActiveCell.Offset(1, 0).Select 'move to next row to new filename position

'MsgBox "compound data propeties written to sheet 2"

'close or save the analysis ???

MyAnalysis.Save 'save the analysis before closing it

MyAnalysis.Close

'now we would need to compare the exported compound list data on page 2 of the worksheet

'with the input data on page 1 of the worksheet in code

'could use vb excel function to seach for x and report back y ??

'eg navigate to input data, look it up in the output data

'result = Excel.WorksheetFunction.Find("x", "y")

'if (result=z) then

' do something

'end if

'after we get to the end of first file we could put another loop here to move on to next file in excel and next lot of input params

'and clear previous compound outputs on page 2 from previous file

'MsgBox "finished"

End Sub

Public Sub zzAddEICjunk()

'attempt to add and extracted ion chromatagram

'MyAnalysis.Chromatograms.Add daEIC, daAllMS, "", daBoth 'add an extracted ion chromatogram, All MS filter all range both polarity to the analysis

Set MyApp = New DataAnalysis.Application

Set MyAnalysis = MyApp.ActiveAnalysis 'set the newly opened analysis to the current one

'add an eic for the values in excel

Dim MyMoleculeSize

'get the first cell 1 column over from the file name

'do until active cell empty

Excel.ActiveCell.Offset(0, 1).Select 'move excel 1 column to the right

Dim NumOfChroms

NumOfChroms = MyAnalysis.Chromatograms.Count ' get the current number of chroms.

Do Until (ActiveCell.Value = "END")

'MsgBox "current molecule size is" & ActiveCell.Value

MyAnalysis.Chromatograms.Add daEIC, ActiveCell.Value, "", daBoth ' add a chromatogram with the current molecule size

NumOfChroms = MyAnalysis.Chromatograms.Count ' get the new current number of chroms.

MyAnalysis.Chromatograms(NumOfChroms).Select (False) 'select the last (newly added chrom)

'MyAnalysis.Chromatograms(NumOfChroms).Smooth 'smooth the crom

'MyAnalysis.Chromatograms(NumOfChroms).IntegrateOnly 'intergrate the chrom

ActiveCell.Offset(1, 0).Select 'move excel to the next row

Loop 'loop until we have a cell that contains "END"

'MyAnalysis.Chromatograms.Add daEIC, "408.72", "", daBoth

'MyAnalysis.Chromatograms.Add daEIC, daAllMS, "", daBoth

'MyAnalysis.Chromatograms.Add daEIC, daAllMS, "408.72", daBoth

'MyAnalysis.Chromatograms.Add daEIC, daAllMS, "-", daBoth

' An alternative way to add an EIC may be with a definition testing outside the loop of input data from excel

Dim MyEICDefinition As New DataAnalysis.EICChromatogramDefinition

MyEICDefinition.BackgroundType = daBgrdTypeConstant

MyEICDefinition.MSFilter.FragmentationMode = daAny

'MyEICDefinition.MSFilter.FragmentationPath = ""

MyEICDefinition.MSFilter.Type = daMSFilterAll

'MyEICDefinition.Name = "TEST"

MyEICDefinition.Polarity = daBoth

MyEICDefinition.Range = "300-400" 'combined range, e.g., “290 – 350; 400” or single value

MyEICDefinition.ScanMode = daScanModeFullScan

'MyEICDefinition.Type = ""

MyEICDefinition.WidthLeft = 0 'Extends the r#ange if it is defined with single values to the left and to the right of a ChromatogramDefinition object.

MyEICDefinition.WidthRight = 0 'Extends the range if it is defined with single values to the left and to the right of a ChromatogramDefinition object.

'add analysis by definition

'MyAnalysis.Chromatograms.AddChromatogram MyEICDefinition 'add the definition to the analysis

For Each chrom In MyAnalysis.Chromatograms 'loop through each chrom in the file

' Access Chrom here.

' For example:

'Chrom.IntegrateOnly

'how do we determine the newly added chroms?

MsgBox chrom.Name 'display the name of the chrom

If (chrom.Name = "") Then 'if this is a chrom we want to do something with

'do something for this chrom

End If

chrom.FindCompounds 'find compounds at a chrom level? or analysis level?

Next 'loop

'now all the chromatograms for each molecule size have been added, smoothed and integrated

'export the compond list

MsgBox "getting compounds"

MyAnalysis.FindAllCompounds 'find the compounds

MsgBox "this analysis has compunds count of" & MyApp.ActiveAnalysis.Compounds.Count

Dim MyCompound As DataAnalysis.Compound

'we can either export the compound list as a csv or loop through them and output propeties of the compound directly to excel somewhere

Dim CompoundFilename As String

CompoundFilename = "C:\zzHistomatic\" & MyAnalysis.Name & "CompoundList.csv" 'set up the variable for the export compound filename

'MyAnalysis.Compounds.Export "C:\zzTempDion\compoundexport.csv", daCSV 'hard coded filename output test

MyAnalysis.Compounds.Export CompoundFilename, daCSV 'output the compound file based on the name of the analysis

MsgBox "compounds exported to:" & CompoundFilename

For Each MyCompound In MyAnalysis.Compounds 'loop through the collection of compounds

'MsgBox MyCompound.Chromatogram 'the chromatogram for the compound

'MsgBox MyCompound.CompoundNumber

'other paramaters of compunds are available

Excel.Worksheets(2).Activate 'active the page 2 worksheet in the current spreadsheet

ActiveCell.Offset(1, 0).Select 'select the next row

ActiveCell.Offset(0, 0).Value = MyCompound.CompoundNumber ' output the compound number to first column (no offset)

ActiveCell.Offset(0, 1).Value = MyCompound.RetentionTime 'output the retentiontime to col 2

ActiveCell.Offset(0, 2).Value = MyCompound.Chromatogram 'output the chromatogram to the column 3(0 row offset and 2 column offset)

ActiveCell.Offset(0, 3).Value = MyCompound.Area 'output area to column 4

Next MyCompound

Excel.Worksheets(1).Activate ' active page 1 of the excel workbook again

MsgBox "compound data propeties written to sheet 2"

'close or save the analysis ???

MyAnalysis.Save

'MyAnalysis.Close

'now we would need to compare the exported compound list data on page 2 of the worksheet

'with the input data on page 1 of the worksheet in code

'could use vb excel function to seach for x and report back y ??

'eg navigate to input data, look it up in the output data

'result = Excel.WorksheetFunction.Find("x", "y")

'if (result=z) then

' do something

'end if

'after we get to the end of first file we could put another loop here to move on to next file in excel and next lot of input params

'and clear previous compound outputs on page 2 from previous file

'Excel.Workbooks(2).Select

'Excel.ActiveSheet.Clear

MsgBox "finished"

End Sub

Public Sub TestEICArraySkip()

Dim aEICsAdded() As Double 'dimension array type as a double

ReDim aEICsAdded(0) 'set the array size to 1 element (0 based index)

aEICsAdded(0) = -9999 'set an impossible low value for the first element value

Dim CurrentEICvalue As Double

'CurrentEICvalue = ActiveCell.Value 'store the current cells eic value

Dim useEIC As Boolean

useEIC = True 'initialse as true

Dim i As Integer ' define the for loop counter

Dim j As Integer

Dim MyMoleculeSize 'input size of molecule

Do Until (ActiveCell.Value = "")

MyMoleculeSize = Excel.ActiveCell.Value

Dim MyMoleculesArr() As String

MyMoleculesArr = Split(MyMoleculeSize, "/", , vbTextCompare) 'create an array of the molecule size 1 element for every "/" char

For j = 0 To UBound(MyMoleculesArr) ' loop on each element in the array

useEIC = True 'reset to true for next cell

CurrentEICvalue = Val(MyMoleculesArr(j)) 'convert the string into a number so we can range check

For i = LBound(aEICsAdded) To UBound(aEICsAdded)

If ((CurrentEICvalue + 0.02) >= aEICsAdded(i)) Then 'if the current value + 0.02 is greater or = to the array element

If ((CurrentEICvalue - 0.02) <= aEICsAdded(i)) Then 'if the current value -0.02 is less than or = to the array element

useEIC = False ' set use flag to false as it is within + - 0.02 of already added values

Exit For 'exit the loop as there is no need to check other values now

End If

End If

Next i

If (useEIC = True) Then 'now the loop has finished comparing added value check if we will use the eic

If (UBound(aEICsAdded) = 0) Then 'if this is the first element

If (aEICsAdded(0) <> -9999) Then 'if a value has been added (its not -9999)

ReDim Preserve aEICsAdded(UBound(aEICsAdded) + 1) 'add another element to the array

End If

Else

ReDim Preserve aEICsAdded(UBound(aEICsAdded) + 1) 'add another element to the array

End If

aEICsAdded(UBound(aEICsAdded)) = CurrentEICvalue 'add the current cell value to the highest element in the array

'do thie adding of the eic here

Else

ActiveCell.Font.Color = vbRed 'set the font as strike through as we didn't use this value

End If

Next j

ActiveCell.Offset(1, 0).Select 'move cell one down row

' CurrentEICvalue = ActiveCell.Value 'store the next cell value as the active cell

Loop

MsgBox "finished" & (UBound(aEICsAdded))

End Sub

Public Sub zzSetColorJunk(inColor As String)

Dim MyColor As DataAnalysis.DaColors

Select Case (inColor)

Case of(inColor) = "black"

MyColor = daBlack

Case of(inColor) = "white"

MyColor = daWhite

Case of(inColor) = "red"

MyColor = daRed

Case of(inColor) = "bright green"

MyColor = daBrightGreen

Case of(inColor) = "blue"

MyColor = daBlue

Case of(inColor) = "yellow"

MyColor = daYellow

Case of(inColor) = "pink"

MyColor = daPink

' and so on

'daBlack black

'daWhite white

'daRed red

'daBrightGreen bright green

'daBlue blue

'daYellow yellow

'daPink pink

'daTurquoise turquoise

'daDarkRed dark red

'daGreen green

'daDarkBlue dark blue

'daDarkYellow dark yellow

'daViolet violet

'daTeal teal

'daGray25 25% gray

'daGray50 50% gray

'daSkyBlue sky blue

'daLightTurquoise light turquoise

'daLightGreen light green

'daLightYellow light yellow

'daPaleBlue pale blue

'daRose rose

'daLavender lavender

'daTan Tan

'daLightBlue light blue

'daAqua aqua

'daLime lime

'daGold gold

'daLightOrange light orange

'daOrange orange

'daBlueGray blue gray

'daGray40 40% gray

'daDarkTeal dark teal

'daSeaGreen sea green

'daDarkGreen dark green

'daOliveGreen olive green

'daBrown brown

'daPlum plum

'daIndigo indigo

'daGray80 80% gray

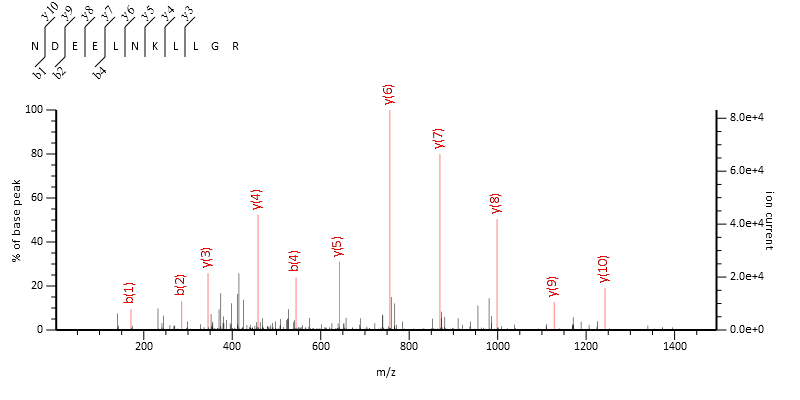
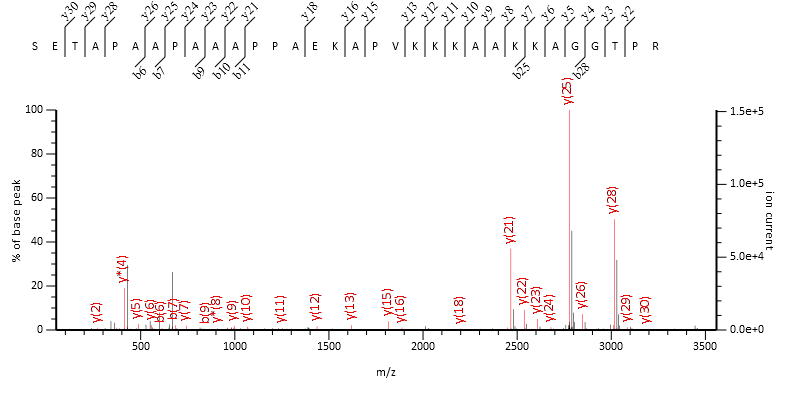
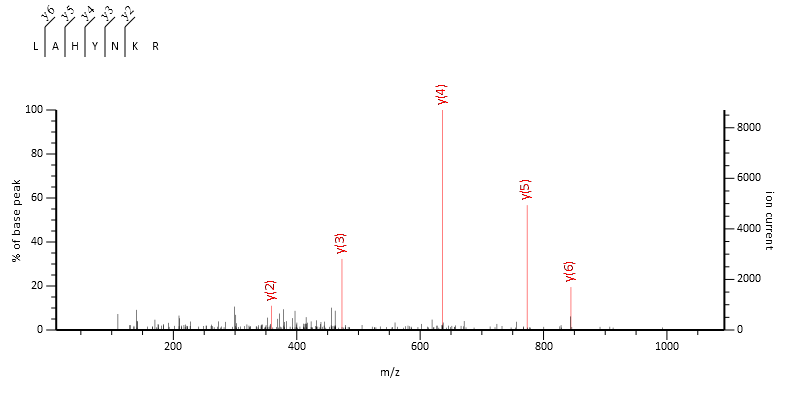
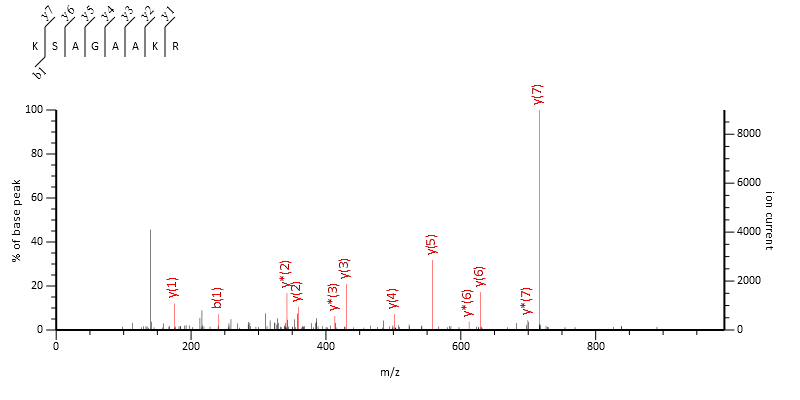
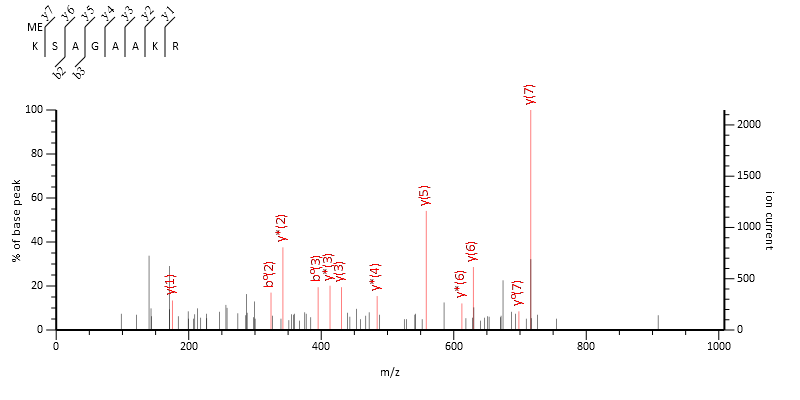
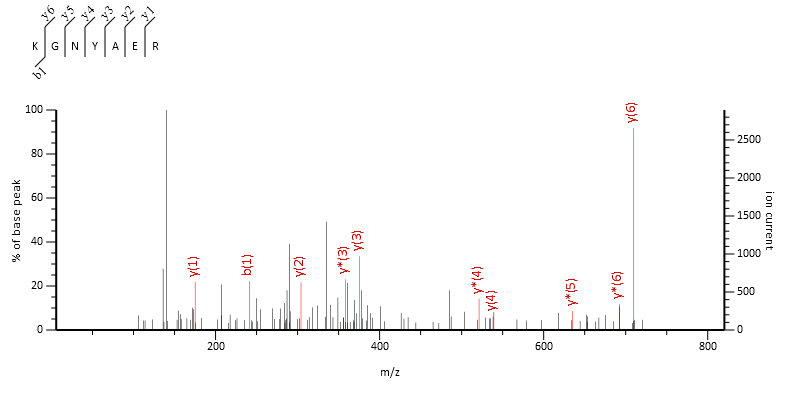
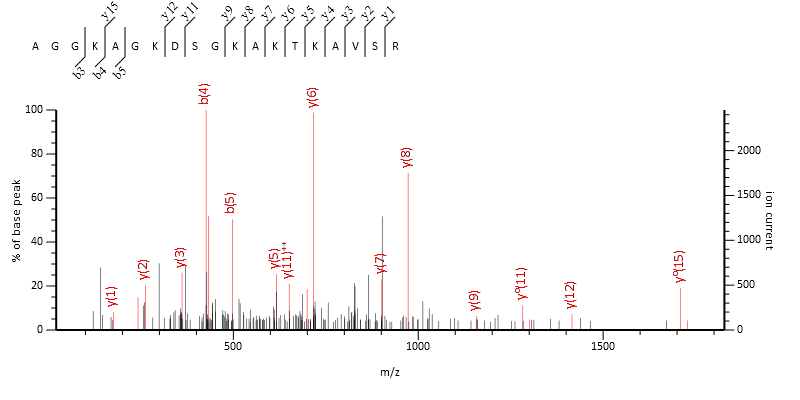
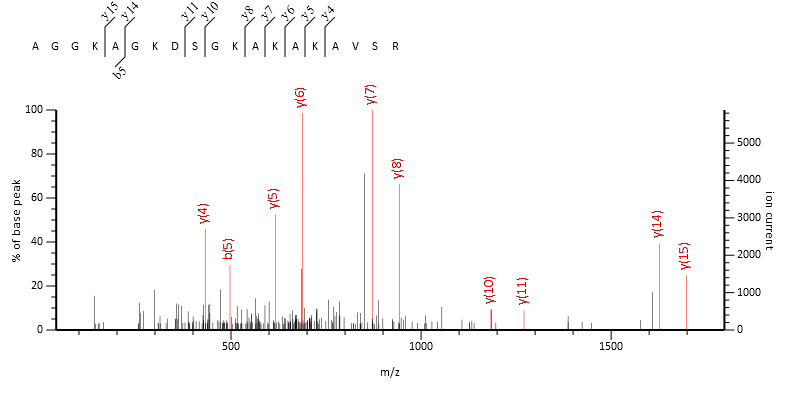
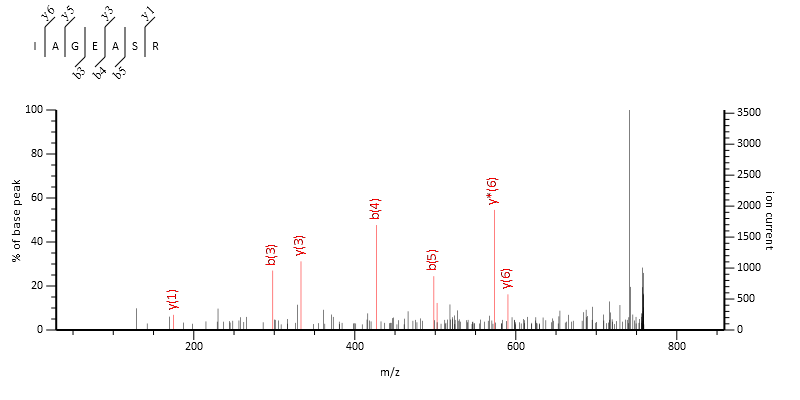
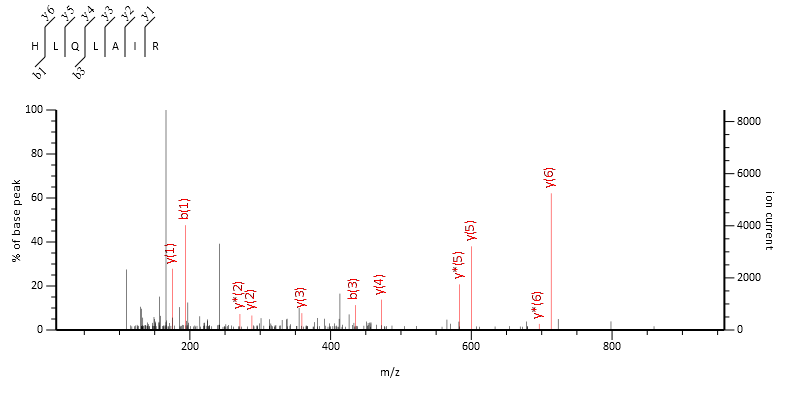
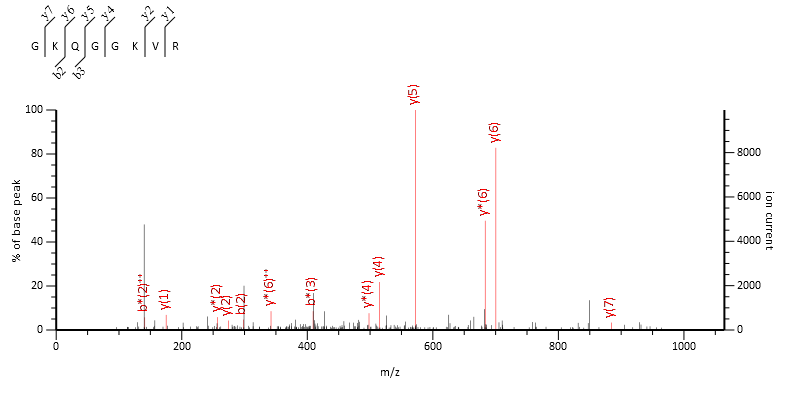
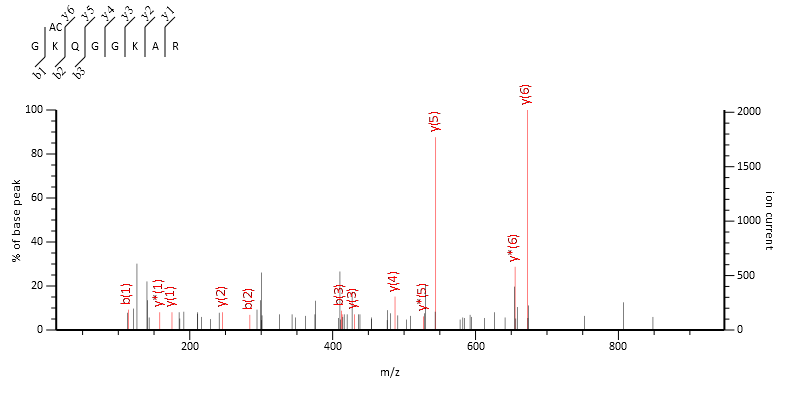
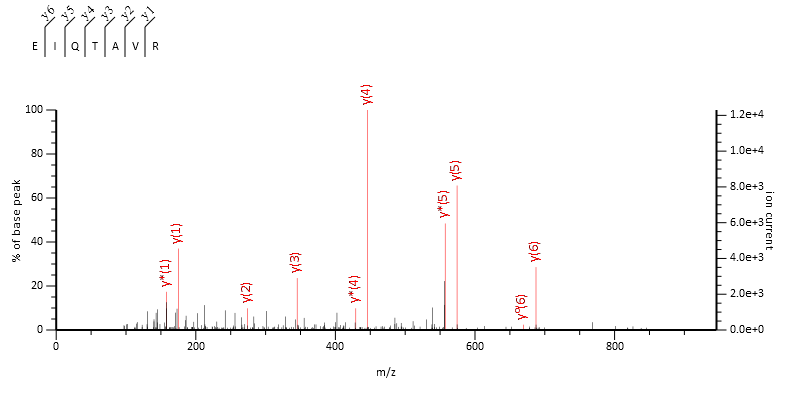
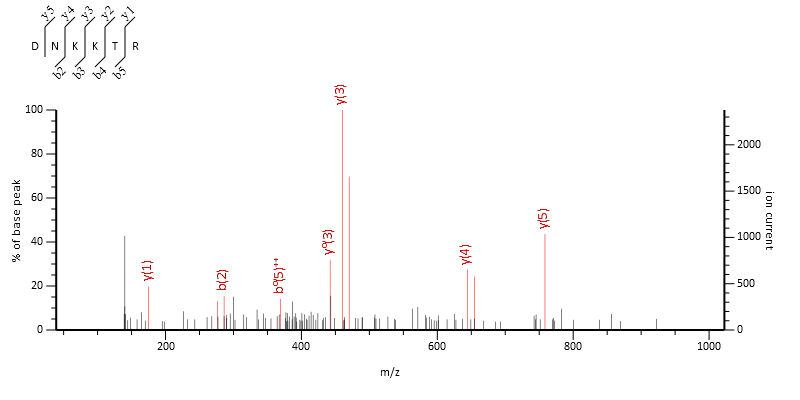
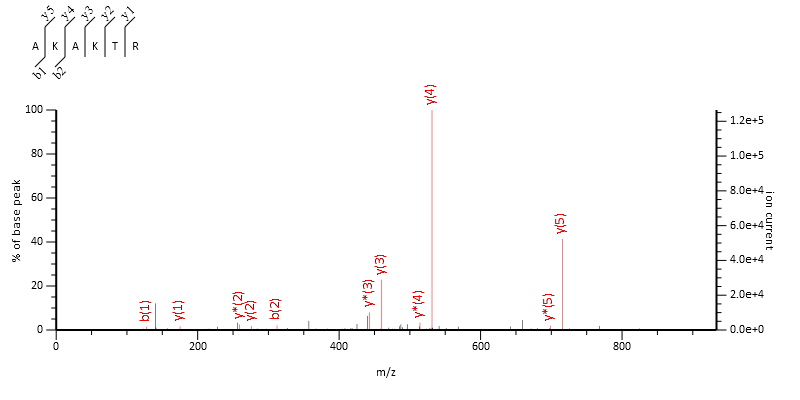
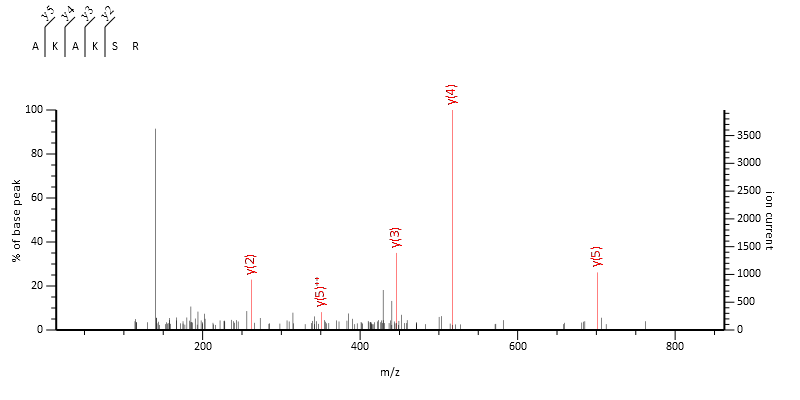
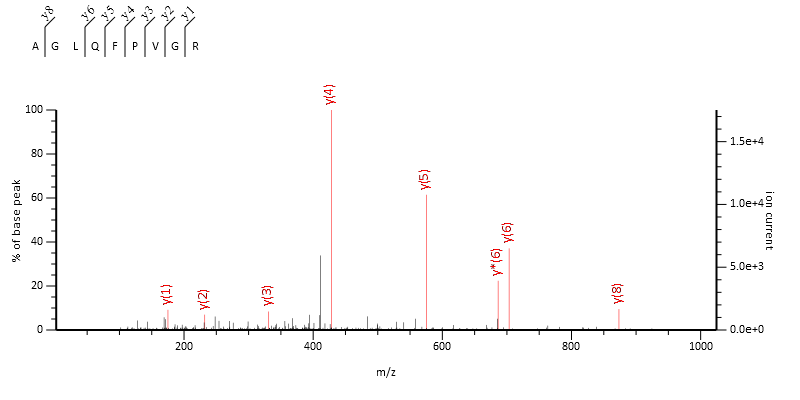
End Select

