package Codes;

import java.util.\*;

/\*window number,range,(peak-avg)/sd and merge peaks if they are very close say 3-5 window numbers..find the threshold using analysis and data to be outputed should be of top5 peaks only if there are too many peaks above that threshold but if optimisation algorith is applied then all peaks are to be printed.\*/

public class peak

{

public ArrayList<Integer> arraylistsix=new ArrayList<Integer>();

public ArrayList<Double> arraylistfive=new ArrayList<Double>();

public static double average (double[] list)

{

// 'average' is undefined if there are no elements in the list.

if (list == null || list.length==0)

return 0.0;

// Calculate the summation of the elements in the list

double sum = 0;

int n = list.length;

// Iterating manually is faster than using an enhanced for loop.

for (int i = 0; i < n; i++)

sum += list[i];

//System.out.println(1);

//System.out.println(sum);

// System.out.println(n);

// We don't want to perform an integer division, so the cast is mandatory.

return (double)(sum/n);

}

public static double sd (double[] table,double av)

{

// Step 1:

double mean = av;

double temp = 0;

for (int i = 0; i < table.length; i++)

{

double val = table[i];

// Step 2:

double squrDiffToMean = Math.pow(val - mean, 2);

// Step 3:

temp += squrDiffToMean;

}

// Step 4:

double meanOfDiffs = (double) temp / (double) (table.length);

// Step 5:

return Math.sqrt(meanOfDiffs);

}

public peak (double[] arrayone)

{

double[] arraytwo=new double[arrayone.length-1];

for(int i=0;i<arrayone.length-1;i++)

{

arraytwo[i]=arrayone[i+1]-arrayone[i]; //finding difference between consecutive terms aka dx

}

double[] arraythree=new double[arrayone.length];

for(int i=0;i<arraythree.length-1;i++)

{

arraythree[i]=arraytwo[i];//constructing (dx,0)

if(arraythree[i]<0)

{

arraythree[i]=1;

}

else

{

arraythree[i]=0;

}

}

arraythree[arraythree.length-1]=0;//(dx,0)

double[] arrayfour=new double[arrayone.length];

for(int i=1;i<arrayfour.length;i++)

{

arrayfour[i]=arraytwo[i-1]; //constructing (0,dx)

if(arrayfour[i]>0)

{

arrayfour[i]=1;

}

else

{

arrayfour[i]=0;

}

}

arrayfour[0]=0;//(0,dx)

//System.out.println(Arrays.toString(arraythree));

//System.out.println(Arrays.toString(arrayfour));

//System.out.println(Arrays.toString(arraytwo));

//System.out.println(Arrays.toString(arrayone));

ArrayList<Integer> arraylistone=new ArrayList<Integer>();

for(int i=0;i<arraythree.length;i++)

{

if (arraythree[i]==arrayfour[i])

{

if(arraythree[i]==1)

{

arraylistone.add(i);//index of peaks

}

}

}

ArrayList<Double> arraylisttwo=new ArrayList<Double>();

for(int i=0;i<arraylistone.size();i++)

{

arraylisttwo.add(arrayone[(int)arraylistone.get(i)]); //elements in arrayone corresponding to indexes in arraylistone

}

//System.out.println(arraylisttwo);

//System.out.println(arraylisttwo.size());

ArrayList<Double> arraylistthree=new ArrayList<Double>();

arraylistthree.addAll(arraylisttwo); //Copying arraylisttwo to arraylistthree

//System.out.println(arraylistthree);

//System.out.println(arraylistthree.size());

Collections.sort(arraylistthree); //arraylistthree is in ascending order

//System.out.println(arraylistthree);

//System.out.println(arraylistthree.size());

double avg=average(arrayone); //average of elements in arrayone

//System.out.println(avg);

double std=sd(arrayone,avg); //standard deviation of elements in arrayone

//System.out.println(std);

ArrayList<Double> arraylistfour=new ArrayList<Double>();

for(int i=0;i<arraylistthree.size();i++)

{

arraylistfour.add((double)((arraylistthree.get(i)-avg)/std)); //array of (peak-average)/std deviation

}

//System.out.println(arraylistfour);

//System.out.println(arraylistfour.size());

int a=arraylistfour.size()-1;

if(arraylistfour.get(arraylistfour.size()-1)<10.45)

{

while(arraylistfour.get(a)>((arraylistfour.get(arraylistfour.size()-1))/1.760105116)) //constraints decided

{

arraylistfive.add(arraylistfour.get(a));

a--;

}

}

else

{

while(arraylistfour.get(a)>((arraylistfour.get(arraylistfour.size()-1))/3.043552241)&& arraylistfour.get(a)>6)

{

arraylistfive.add(arraylistfour.get(a));

a--;

}

}

for(int i=arraylisttwo.size()-arraylistfive.size();i<arraylisttwo.size();i++)

{

arraylistsix.add(arraylistone.get(arraylisttwo.indexOf(arraylistthree.get(i)))); //arraylist six gives us the window number

}

Collections.reverse(arraylistsix);

//System.out.println(arraylistsix);

//System.out.println(arraylistsix.size());

}

public ArrayList<Integer> peakfunction1()

{

return arraylistsix;

}

public ArrayList<Double> peakfunction2()

{

return arraylistfive;

}

public static void main(String[] args)

{

//

}

}