package Codes;

import java.io.BufferedInputStream;

import java.io.FileInputStream;

import java.io.FilterInputStream;

import java.io.IOException;

import java.io.InputStream;

import Codes.Complex;

public class iCorr extends Thread

{

public double[] finalarr;

//Complex complex = new Complex(0.0,0.0);// = new Complex();

public iCorr(String nm, int ws, int ss, int len)

{

InputStream is = null;

FilterInputStream fis = null;

int p=0,k=0,l=0,j=0,s=0;

int len1=0,len2=0;

int gl=len;

int windows=(int)((gl-ws)/ss)+1;

char c;

double Cg;

byte [] buffer = new byte[ws];

double arr[]=new double[windows];

Complex[] newarr= new Complex[ws];

Complex z = new Complex(0, 0);

try

{

// create input streams

is = new FileInputStream(nm);

fis = new BufferedInputStream(is);

for(p=0;p<windows;p++)

{

len1 = fis.read(buffer, 0,ss);

fis.mark(ws);

len2=fis.read(buffer,ss,ws-ss);

for(j=0;j<ws;j++)

{

//System.out.println("the array= " + (char)buffer[j]);

if((char)buffer[j]=='G')

newarr[j]=new Complex(1.0,0.0);

else if((char)buffer[j]=='A')

newarr[j]=new Complex(0.0,1.0);

else if((char)buffer[j]=='C')

newarr[j]=new Complex(-1.0,0.0);

else if((char)buffer[j]=='T')

newarr[j]=new Complex(0.0,-1.0);

}

Cg=0;

for(k=1;k<ws;k++)

{

Complex Ck = new Complex(0, 0);

for (l=1;l<ws-k+1;l++)

{

z=(newarr[l-1]).times((newarr[l+k-1]));

Ck=Ck.plus(z);

}

// System.out.println("Ck="+(Ck));

Cg=Cg+(double)((Ck.abs()+0.0)/(ws-k));

//System.out.println("Cg="+(Cg));

}

//System.out.println("value="+((Cg+0.0)/(ws-1)));

arr[s]= ((Cg+0.0)/(ws-1));

s++;

fis.reset();

// System.out.println("we have performed the reset!");

}

/\* for(s=0;s<windows;s++)

{

System.out.println((double)arr[s]);

}\*/

}

catch(IOException e)

{

// if any I/O error occurs

e.printStackTrace();

}

finally

{

// releases any system resources associated with the stream

if(is!=null)

{

try

{

is.close();

}

catch (IOException e)

{

// TODO Auto-generated catch block

e.printStackTrace();

}

}

if(fis!=null)

{

try

{

fis.close();

}

catch (IOException e)

{

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

finalarr=arr;

}

public double[] icorrfunction()

{

return finalarr;

}

public static void main(String[] args)

{

}

}