End Sub

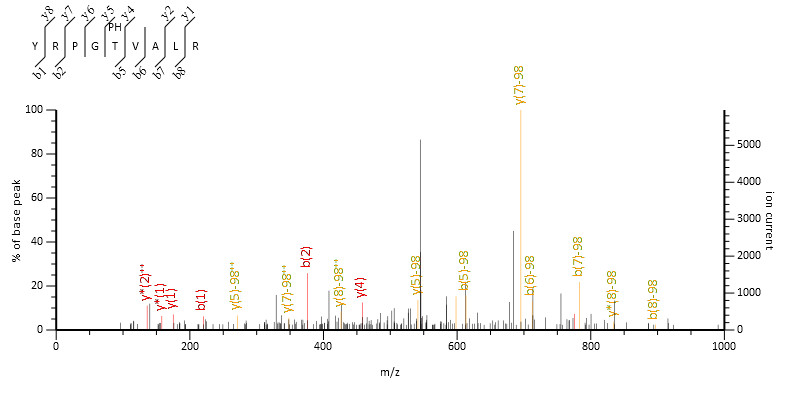
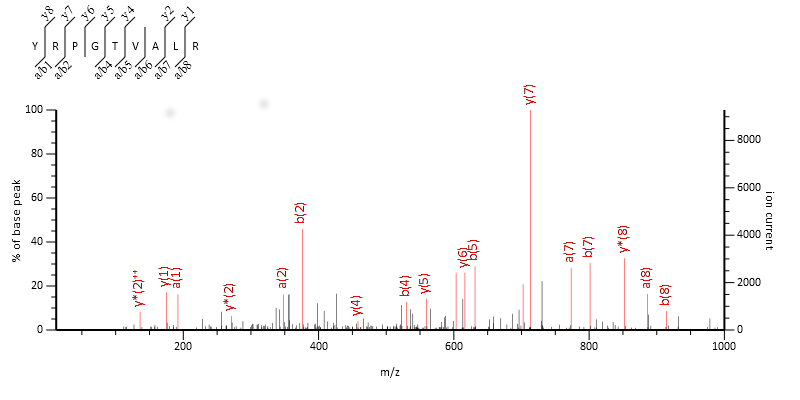
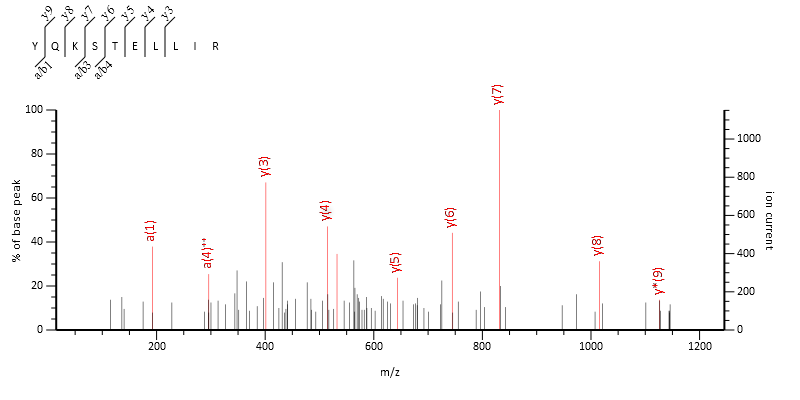
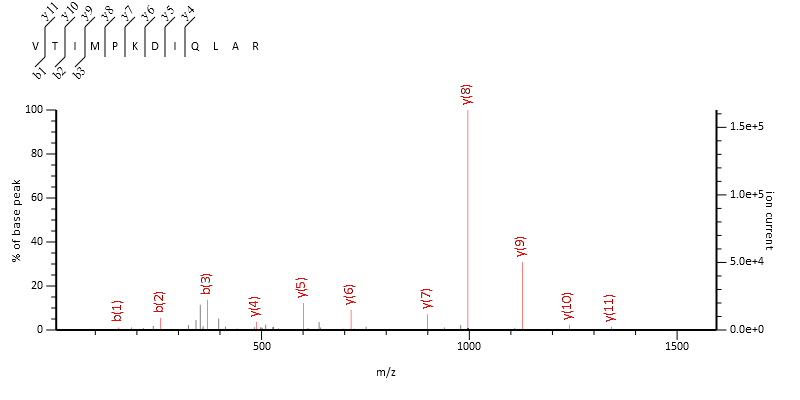
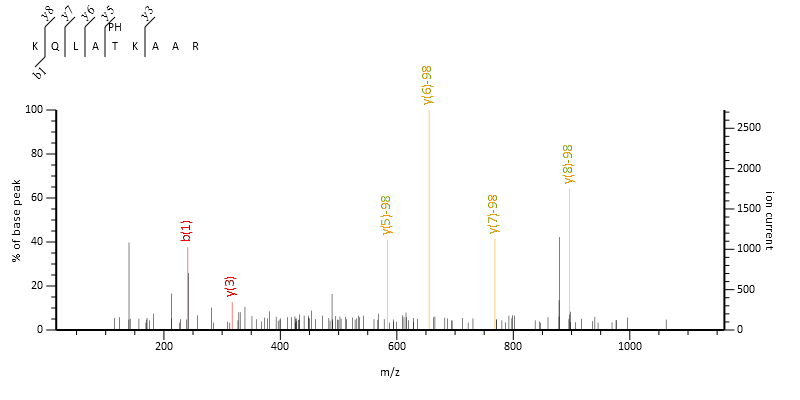
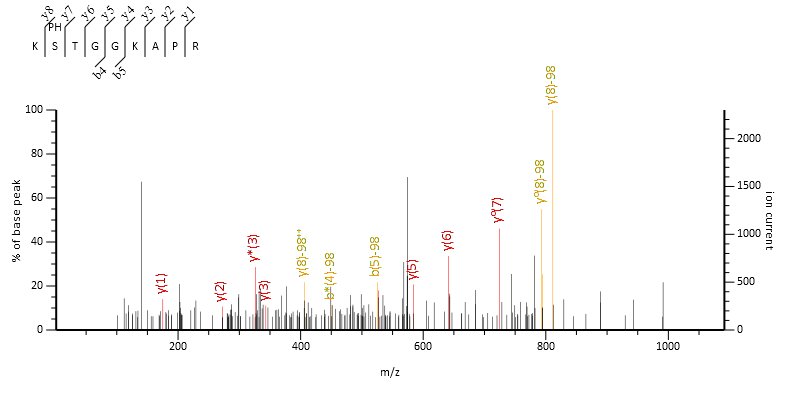
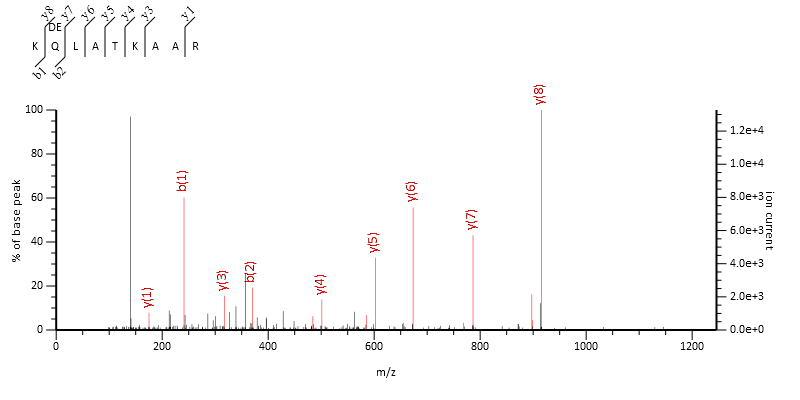
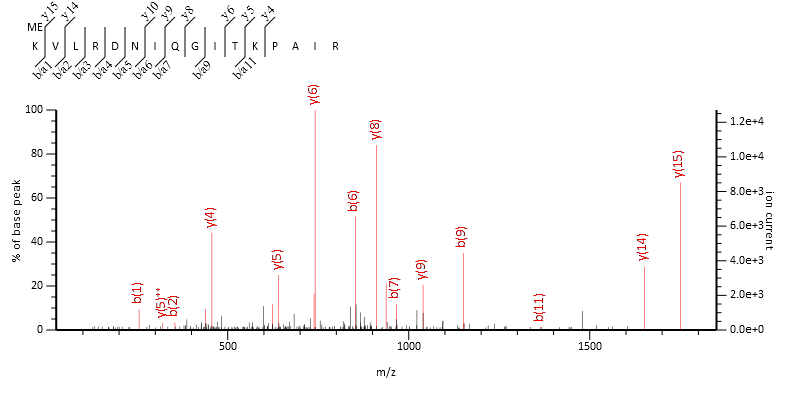
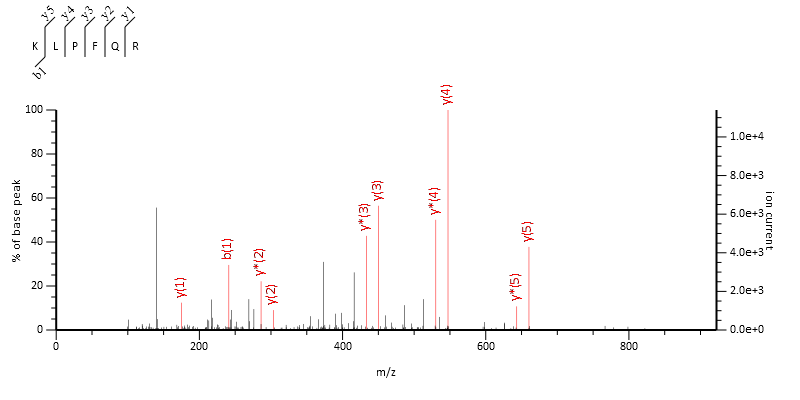
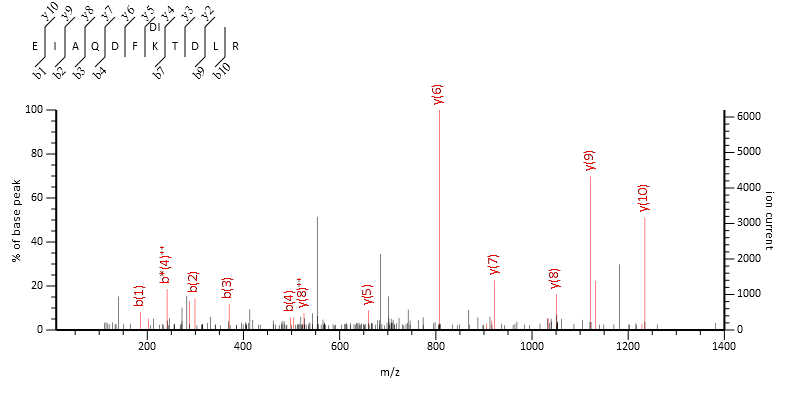
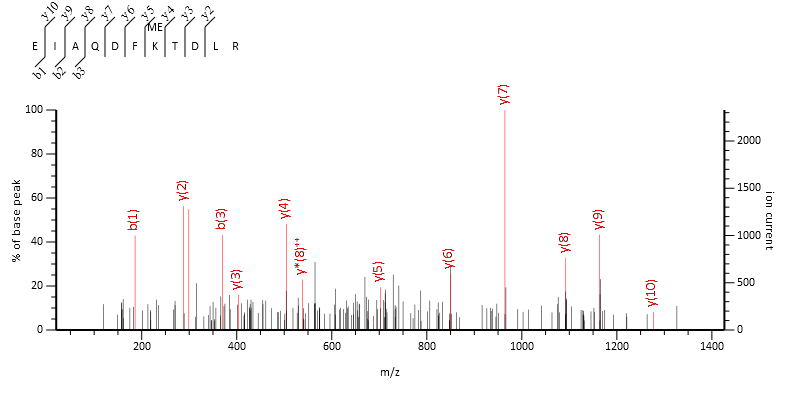
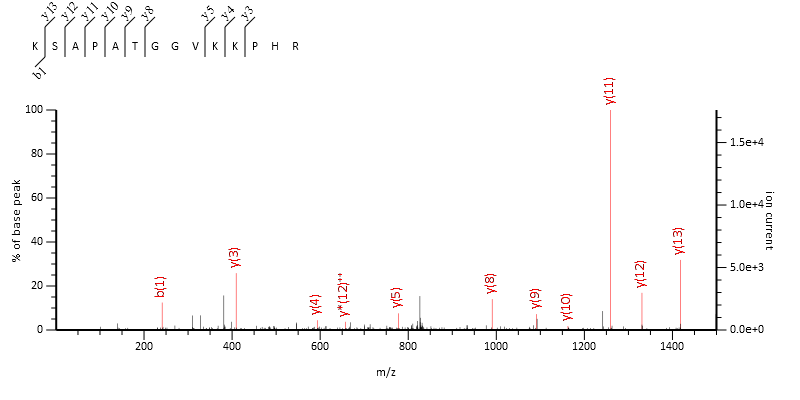
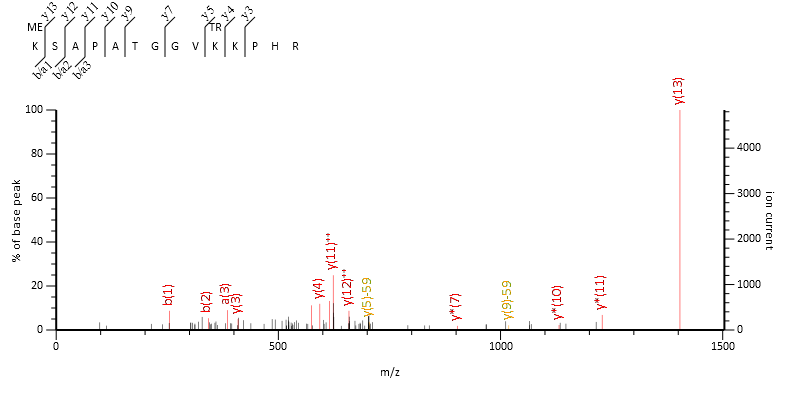
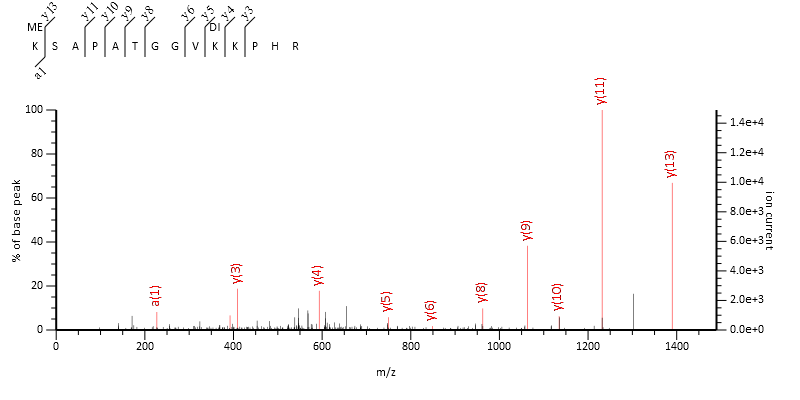
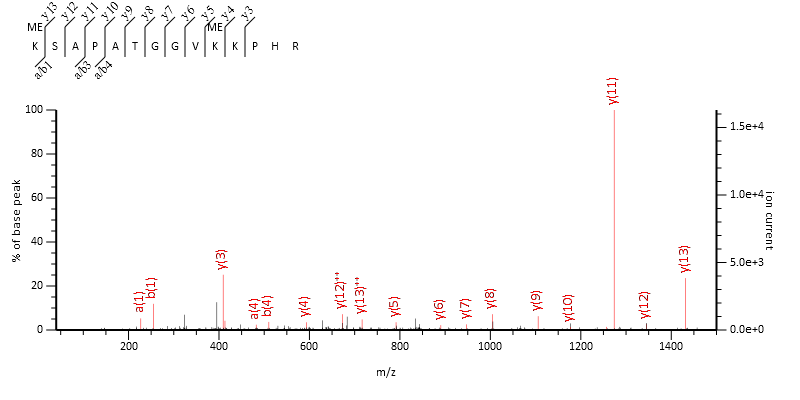
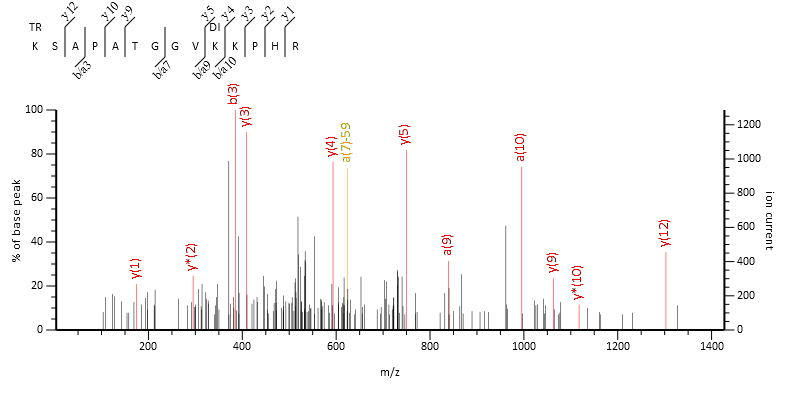
