

Documentation

Description of the program

Peptide viewer reads two file inputs, a fasta file and a .csv file that contains OMSSA output. Then it parses both files in order to find matching peptides between both files. After finding the matches, the program highlights identified peptides visually in three different areas by three different methods. First, it shows a graphic which contains a rectangle which represents identified peptide sequences as colored, while the unidentified sequences remain white. Second, it shows the whole input fasta sequence as a text with identified peptides as colored text. Finally, it shows a table of identified peptides with their various properties, such as the starting point, ending point, e-value, mass, p-value etc.

Class Design

Four different classes are created:

- Peptide**, to store information about a peptide sequence
- Range**, to store the start and end positions of a matching peptide sequence
- Visual**, functions to create visualizations for printing the sequence rectangle, sequence text and the peptide table list.
- Parse**, functions to parse .fasta or .csv files into String or Peptide objects, respectively.

Peptide class

- Spectrum number
- File name
- Peptide sequence
- E-value
- Mass
- GI
- Accession
- Start
- End
- Defline
- Mods
- Charge
- Theoratical mass
- P-value

-Nist score

Range class

-Starting index

-End index

Visual class

void paint(Graphics g, JPanel panel)

-Parameters: g = Graphics object of the JPanel

panel = JPanel object to be painted on

-Sets the color of the Graphics parameter component to white, then fills a rectangle on the JPanel parameter covering the whole panel

double printRectangle(Jpanel panel, List<Range> rangeList, String sequence, Graphics g)

-Parameters: panel = JPanel object to be painted on

rangeList = Range object list that has the identified peptides' ranges

sequence = Fasta sequence input

g = Graphics object of the JPanel

-Draws an empty, non-filled rectangle at the middle of the JPanel parameter with the length of the fasta sequence that is currently uploaded by using drawRect()

-Loops through the Lis<Range> object list (match ranges) parameter and highlights the ranges with the color green by using fillRect()

-Returns the overall coverage of the match lengths in the fasta sequence to display below the rectangle graphic

void printSequence(JtextPane textPane, String sequence, List<Peptide> peptideList, List<Range> rangeList)

-Parameters: textPane = JTextPane to be printed on

sequence = Fasta sequence input

peptideList = Identified matching Peptide object list

rangeList = Range object list that has the identified peptides' ranges

-Shows the raw fasta sequence on sequenceTextPane

-Loops through every matched Peptide objects and removes the matches sequence, then inserts the same sequence with the color green.(using StyledDocument and SimpleAttributeSet objects)

-To correctly visualize the fasta format, adds line breaks after 60 characters in the sequence

void printTable(JTable table, List<Peptide> peptideList)

-Parameters: table = JTable to be populated

peptideList = Identified matching Peptide object list

-Creates a DefaultTableModel object with the Jtable parameter and removes the current entries if there are any

-Loops through the List<Peptide> and adds a new entry for each Peptide object using the attributes by using .setValueAt(), along with the accession number and the full name of the peptide

Parse class

String readFasta(File file)

-Parameters: file = File object to be read from

-Creates a scanner object from the File parameter

-Skipping the first line(because .fasta header), it reads all lines and creates a String object by concatenating the lines.

-Returns the created String

List<Peptide> readOmssa(File file)

-Parameters: file = File object to be read from

-Creates a scanner object from the File parameter

-Skipping the first line(because .csv header), it reads all lines and splits each line by “,”.

-Creates a new Peptide object with the attributes that are parsed from the each line in the .csv file

-Returns the list of all Peptide objects that are created

INTERFACE

Peptide viewer consists of a JFrame which has five main JPanel elements that are used for the file inputs and visual outputs. These Jpanels are namely:

-uploadMenuPanel[JPanel]

-uploadFastaButton[JButton]

-uploadOmssaButton[JButton]

-showPeptidesButton[JButton]

-labelFasta[JLabel]

-labelOmssa[JLabel]

-visualRectanglePanel[JPanel]

-rectangleLabelContainer[JPanel]

