IMPLEMENTACION DEL ALGORITMO DE ORDENAMIENTO DCJ

BIOINFORMÁTICA

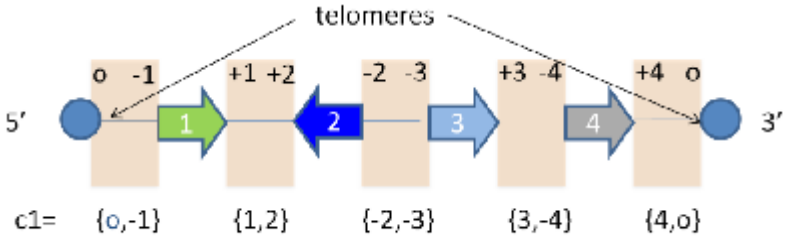
11 de mayo del 2016

De Anda Gil Janeth

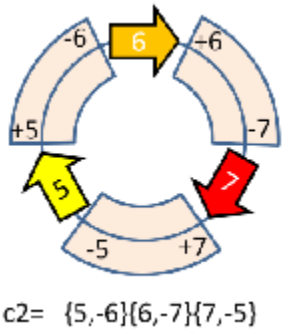
Objetivo: Implementar el algorítmo de ordenamiento DCJ (double cut and join)

Introducción:

El DCJ es un modelo que se usa para hacer operaciones de rearreglos, en este modelo un genoma es agrupado en cromosomas , estos pueden ser lineales (telomero) y/o circulares, en el cromosoma lineal se comienza con un cero y se finaliza con un cero ya que no hay forma de conectar el primer elemento del cromosoma con el último como sería el caso de un cromosoma circular.
Un gen en la hebra froward está representado con [-g,+g], mientras que un gen en la hebra reverse está representado con [+g,-g]
Por ejemplo en un cromosoma lineal c1=0,1,-2,3,4,0 nuestro cromosoma quedaría representado como en la figura de abajo.



Mientras que en un cromosoma circular c2=5,6,7 , nuestro cromosoma quedaría representado como en la figura de abajo, en este caso no se requieren de ceros, porque todos los genes forman un ciclo.



Se realizan tres operaciones DCJ1,
DCJ2, DCJ3 para transformar un genoma en otro, más adelante se hablará de cada uno de estos.

Implementación :

La implementación del algoritmo DCJ se realizo en el lenguaje Java, en una clase llamada Rearreglos_DCJ, la cual contiene el método inicio(), adyacente(int [][], int [][]), DCJ1(), DCJ3() e imprime_genoma(int [][], int[][]).

Método inicio

Se generan las representaciones de los cromosomas lineal, circular o ambos, de un genoma, primero se le pide al usuario que escriba sus dos genomas, en caso de que algun genoma contenga una representación lineal el usuario debe de agregar un cero al inicio y un cero al final.