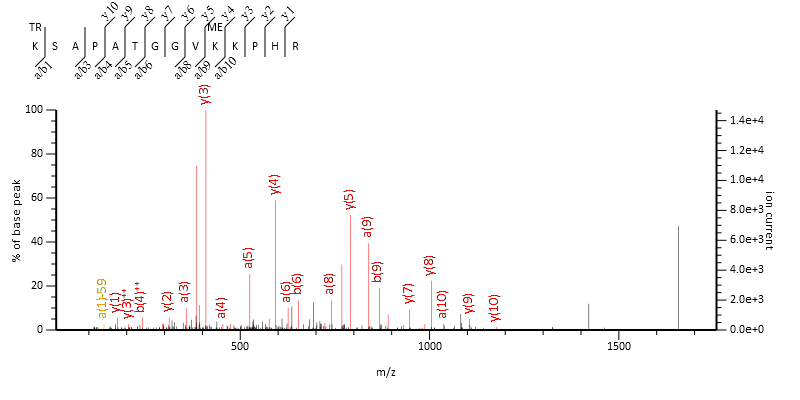
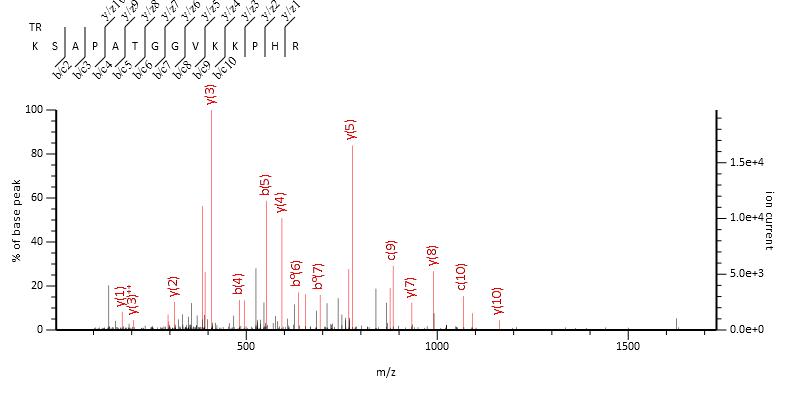
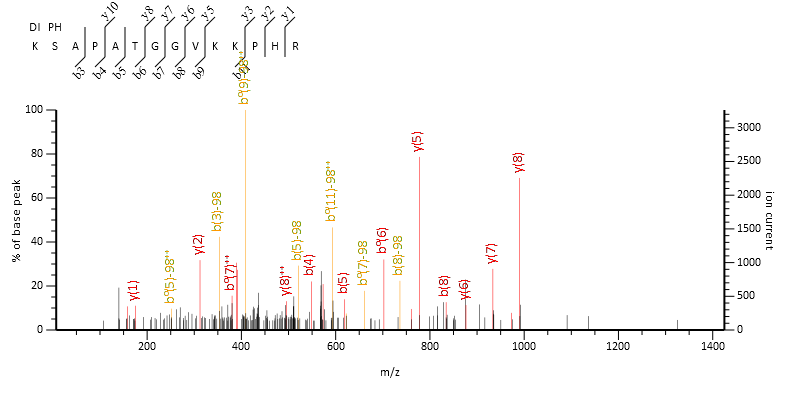
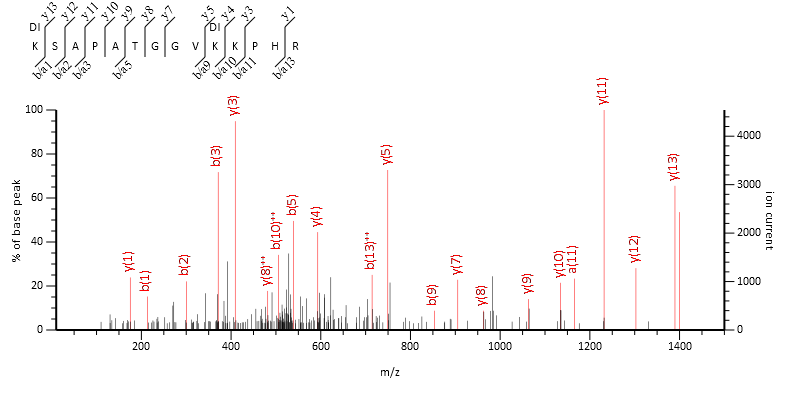
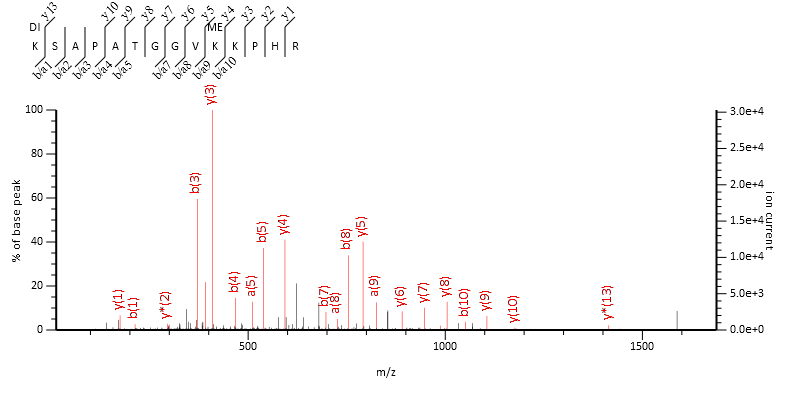
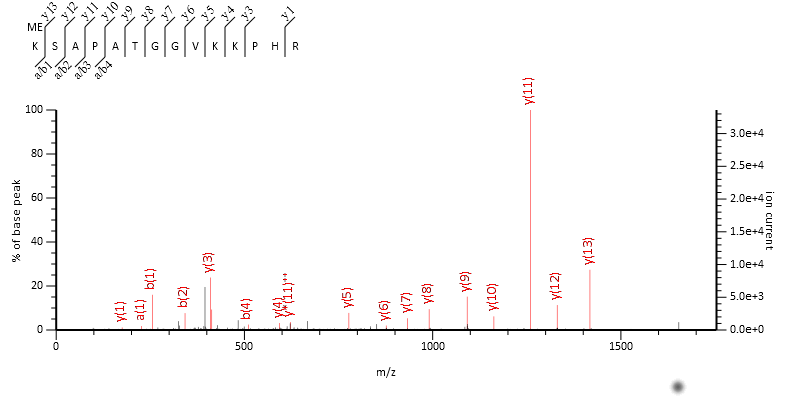
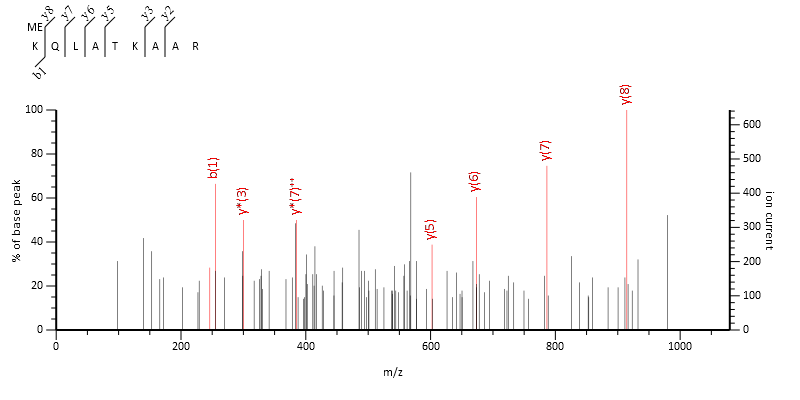
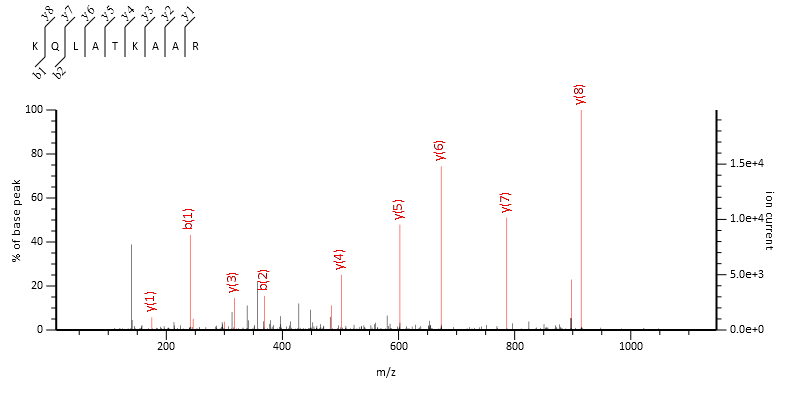
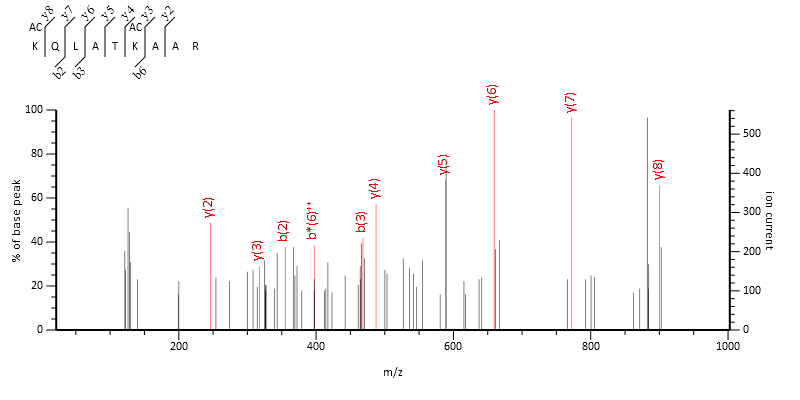
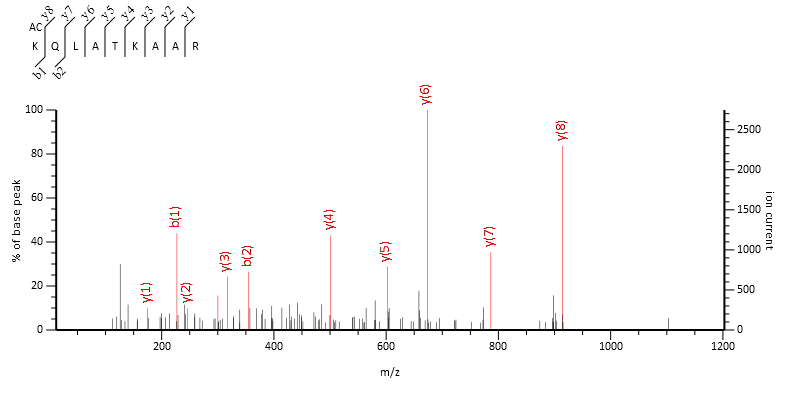
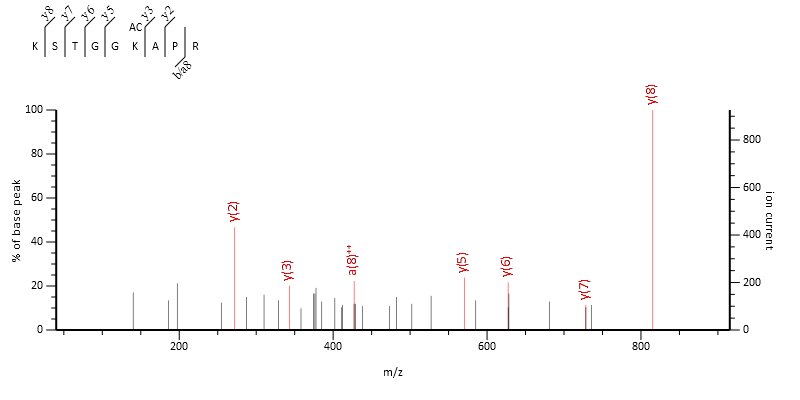
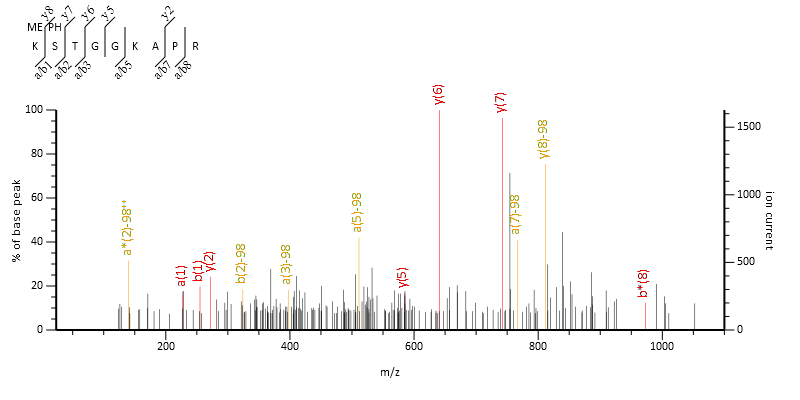
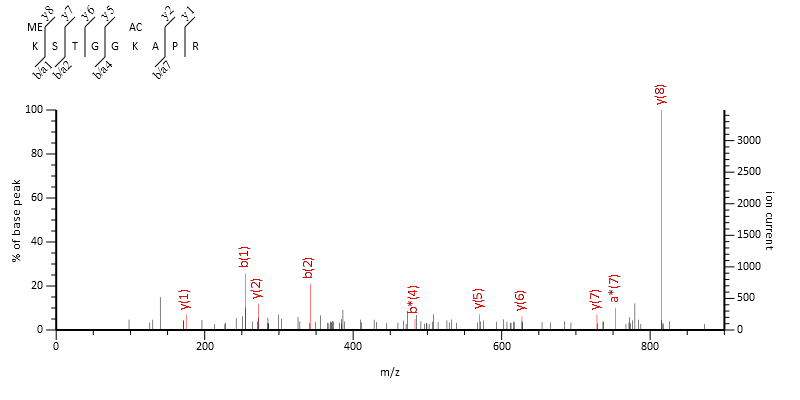
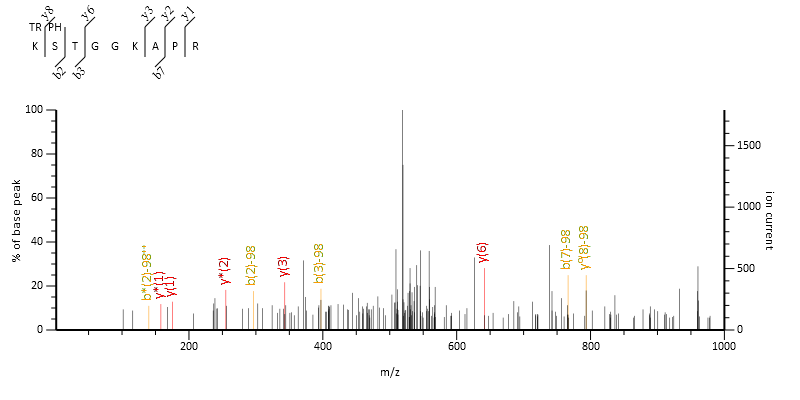
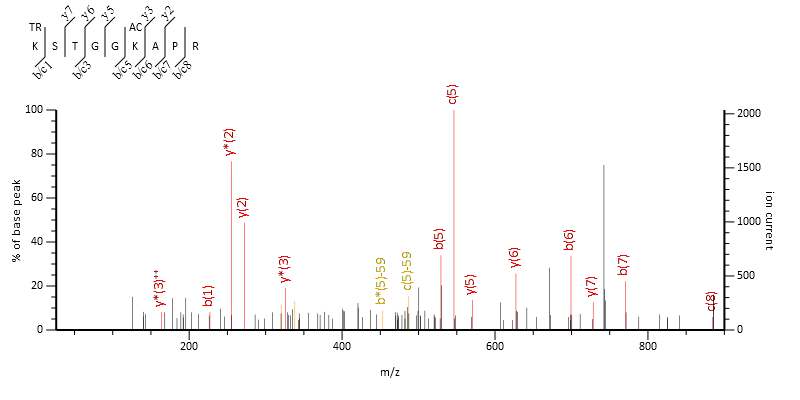
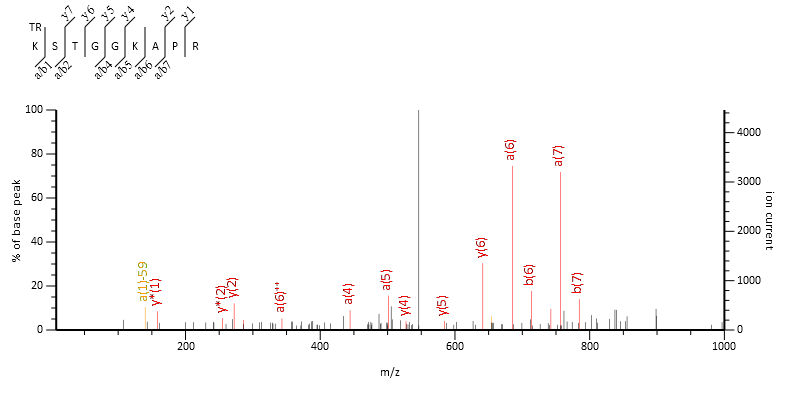
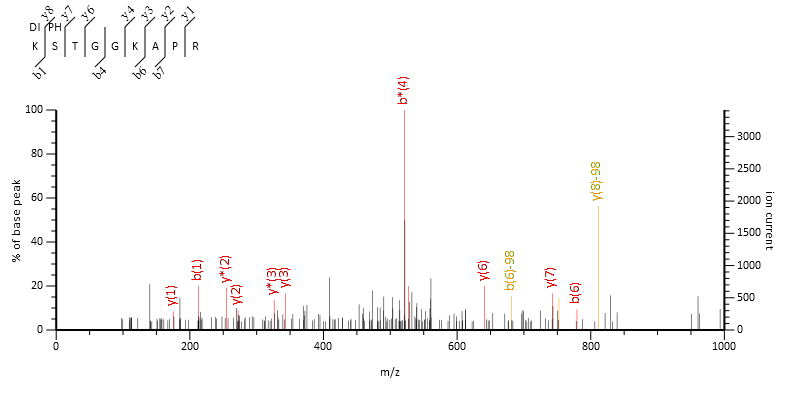