-rectangleLabel[JLabel]

-visualRectangle[JPanel]

-coverageLabel[JLabel]

-sequenceTextPanel[JPanel]

-sequenceScrollPane[JScrollPane]

-sequenceTextPane[JTextPane]

-sequenceLabelPanel[JPanel]

-sequenceLabel[JLabel]

-peptideListPanel[JPanel]

-tableScrollPane[JScrollPane]

-peptideTable[JTable]

-recentsPanel[JPanel]

-recentsLabel[JLabel]

-recentsScrollPane[JScrollPane]

-recentsList[JList]

uploadMenuPanel

uploadMenuPanel is the top panel in the program which contains uploadFastaButton, uploadOmssaButton, showPeptidesButton, labelFasta and labelOmssa.

uploadFastaButton & uploadOmssaButton

These buttons has an actionPerformed listener attached to them which lets the user select an input .fasta or .csv file. Listener works as follows:

- Creates a JFileChooser object
- Creates a FileNameExtensionFilter object that only accepts .fasta or .csv files(depending on the button) and attaches it to the JfileChooser object
- Checks if the user has selected a valid input file
- If a valid input has been selected, the file is read with readFasta() or readOmssa()(see Parse class) function
- For the uploadFastaButton, the String that is returned from readFasta is stored in the global variable fastaSequence
- For the uploadOmssaButton, the Peptide object list that is returned from readOmssa is stored in the global variable peptideList
- Changes the background color of the the relevant button to green in order to show that the file has been successfully loaded

showPeptidesButton

This button is where the main functions of the program is performed. It uses the global variables that were initialized from uploadFastaButton and uploadOmssaButton to visualize various information. The button works as follows:

- Checks if both .fasta and .csv files are successfully uploaded, shows a message dialog warning if not
- If the files are uploaded, it searches for matches between the fastaSequence and peptideList objects' sequence attributes.
- It stores the matching sequences and their matching index ranges in matchPeptideList and matchRangeList which are initialized at the beginning
- Checks if there are any matches, displays a message dialog that warns the user if not. Also changes the background color of the upload buttons from green to default to show that the files need to be reuploaded for valid results
- If there are any matches, stores the accession number of the protein on the recentsList (after checking for uniqueness)
- Before rectangle visualizing, it removes the existing rectangle visual by using paint()(see Visual class)

-Visualizes the sequence graphic as a rectangle which shows matching peptides as colored and the rest white on visualRectangle JPanel by using printRectangle()(see Visual class)Also shows the coverage of the matching peptides under the rectangle visual

-Visualizes the fasta sequence text with identified peptides highlighted as colored on sequenceTextPane JTextPane using printSequence()(see Visual class)

-Populates the peptideTable Jtable with identified peptides list using printTable()(see Visual class)

visualRectanglePanel

visualRectanglePanel is the panel that contains the rectangle visualization of the identified peptides in the fasta sequence. It has JLabel objects for the title and the coverage. Rest of the panel is used for the rectangle graphic.

sequenceTextPanel

sequenceTextPanel is the panel that contains the text sequence visualization of the identified peptides in the fasta sequence. It has a JLabel object for the title. Rest of the text pane is used for the text.