Se tomo en cuenta el caso de que un genoma contuviera las dos representaciones de los cromosomas por lo tanto se coloca una bandera llamada c_l, donde 0 indica que la representación es lineal, 1 indica que la representación es circular y va comenzando el cromosoma. Cuando la bandera c_l=2 y existe un gen 0, nos indica que acaba de finalizar el cromosoma circular y que comienza un genoma lineal o telómero.

```
static public void inicio()
{

    Scanner sc=new Scanner(System.in);
    int gen=0;
    int i=0,s=0,i_r=0;
    int c_l=1;//0=lineal, 1=circular
    System.out.println("\n                    CDJ                    \n Un genoma puede estar compuesto
de cromosomas lineales (telomero ) y/o circulares, el cromosoma lineal  comienza con un cero y finaliza con un cero ya
que no hay forma de conectar el primer elemento del cromosoma con el último como sería el caso de un cromosoma
circular \n Número de genes del genoma1(si es lineal o telomero agrega uno más por cada par de ceros):");
    bloques=sc.nextInt();//número de genes del genoma 1
    rea1=new int[bloques][2];//coloca elementos de la matriz que contiene al genoma1
    System.out.println("Escribe el genoma 1 (Si es un telomero o linear ,comience con un 0 y finalicelo con un 0):");
    for(;i<bloques;)
    {
      System.out.println("Escribe el gen "+(i+1)+" :");
      gen=sc.nextInt();//entrada del gen
       if(gen==0&&c_l==0)//si el gen es cero (telomero) y la bandera c_l=0, es decir finaliza el telomero
      {
        rea1[i][s]=0;//se coloca en la matriz un 0
        c_l=1;//se cambia la bandera c_l=1, que indica que la siguiente parte del gen puede ser circular
      }
      else if(gen==0&&c_l==2)//con el gen =0 y la badera c_l=2 quiere decir que finaliza una parte del genoma que
es circular y comienza un telómero
      {
        rea1[i_r][0]=rea1[i-1][1]*(-1);//el gen esta representado como [-g,+g] o [+g,-g],empezando con el gen con su
signo contrario, este ultimo gen lo conectaremos con el primero ya que es un cromosoma circular
        c_l=0;//bandera que indica un gen circular
        s=0;
        rea1[i][s]=0;
      }
```

En el caso donde acaba un cromosoma circular pero enseguida comienza otro cromosoma circular, la representación cambia, conectaremos los genes que contienen a cero y recorremos los ceros en el anterior gen, ejemplo 0120034 la representación sin contar aún que hay ceros juntos quedá de la siguiente manera {0,-1},{1,-2},{2,0},{0-3},{3,-4},{4,0}, lo que se hace es conectar el 2 con el 3 y al gen anterior {1,-2}, le intercalamos los ceros, es decir, {1,0},{0-2}, por lo tanto la representación completa seria {0,-1},{1,0},{0,-2},{2,3},{0-3},{3,-4},{4,0}, en la siguiente imagen , se muestra la parte de código donde se encuentran dos genes lineales o telomeros juntos, en este caso cuando el gen sea cero indica que inicia un cromosoma gen lineal y la bandera c_l=1 indica que había terminado un cromosoma y que fue lineal, en caso de que hubiese sido circular la bandera se hubiese cambiado a c_l=2, el i>2 se coloca para asegurarnos que el no es el primer cromosoma lineal, por lo tanto hay que hacer los cambios que se mostraron anteriormente.

```
      else if(gen==0&&c_l==1&&i>2)//gen=0 y c_l=1 indica que acaba de finalizar un cromosoma, pero
tambien acaba de empezarun cromosoma lineal, i>2 asegura que no es el primer cromosoma lineal
      {

        rea1[i][0]=rea1[i-1][0];//se coloca el gen anterior
        rea1[i-1][0]=0;//se coloca un cero
        rea1[i-1][1]=rea1[i-2][1]; //al gen anterior se lo camia por el gen que se encuentra dos posiciones
anteriores
        rea1[i-2][1]=0;
        c_l=0;
      }
```