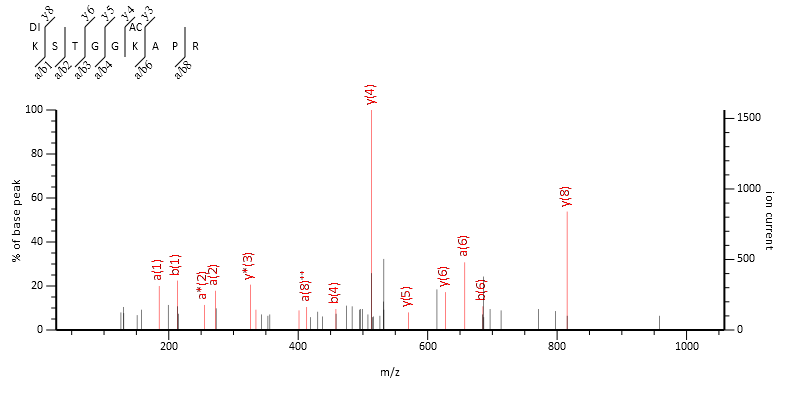
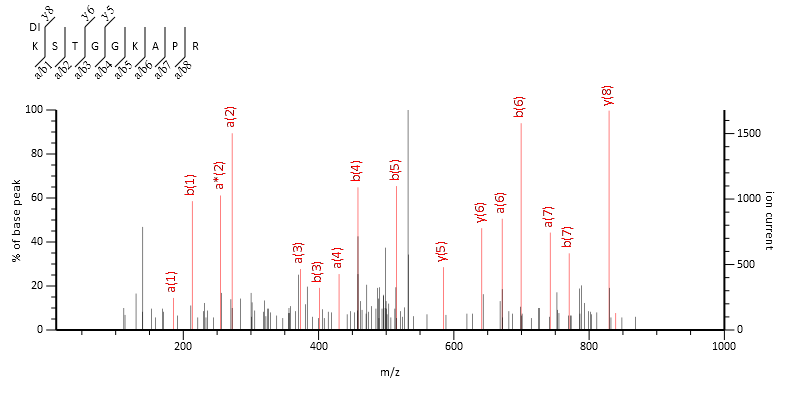
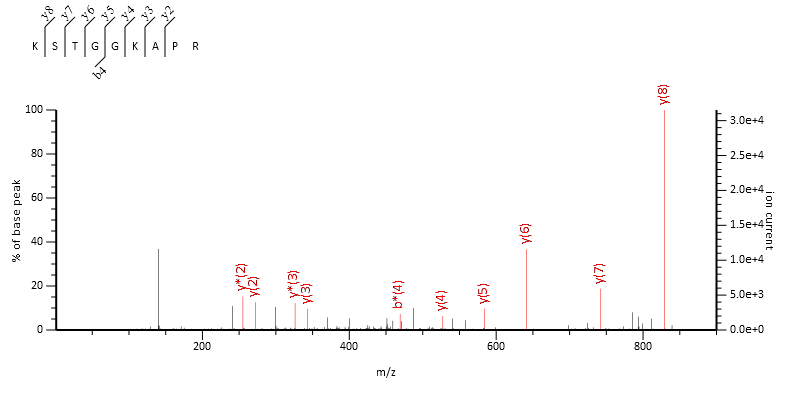
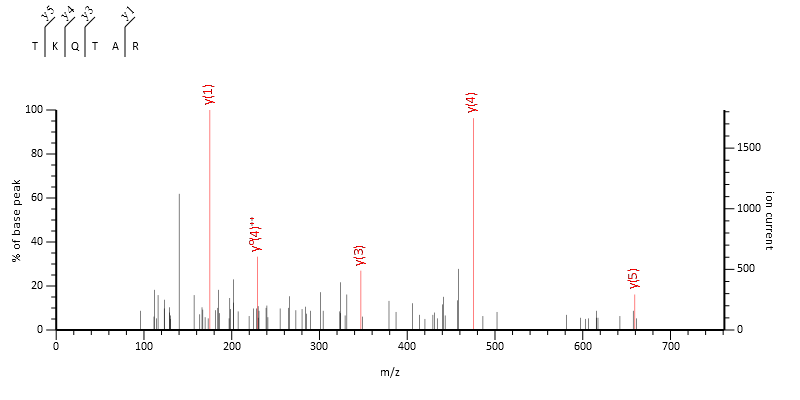
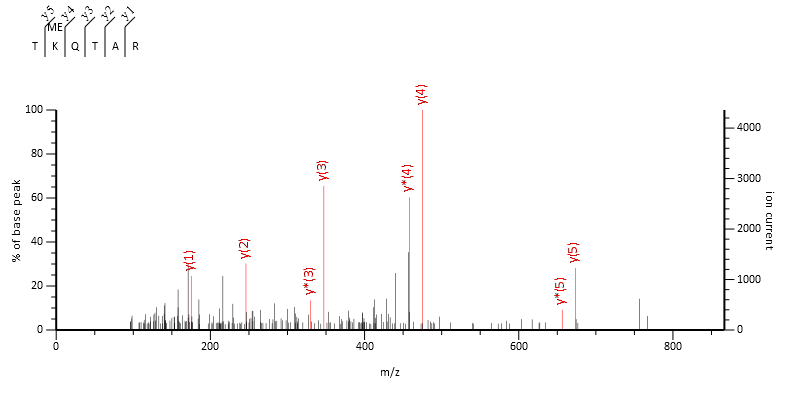
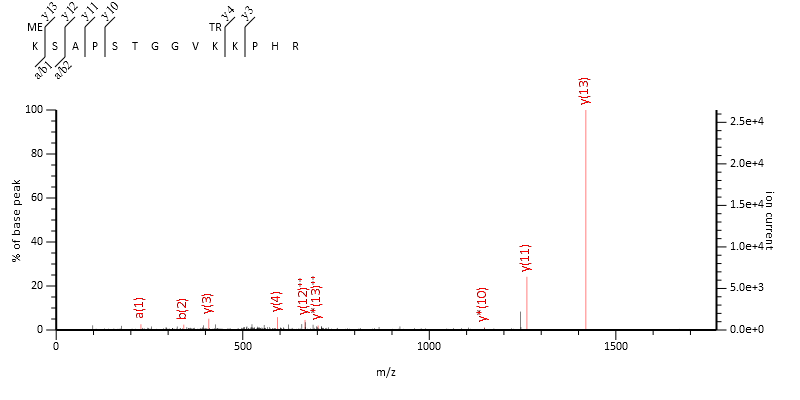
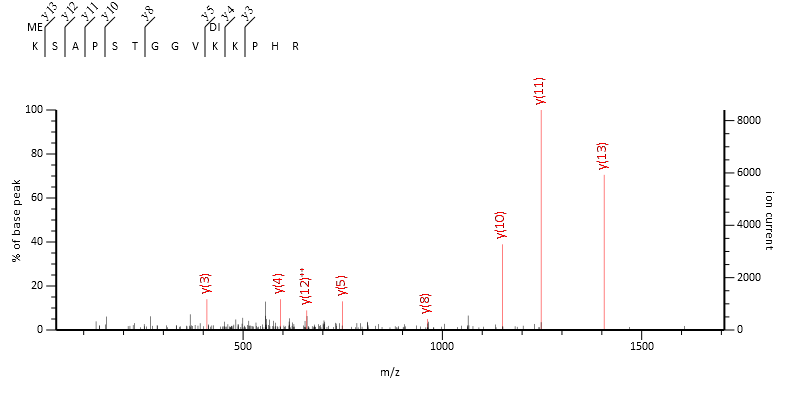
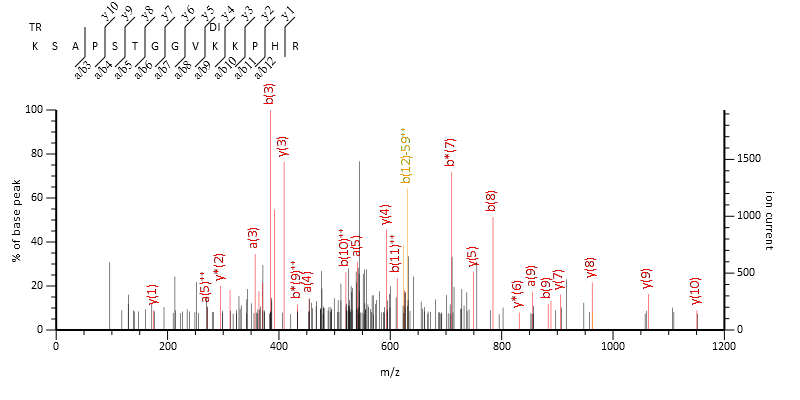
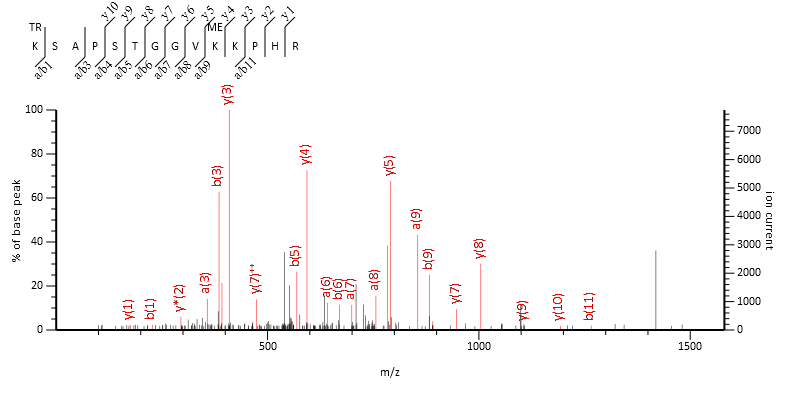
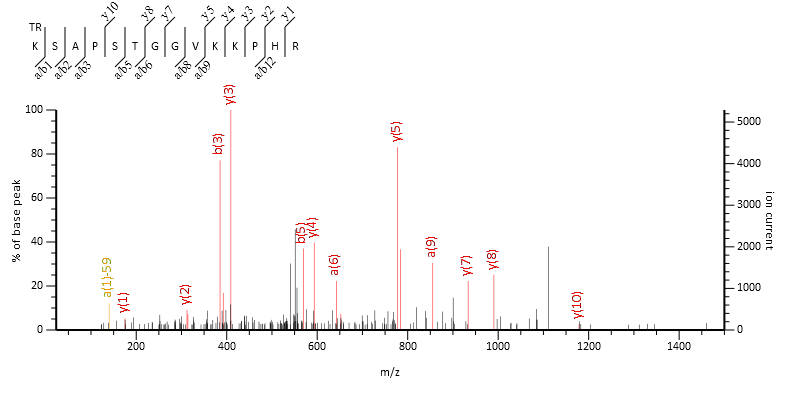
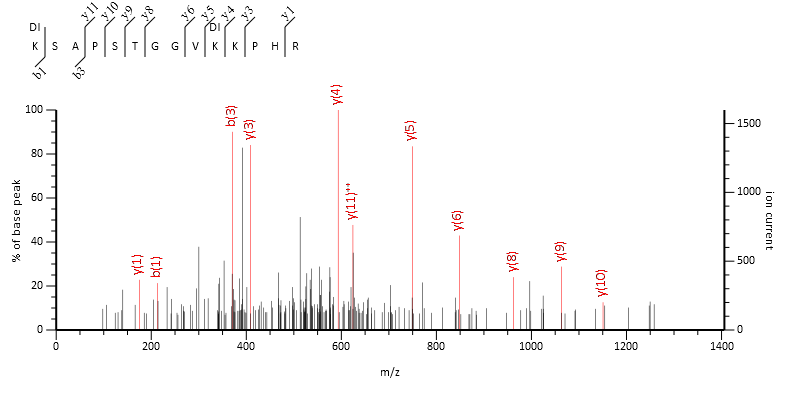
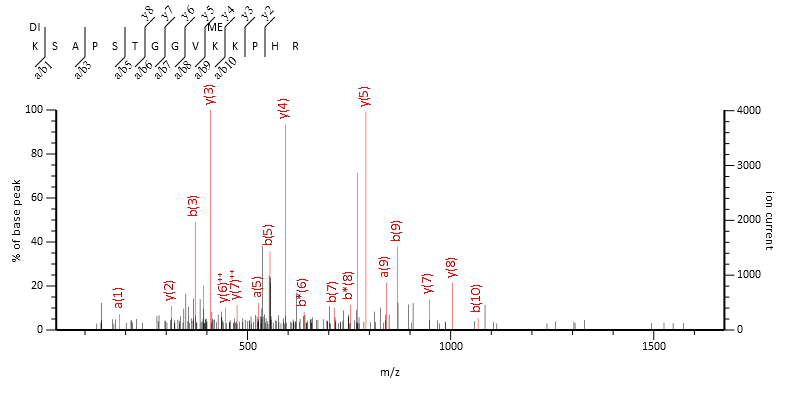
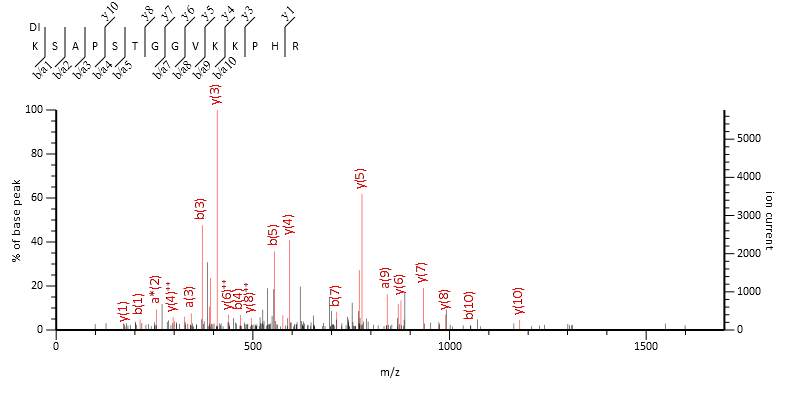
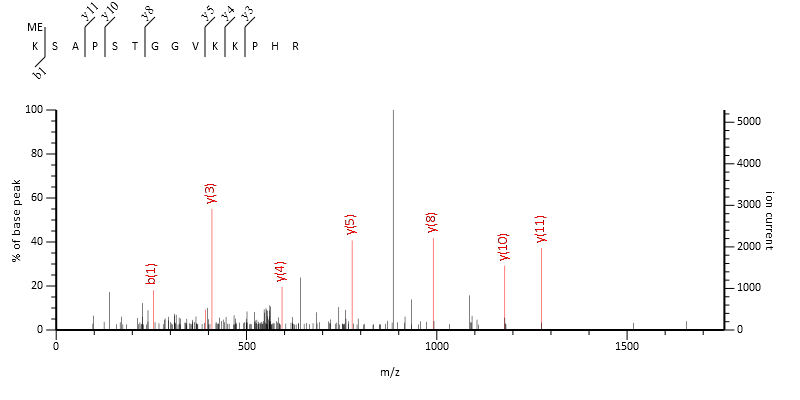
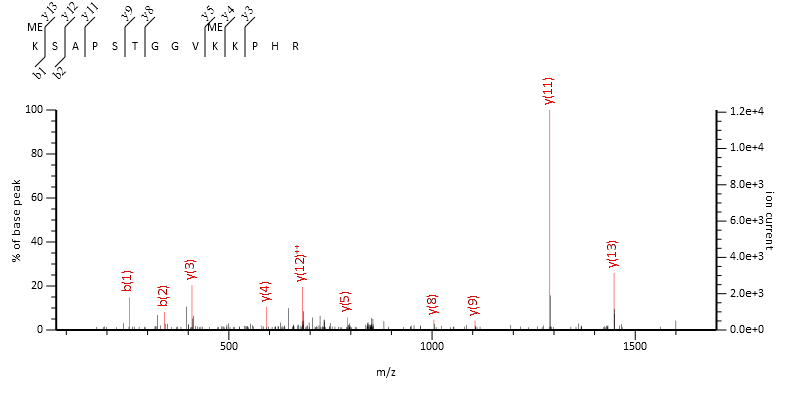
* 1. **Macrophage Histone PTMs MASCOT H3, H4, H2A, H2B, H1**

