1 import java.util.\*;  
 2 import java.io.\*;  
 3   
 4 public class GlobalAlign{  
 5 private int[][] vert;  
 6 private int match, indel, mismatch;  
 7 private String seq1, seq2;  
 8 private String align1, align2;  
 9 public GlobalAlign(String a, String b, int i, int j, int k)  
10 {  
11 vert = new int[a.length()+1][b.length()+1]; //seq1 = rows and seq2 = columns  
12 seq1 = a;  
13 seq2 = b;  
14 match= i;  
15 indel = j;  
16 mismatch = k;  
17 }  
18 public int[][] getMatrix()  
19 {  
20 return vert;  
21 }  
22 public String getAlign1()  
23 {  
24 return align1;  
25 }  
26 public String getAlign2()  
27 {  
28 return align2;  
29 }  
30 public int score(char x, char y)  
31 {  
32 if(x == y)  
33 return match;  
34 //else if (y == '-' || x == '-')  
35 //return indel;  
36 else   
37 return mismatch;  
38 }  
39 public void traceback(int i, int j)//i and j will be the right most corner   
40 {  
41 align1="";  
42 align2 = "";  
43 while(i>0 ||j>0) //[rows][columns]  
44 {  
45 if(i>0 && j>0 && vert[i][j] == vert[i-1][j-1]+score(seq1.charAt(i-1), seq2.charAt(j-1)))  
46 {  
47 align1 = seq1.charAt(i-1) + align1;  
48 align2 = seq2.charAt(j-1)+ align2;  
49 i = i-1;  
50 j = j-1;  
51 } else if(i>0 && vert[i][j] == vert[i-1][j]+indel){//down  
52 align1 = seq1.charAt(i-1) + align1;   
53 align2 = "-" + align2;  
54 i = i-1;  
55 } else if(j>0 && vert[i][j] == vert[i][j-1]+indel){//across  
56 align1 = "-" + align1;  
57 align2 = seq2.charAt(j-1) + align2;  
58 j = j-1;  
59 }   
60 }  
61 }  
62 public void fillMatrix(int i, int j)  
63 {  
64   
65 if(i>=vert.length)//base cases  
66 return;  
67 if(j>=vert[0].length)   
68 return;  
69   
70 int max = 0;  
71 int diagonal = vert[i-1][j-1]+ score(seq1.charAt(i-1), seq2.charAt(j-1));  
72 int across = vert[i-1][j] +indel;  
73 int down = vert[i][j-1] + indel;  
74   
75 if(down>=across && down>=diagonal)//find max of 3 paths  
76 vert[i][j] = down;  
77 else if (across>=down && across>=diagonal)  
78 vert[i][j] = across;  
79 else  
80 vert[i][j] = diagonal;  
81   
82 fillMatrix(i,j+1);  
83 fillMatrix(i+1,j); //recurse through whole matrix  
84 }  
85   
86 }

4 4 public class AlignmentTest  
 5 {  
 6 public static void main(String[] args)  
 7 {  
 8 String seq1 = "TGCAGGGTGAC"; //11  
 9 String seq2 = "CCATGAGCGA"; //10  
10 GlobalAlign g = new GlobalAlign(seq1, seq2, 1, 0, -1);  
11 g.fillMatrix(1,1);  
12   
13 int[][] matrix = g.getMatrix();  
14   
15 g.traceback(matrix.length-1, matrix[0].length-1);  
16 String align1 = g.getAlign1();  
17 String align2 = g.getAlign2();  
18   
19 System.out.println("input is " + seq1 + " and " + seq2);  
20 System.out.println("the scoring scheme is match = 1, indel = 0, and mismatch = -1");  
21 System.out.println();  
22   
23 printArr(matrix);  
24 System.out.println();  
25 System.out.println(align1);  
26 System.out.println(align2);  
27   
28 System.out.println();  
29 System.out.println("the score is "+ matrix[matrix.length-1][matrix[0].length-1]);   
30   
31 }  
32 public static void printArr(int[][] arr)  
33 {  
34 for(int i = 0; i<arr.length; i++)  
35 {  
36 for(int j = 0; j<arr[0].length; j ++)  
37 {  
38 System.out.print(arr[i][j]);  
39 }  
40 System.out.println();  
41 }  
42 }  
43 }

Graphical user interface, text, email

Description automatically generated