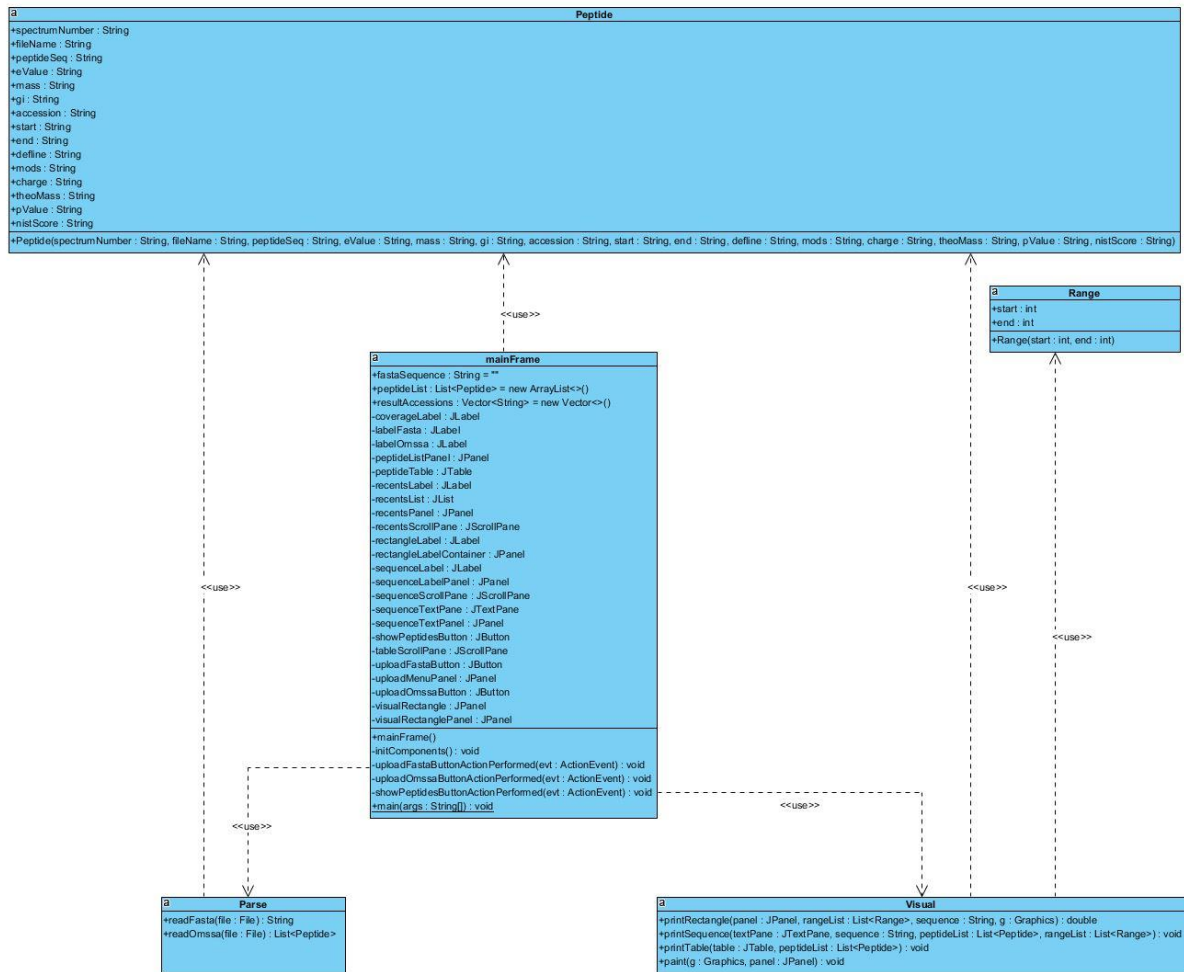
peptideListPanel

peptideListPanel is the panel that contains the table that visualizes the identified peptide objects' attributes.

recentsPanel

recentsPanel is the panel that contains JList which has the accession numbers of the recently uploaded .fasta files. It also contains a JLabel for the title.

UML DIAGRAM



REFERENCES TO IMPORTED CLASSES

java.awt.Color

java.awt.Graphics

java.util.List

java.util.logging.Level

java.util.logging.Logger

javax.swing.JPanel

javax.swing.JTable

javax.swing.JTextPane

javax.swing.table.DefaultTableModel

javax.swing.text.BadLocationException

javax.swing.text.SimpleAttributeSet

javax.swing.text.StyleConstants

javax.swing.text.StyledDocument

java.io.File

java.util.ArrayList

java.util.Vector

javax.swing.JFileChooser

javax.swing.JOptionPane

javax.swing.filechooser.FileNameExtensionFilter

java.io.FileNotFoundException

java.util.Scanner

FLOWCHART OF THE LOGICAL FLOW OF THE PROGRAM