The following section displays the MASCOT output for all found and manually verified histone PTMs for Chapter 3 and 4.

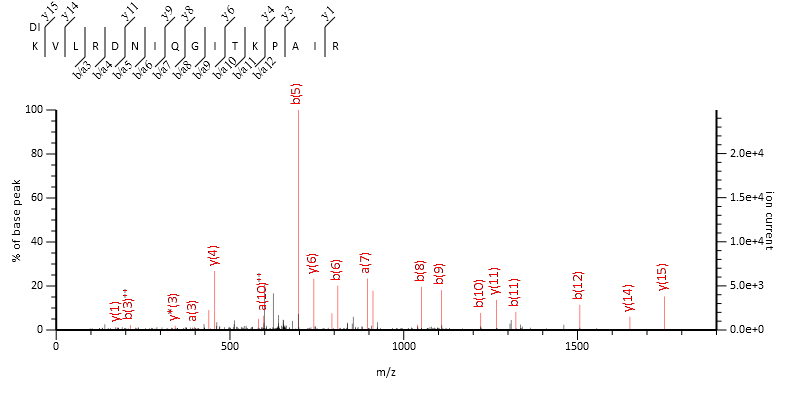
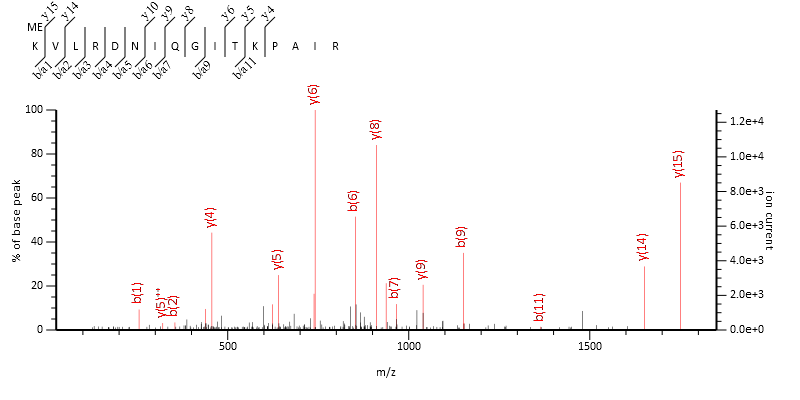
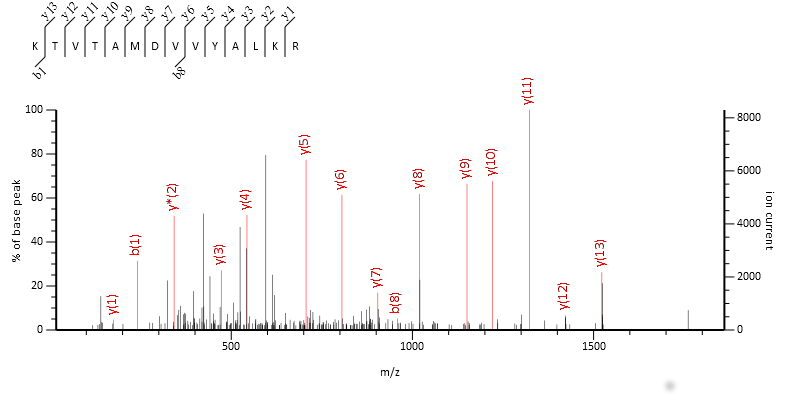
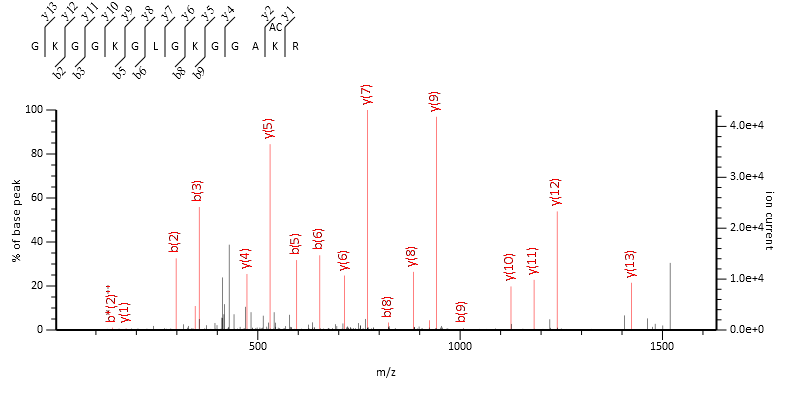
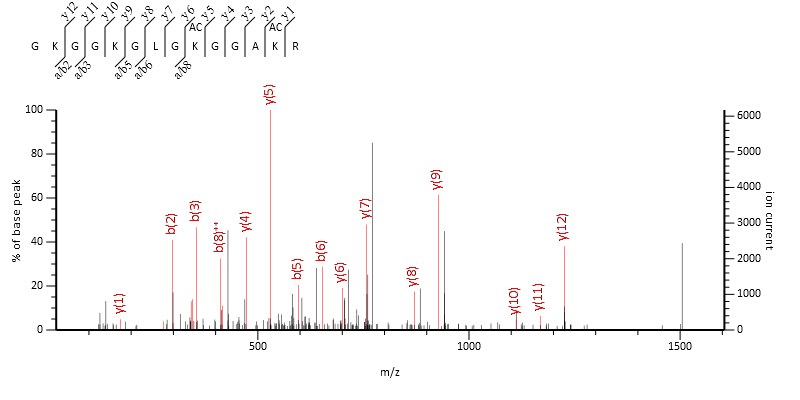
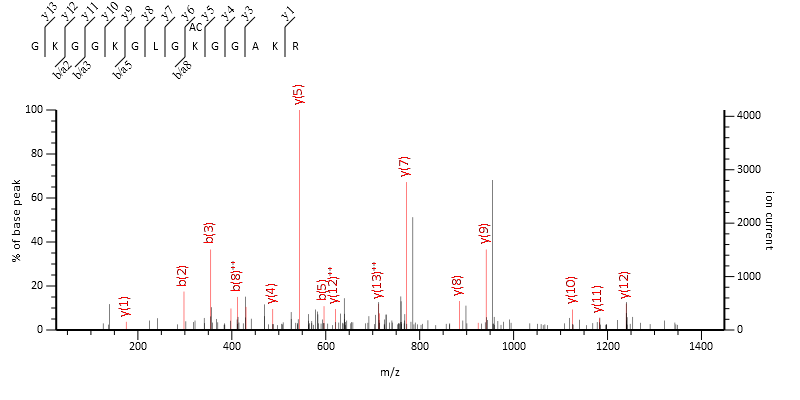
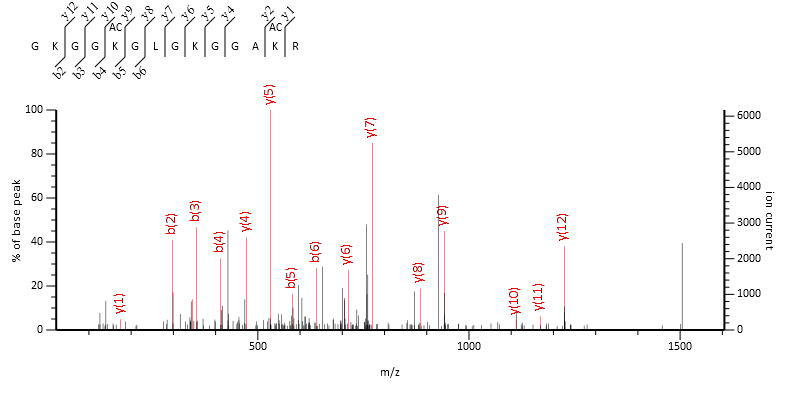
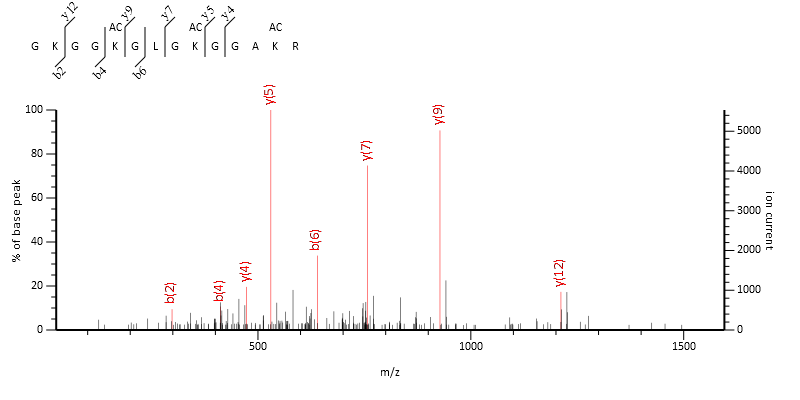
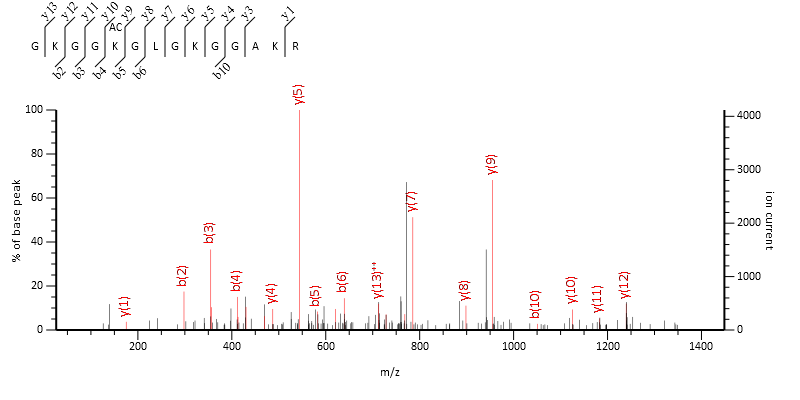
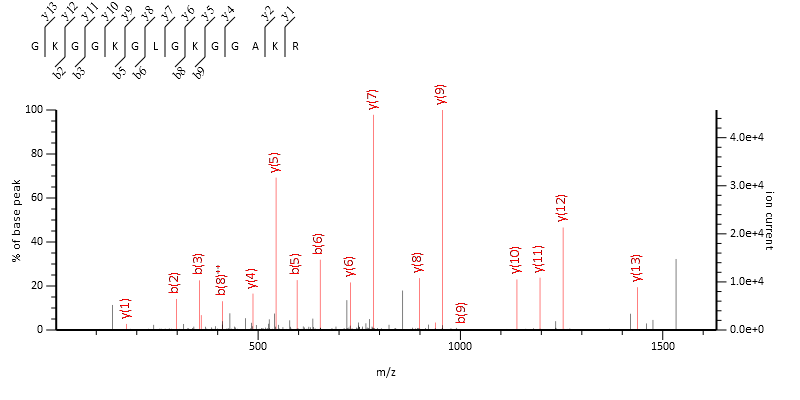
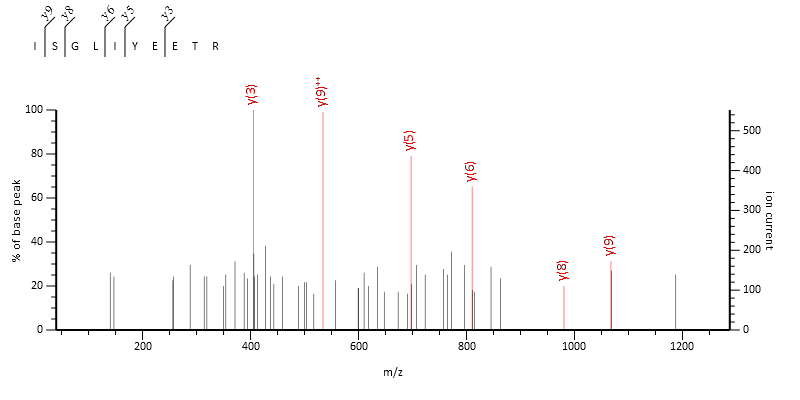
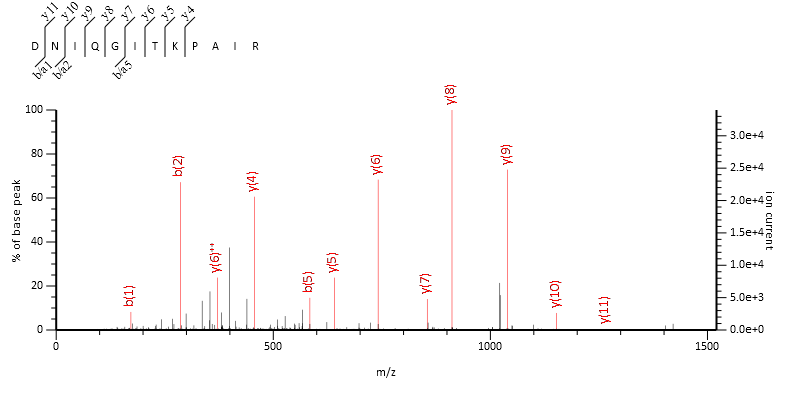
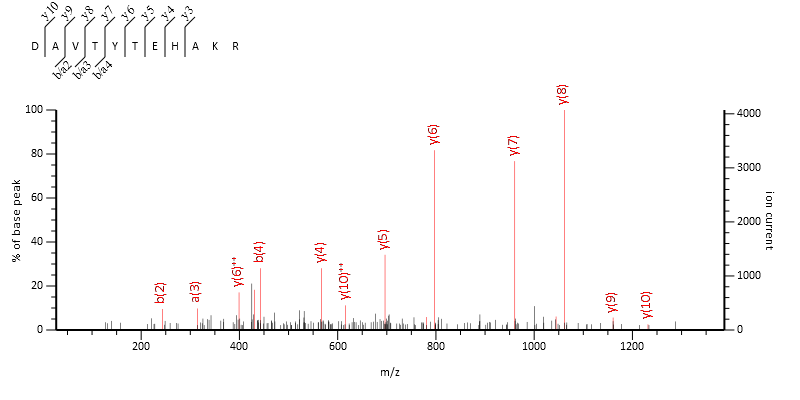
* + 1. **H2A, H2B and H1 peptides**

****

* + 1. **Histone H3.1/2 and H3.3 peptides**

****

* + 1. **Histone H4 peptides**

****

* 1. **Inclusion and exclusion criteria for the septic patients and surgical controls in chapter 5**

Inclusion Criteria

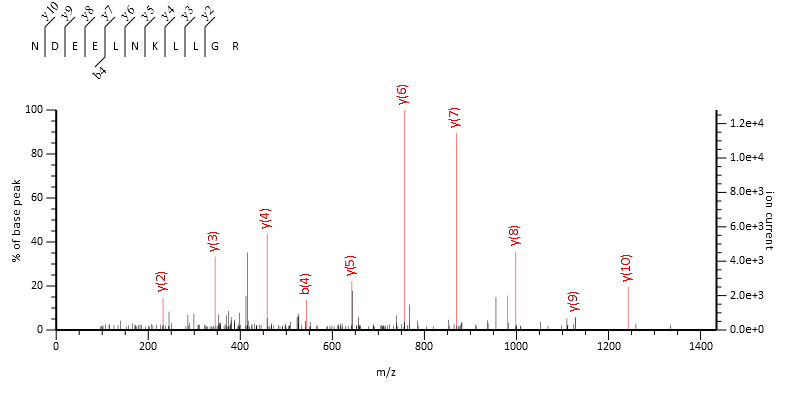
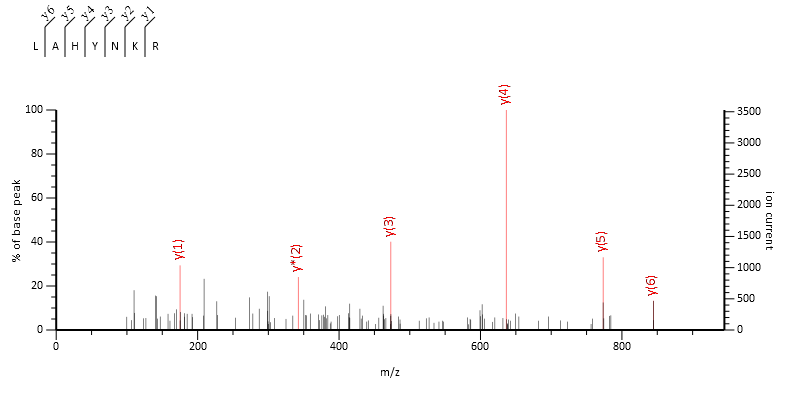
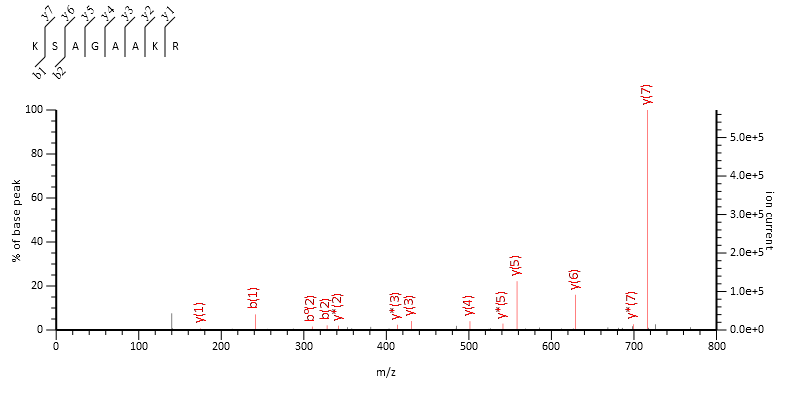
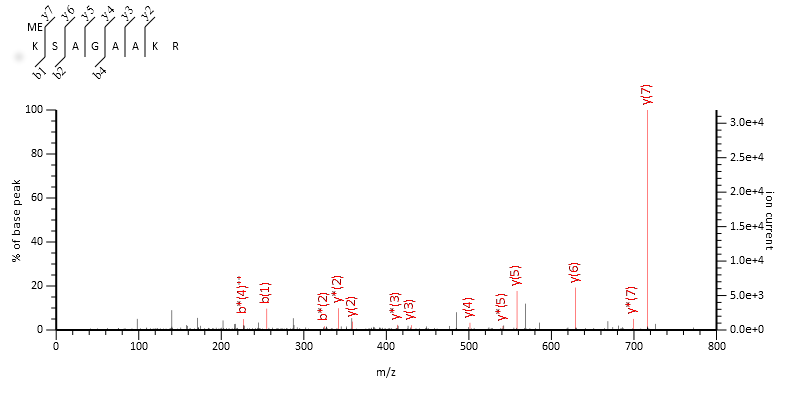
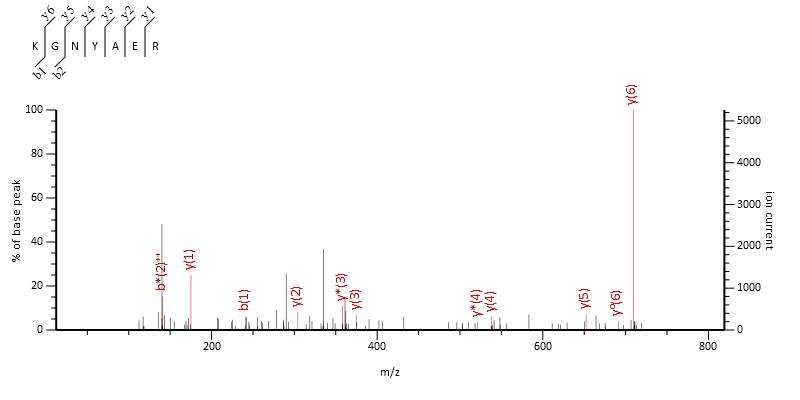
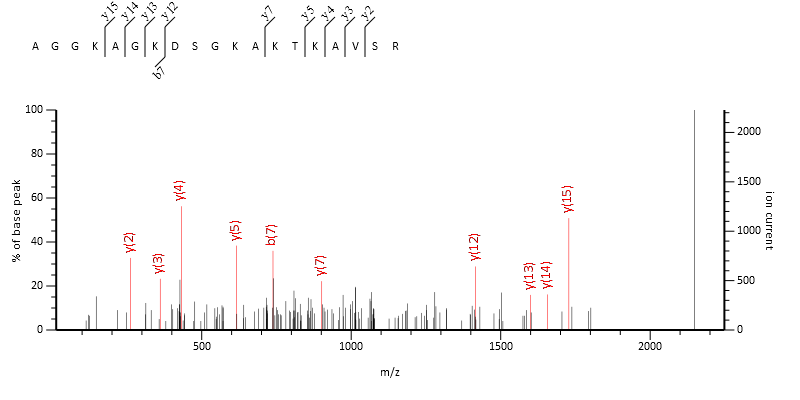
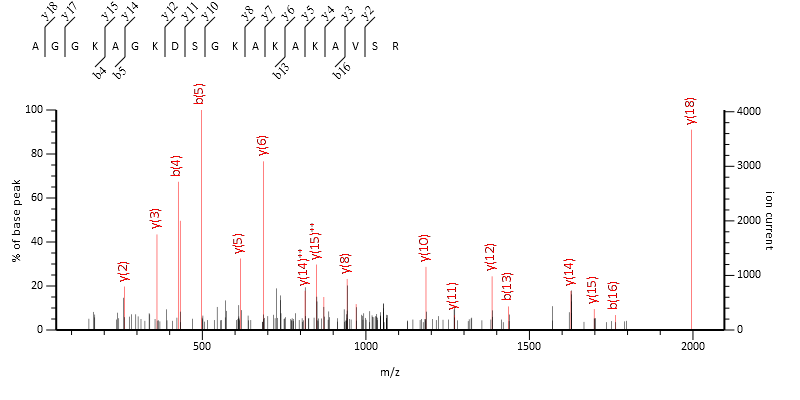
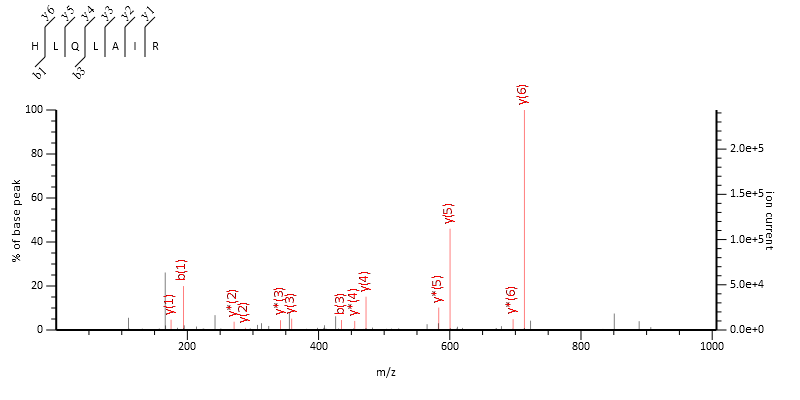
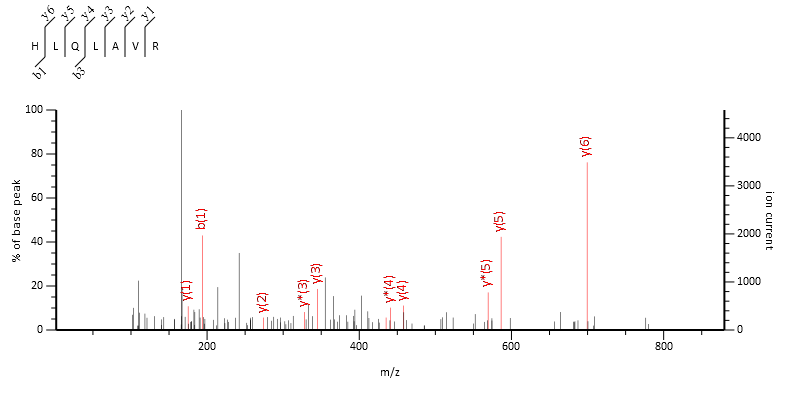
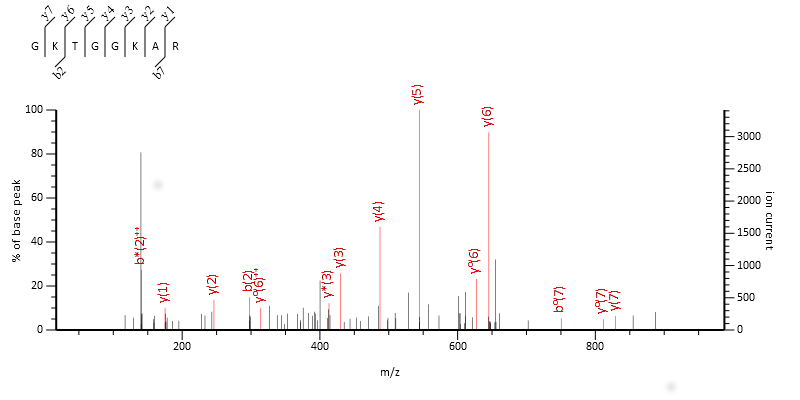
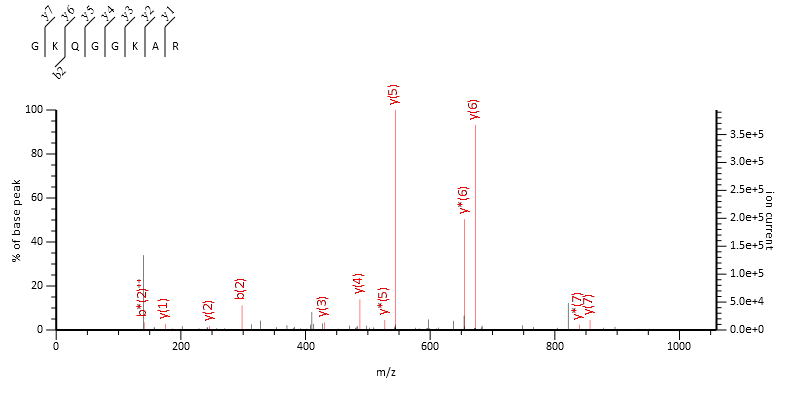
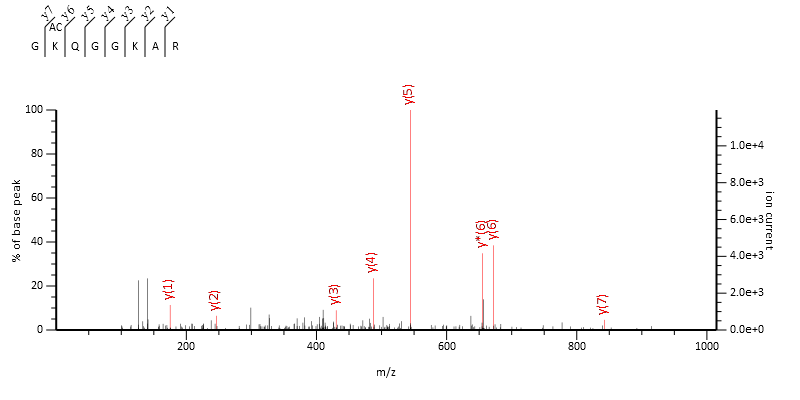
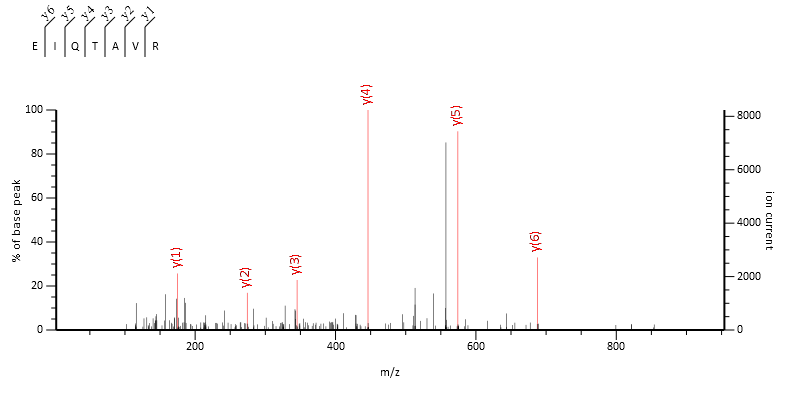
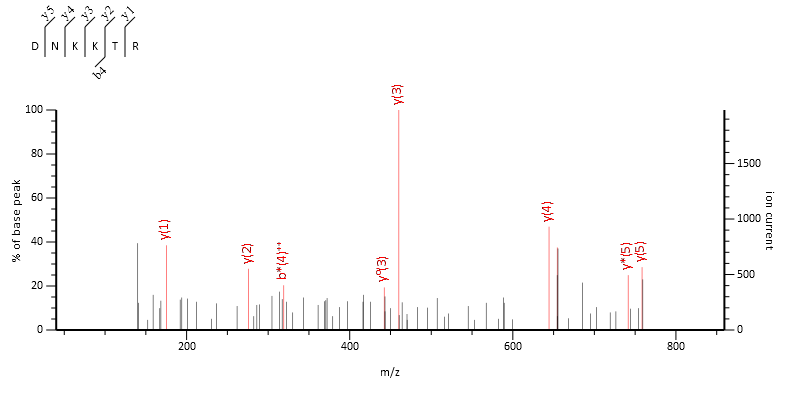
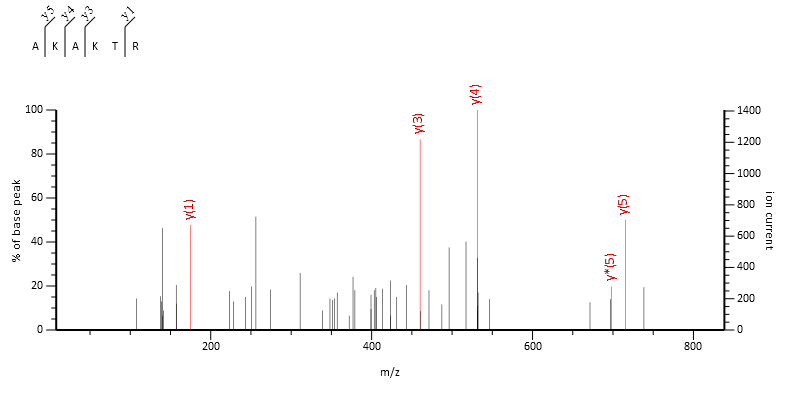
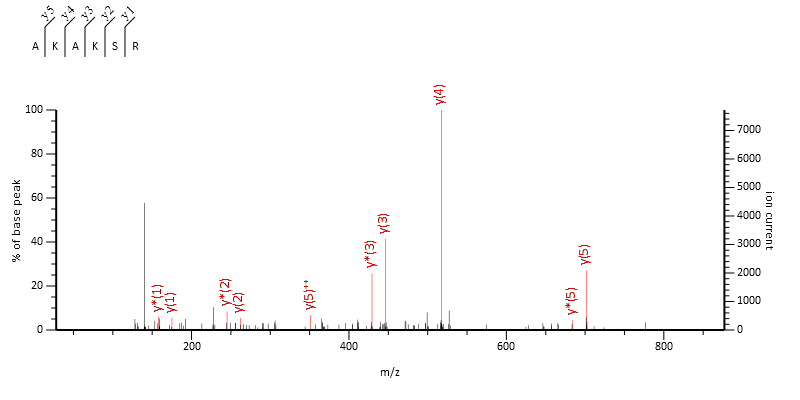
1. Aged 18 years and over
2. Informed consent / assent from next of kin in septic participants lacking capacity
3. Faecal peritonitis confirmed radiologically or surgically for severe sepsis participants, post-operative abdominal surgery for control participants
4. Severe sepsis (septic participants), or post abdominal surgery but without sepsis (control participants)

Exclusion Criteria

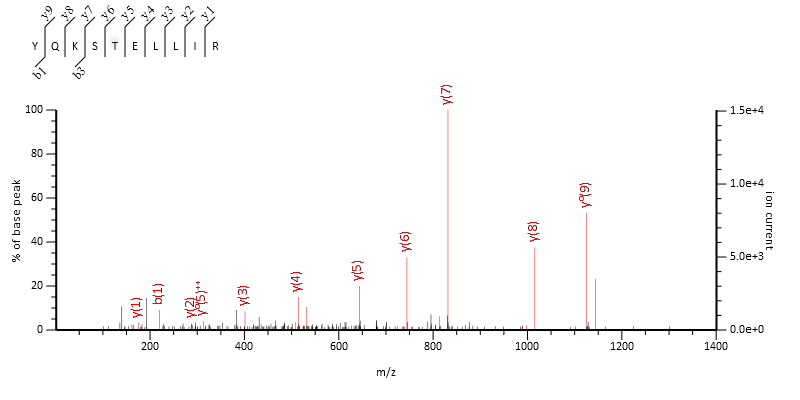
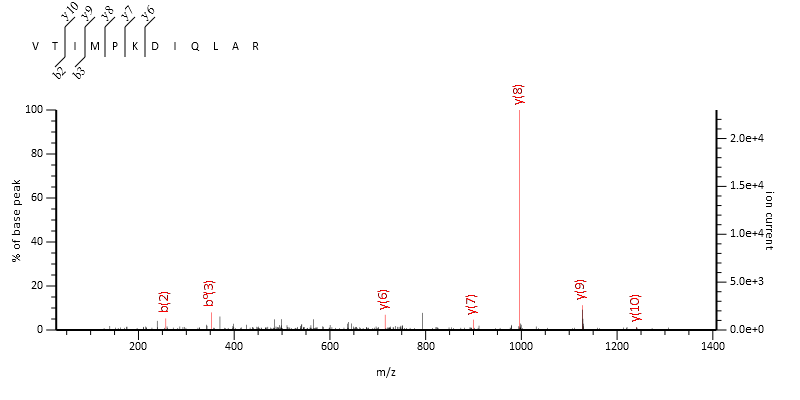
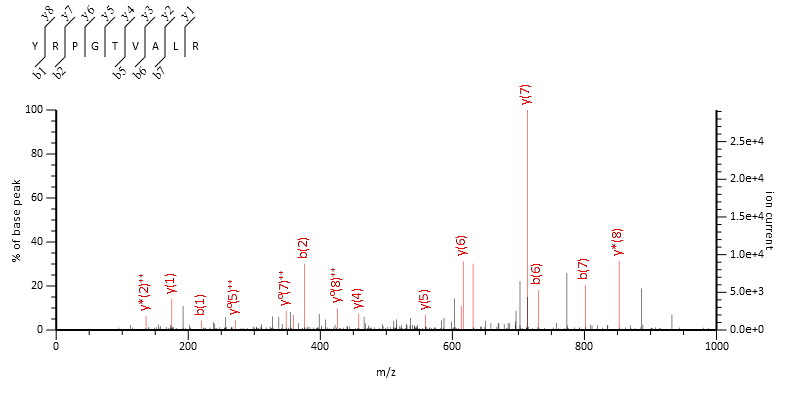
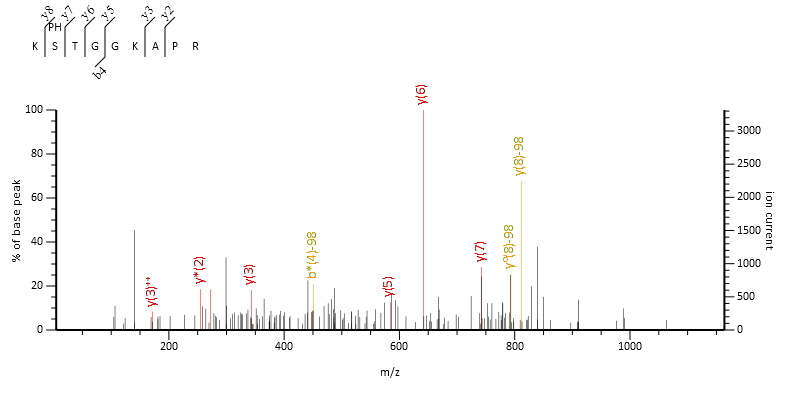
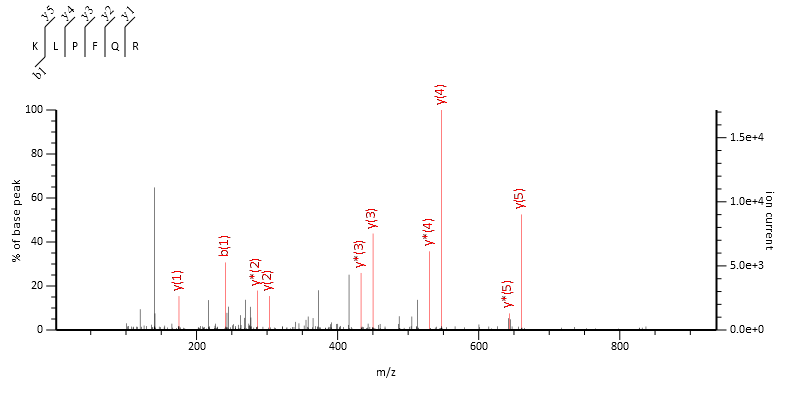
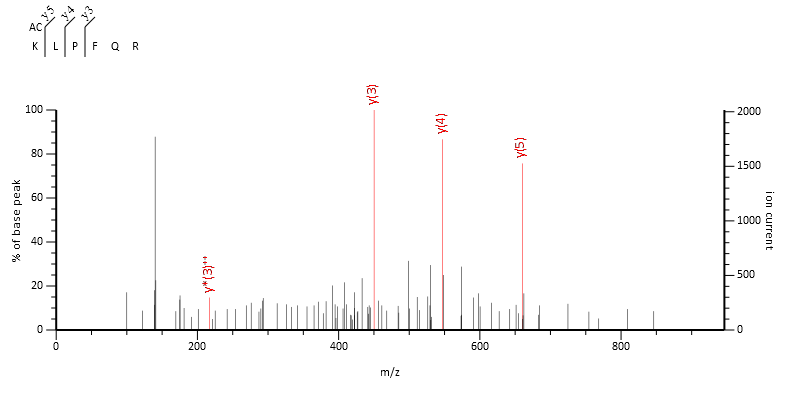
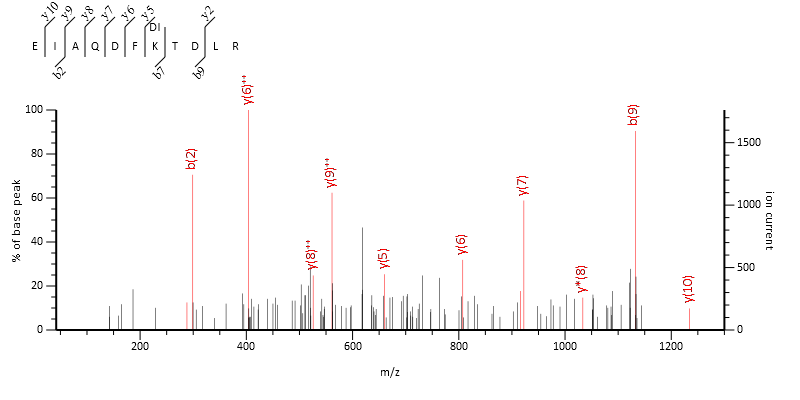
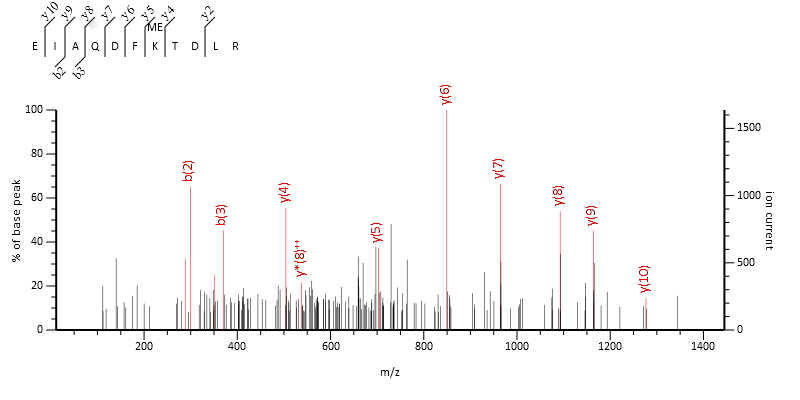
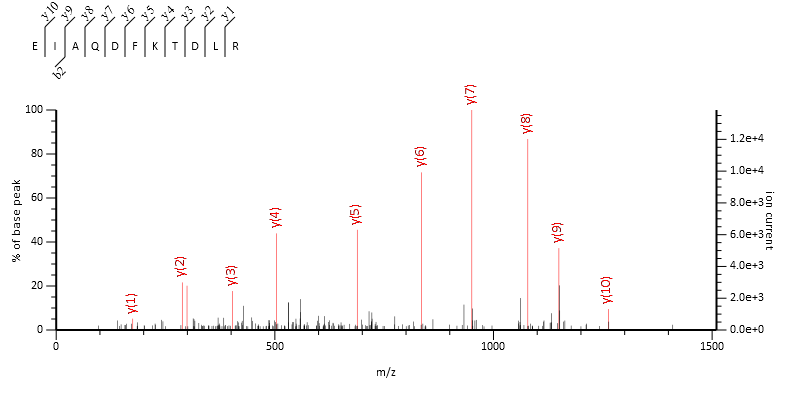
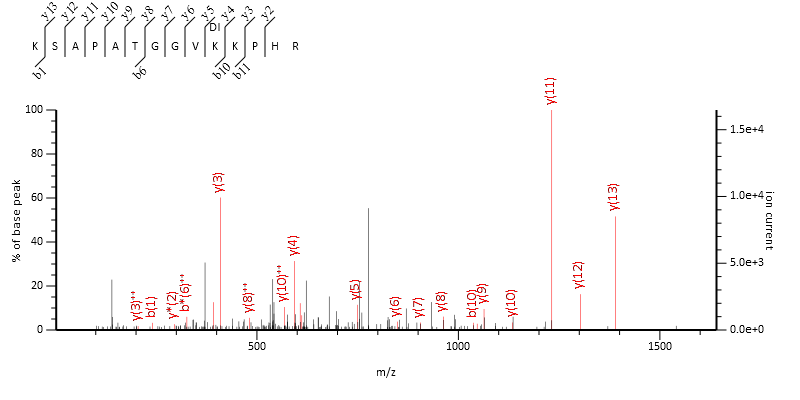
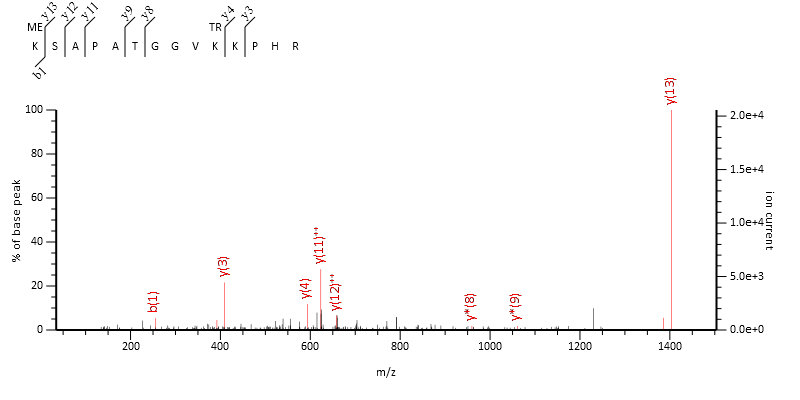
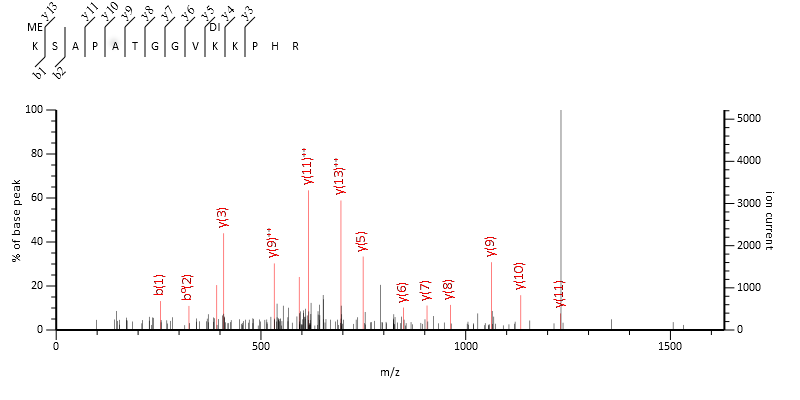
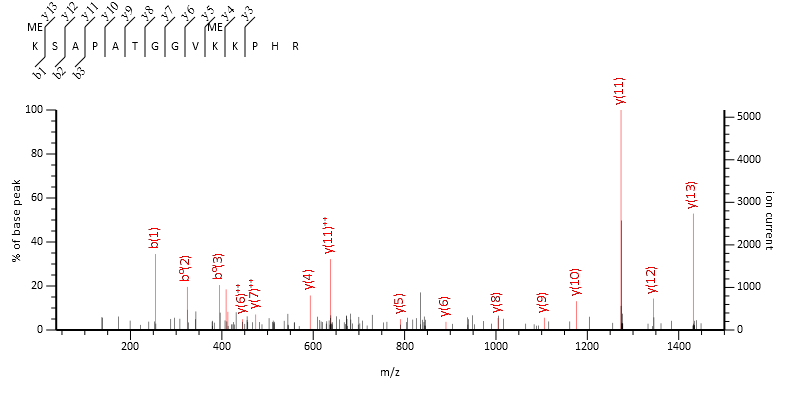
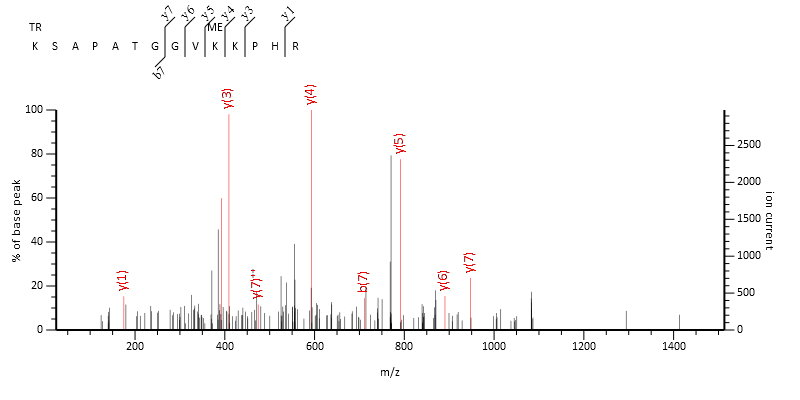
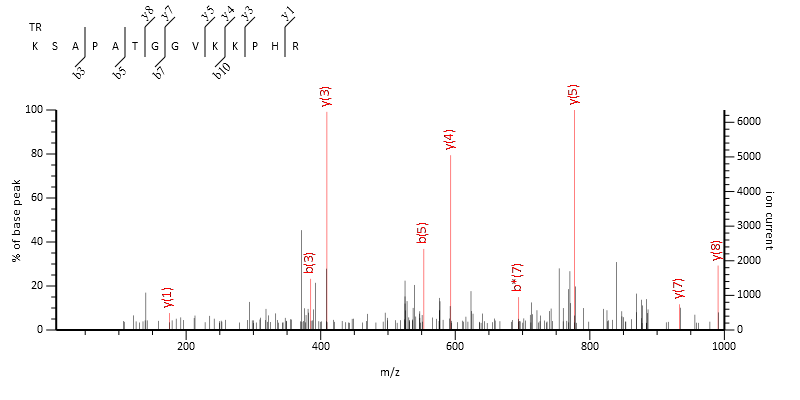
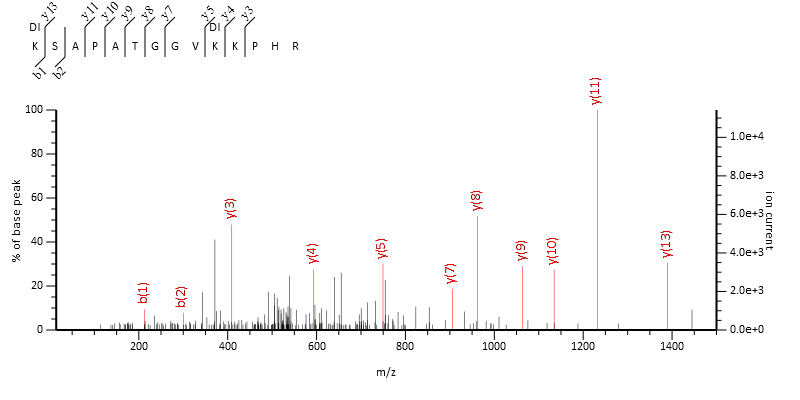
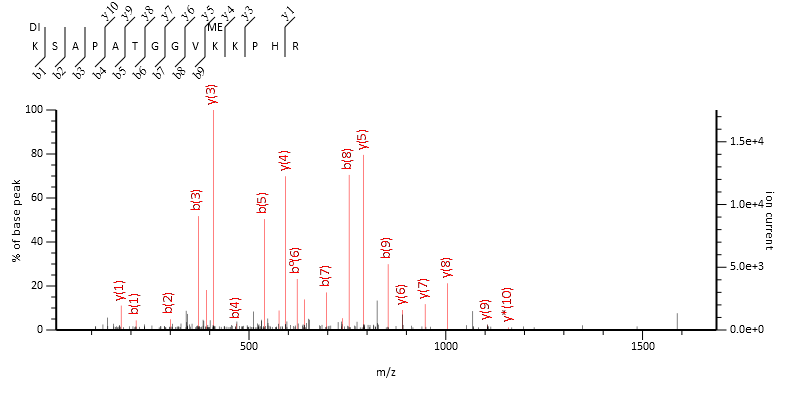
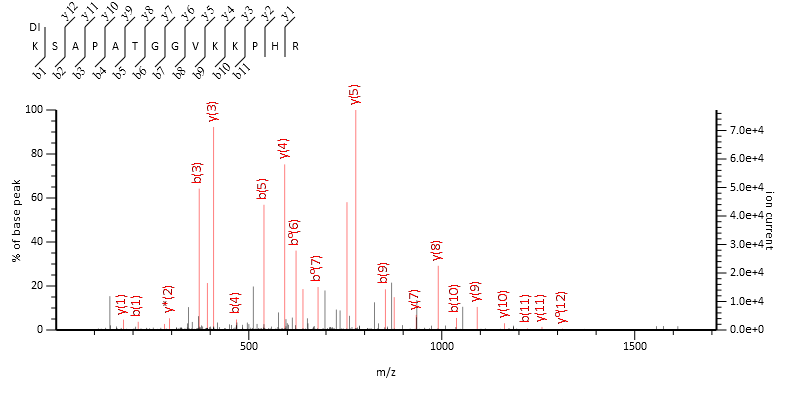
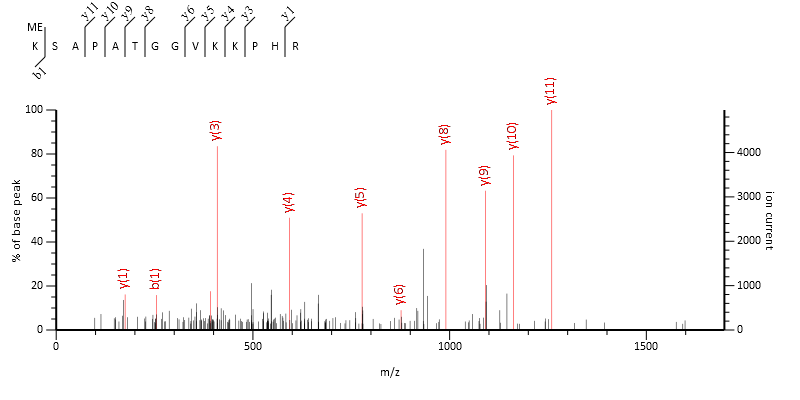
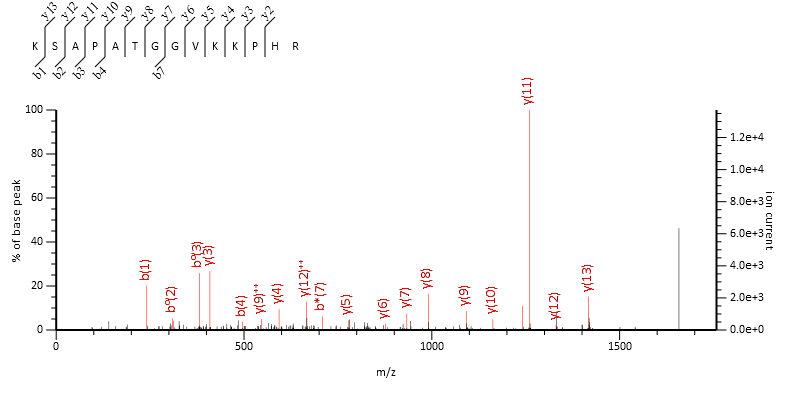
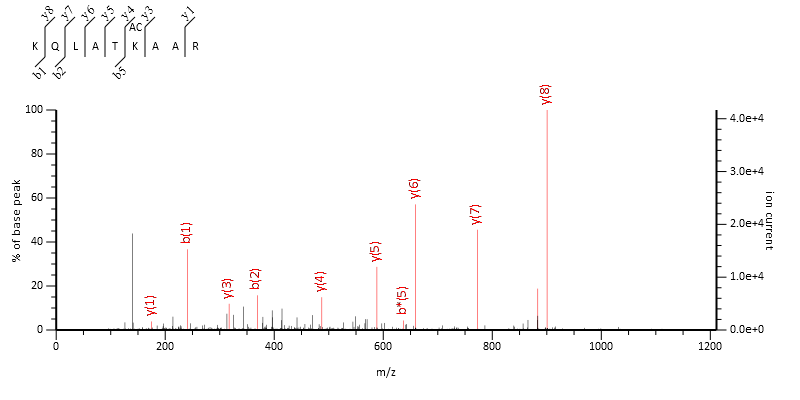
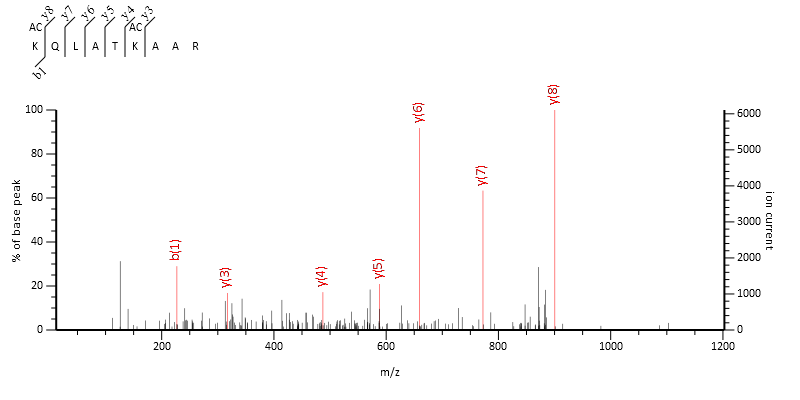
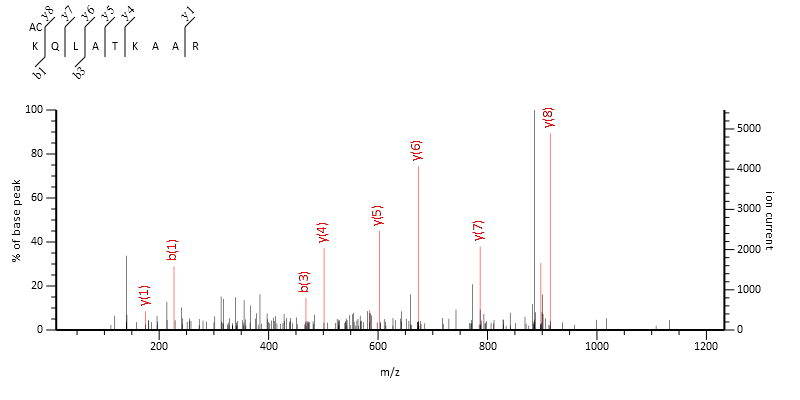
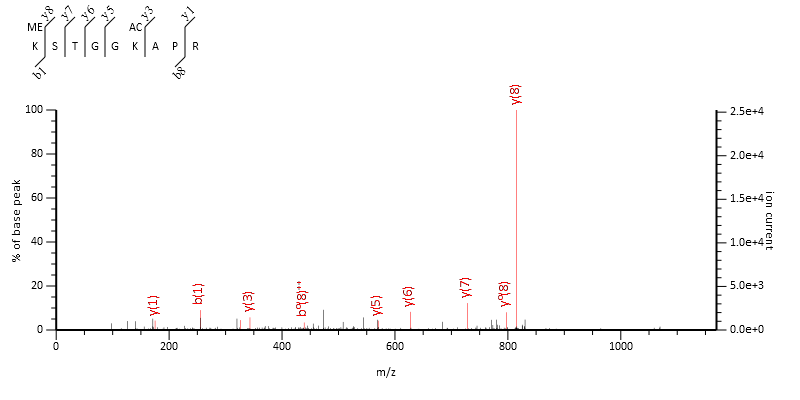
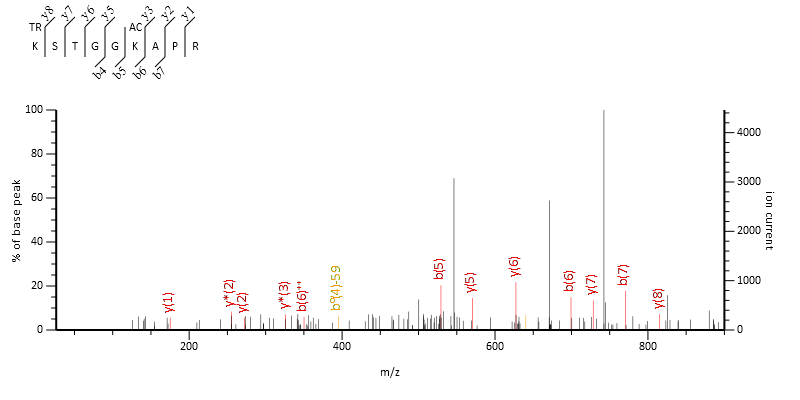
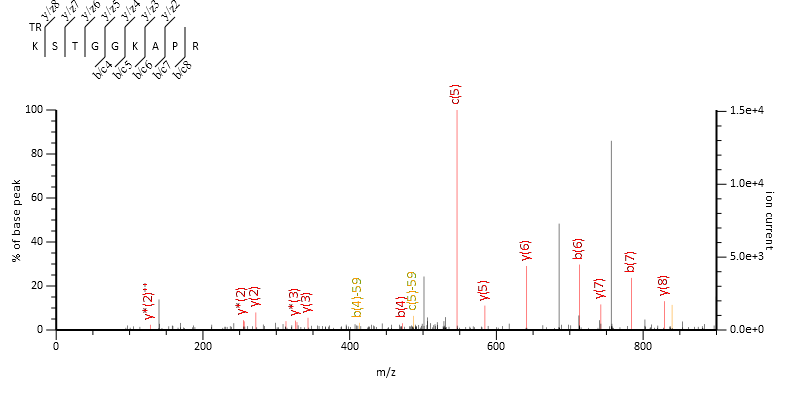
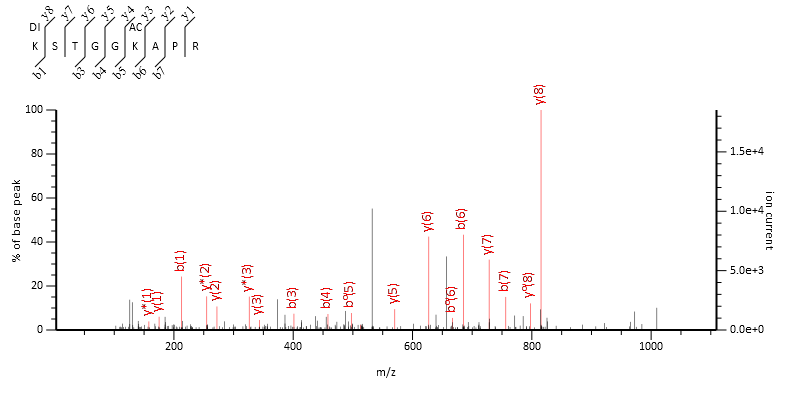
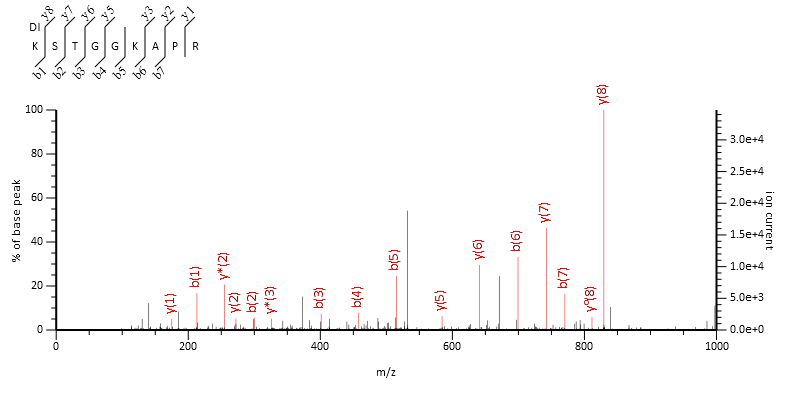
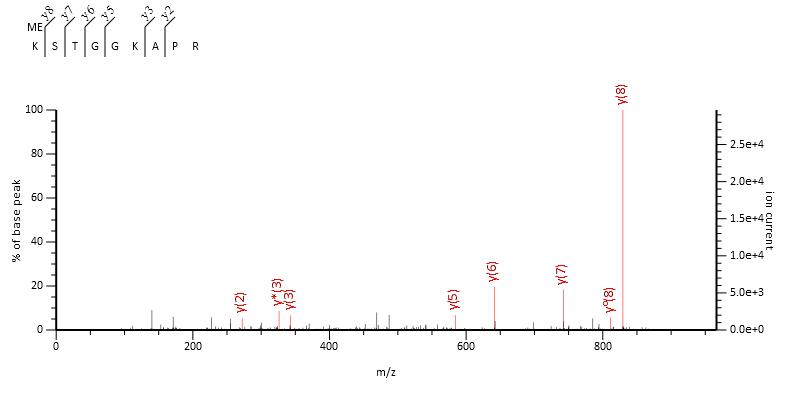
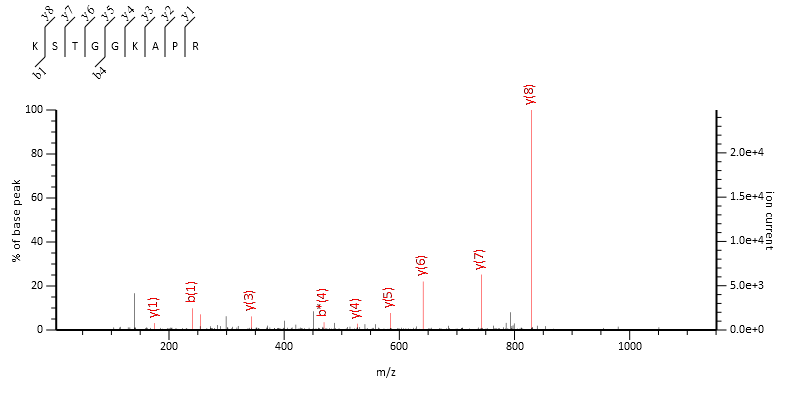
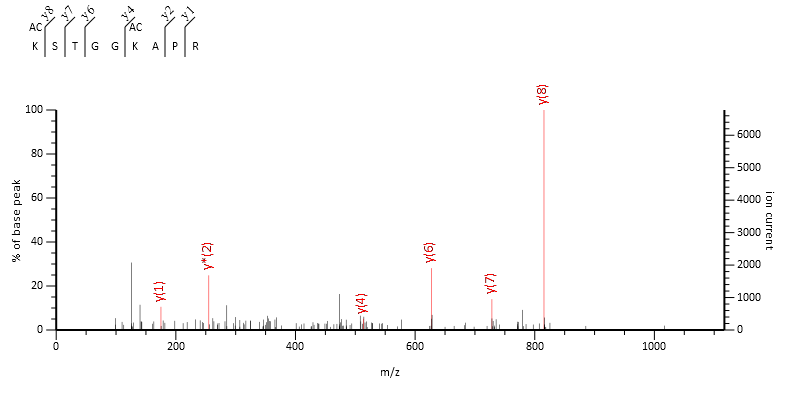
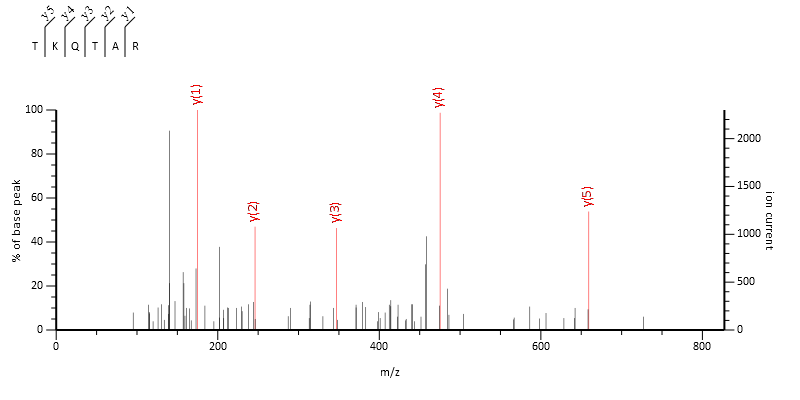
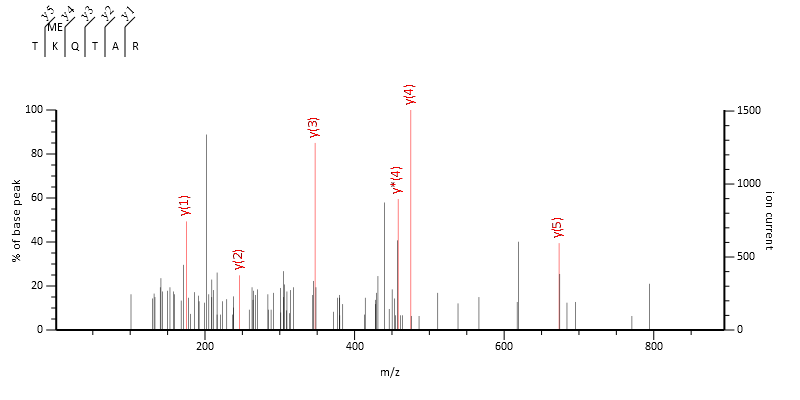
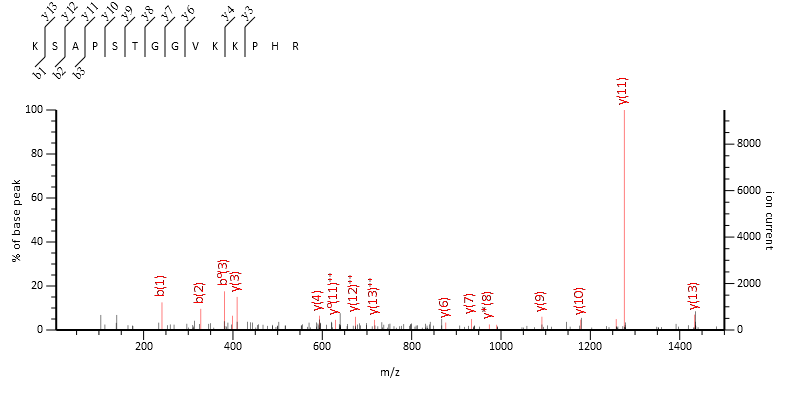
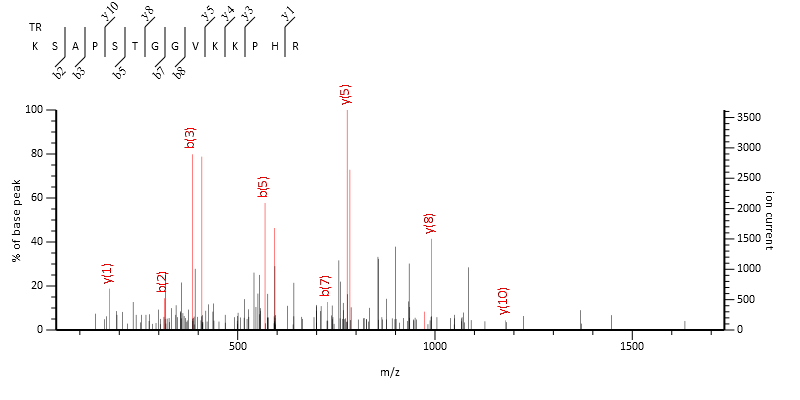
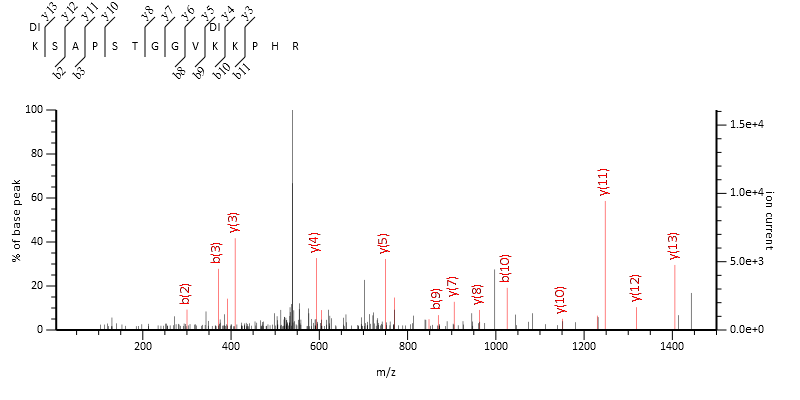
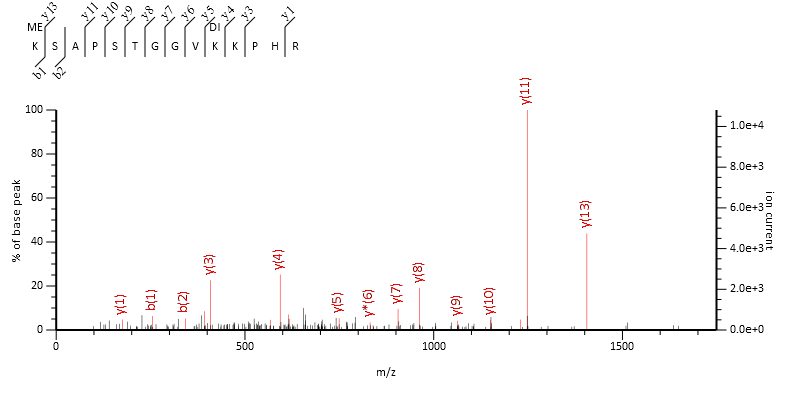
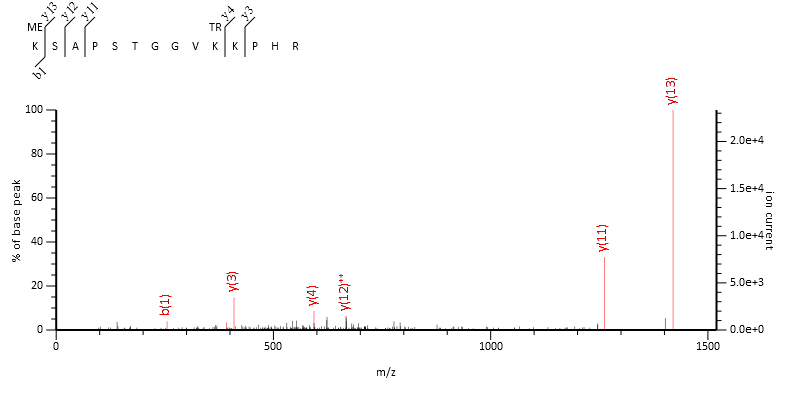
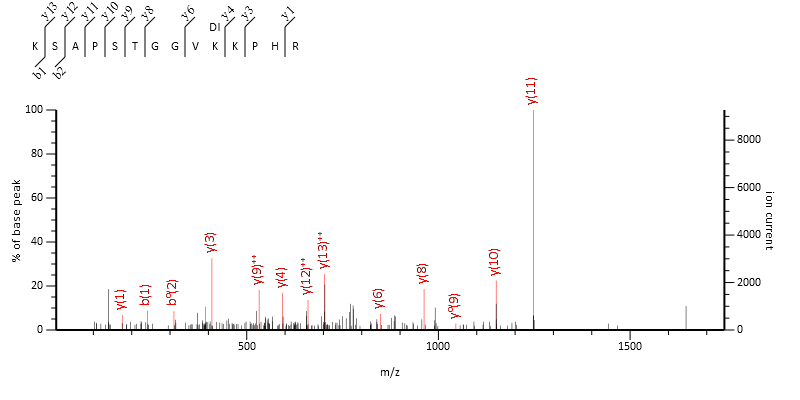
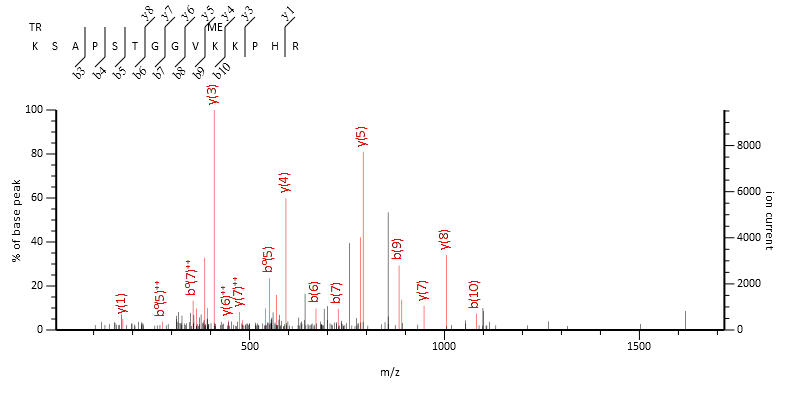
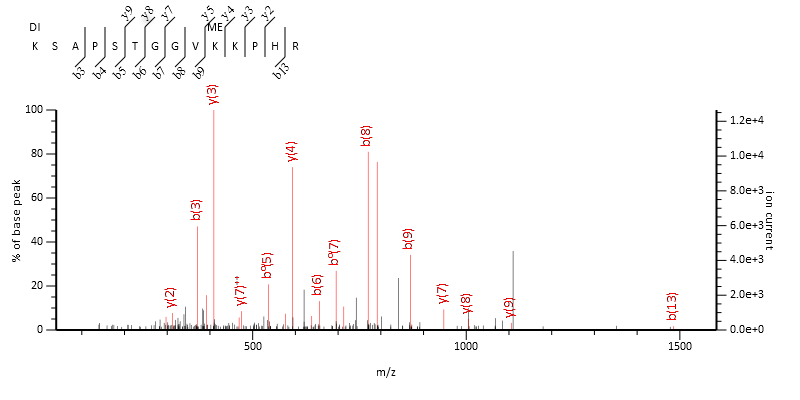
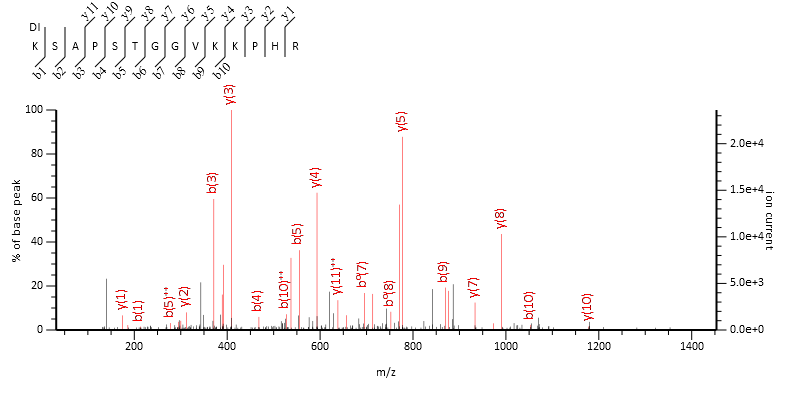
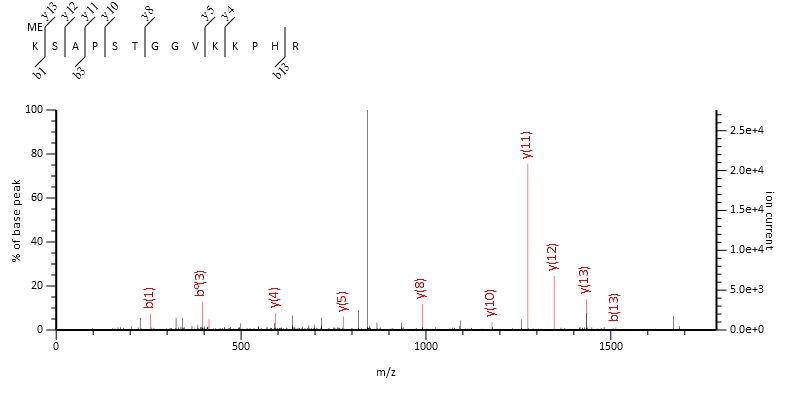
1. Those who have received, in the last six months, therapy that is likely to impair their immune function (e.g. radiotherapy, corticosteroids, chemotherapy or immunosuppressive therapy) except corticosteroids when used as a treatment for septic shock.
2. Those with pre-existing health conditions causing immunodeficiency (e.g. HIV infection)
3. Those who are pregnant
4. Participants who have a low B and T cell assay for any reason
5. Those who do not provide informed consent or in the case of septic participants those whose relatives or close friends do not provide assent or agree to participation.
   1. **Macrophage Histone PTMs MASCOT H3, H4, H2A, H2B, H1**

The following section displays the MASCOT output for all found and manually verified histone PTMs for chapter 3 and 4.

* + 1. Histone peptides from H2A, H2B and H1



* + 1. **Histone peptides from histone H3.1/2 and H3.3**

****

* + 1. **Histone peptides from Histone H4**