Si el gen es igual a cero inicia un cromosoma lineal. Cuando un gen es diferente de cero entonces verificamos si el cromosoma es circular con la bandera en uno,en seguida cambiamos la bandera a 2, ya que cuando empiece un cromosoma circular (gen=0) o cuando ya no existan más genes se debe conectar el primer gen con el último. En este caso se guarda el indice en la variable i_r, que será la posición donde se colocará el último gen que cerrara el ciclo con el primer gen; también se aumenta el indice en uno para ahí comenzar con el primer gen del cromosoma circular. En dado caso de que no se comience un gen circular entonces se comienza a generar la representación del gen , sabemos que primero se coloca el gen con signo contrario , seguido del gen con su signo , es decir, [+g,-g] o en su caso [-g,+g], entonces en el código multiplicamos el gen por -1 y se coloca en la matriz, seguido del gen original, donde i, es el indice de la matriz de los renglones, mientras que s, es el índice de las columnas de la matriz, en este caso se tienen solo dos columnas , debido a la representación {gi,gj}. A continuación se muestra una imagen con el código de está parte

```
    else if(gen==0)//inicia un cromosoma lineal o telómero
    {
      rea1[i][s]=0;
      c_l=0;
    }
    else
    {
      if(c_l==1)//si es circular
      {
        c_l=2;//cambiamos la bandera
        i_r=i;//guardamos el indice de linear
        s++;//aumentamos el indice
      }
      rea1[i][s]=gen*(-1);//se asigna en la matriz el gen con el signo contrario
      s++;//se incrementa s que representa el número de columnas
      if(s>1)//si s>1 , es decir hay más de dos columnas
      {
        s=0;//se inicializa s
        i++;//Se incrementa i, que representa el número de filas
      }
      if(i<bloques)//Si las filas son menor al bloque de genes que inserto el usuario, colocamos el gen en la matriz
        rea1[i][s]=gen;
    }
    s++;
    if(s>1)//si s>1 , es decir hay más de dos columnas
    {
      s=0;//inicializar s, que representa las columnas
      i++;//incrementar i
    }
  }
  if(c_l==2)//si la bandera c_l  es igual a 2 , significa que termino mi genoma circular
  {
    rea1[i_r][0]=rea1[i-1][1]*(-1);//conecto mi último gen con el primero
  }
```

Se realiza el mismo procedimiento para el genoma 2 y se asigna el tamaño de la matriz de adyacencias con el máximo número de bloques entre el genoma una y el genoma dos, también se asigna el tamaño de la matriz are1_or, la cual tendra el ordenamiento final del genoma1.

```
/*inicializo la matriz de adyacencias con el tamaño maximo de los bloques de los genes y la matriz rea1_or*/
if(bloques>m_a)
{
   adyacencias=new int[bloques][2];
   rea1_or=new int[bloques][2];
}
else
{
   adyacencias=new int[m_a][2];
   rea1_or=new int[m_a][2];
}
```

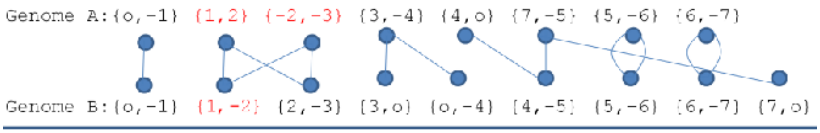
Método adyacente(int,int)

El método adyacente recibe como parámetros dos matrices , una es la representación del genoma 1 y la otra es la representación del genoma 2, en la matriz de adyacencia cada fila representa un gen {gj,gi} del genoma B y las columnas representan el número de la fila de la matriz del genoma A donde se encuentra el mismo elemento del genoma de B. Para construir la matriz de adyacencias se busca cada elemento del genoma A si es igual a algún elemento del genoma B , a continuación se muestra el código que implementa la construcción de la matriz de adyacencia.

```
/* Método para obtener la matriz de adyacencias*/
static void adyacente(int [][] rea1, int [][] rea2)
{
   int i=0,j=0,k=0,l=0,s=0;
   boolean salir;
   /*inicializo la matriz de adyacencias con -1*/
   for(i=0;i<adyacencias.length;i++)
      for(j=0;j<2;j++)
         adyacencias[i][j]=-1;
   /*busco en cada elemento del genoma2 ,con quien esta conectado del genoma 1*/
   for(i=0;i<adyacencias.length;i++)//recorre las filas
   {
      for(j=0;j<2;j++)
      {
         salir=false;
         if(rea2[i][j]!=0)//si el elemento es un gen, es decir, diferente de cero
         {
            for(k=0;salir==false&&k<rea1.length;k++)//filas del genoma 1
            {
               for(l=0;l<2;l++)
                  if(rea1[k][l]==rea2[i][j])//cuando un elemento del genoma2 es igual a un elemento del genoma1
                  {
                     salir=true;
                     adyacencias[i][s]=k;//se coloca en la matriz de adyacencia la fila de la matriz del elemento del
genoma1 que es igual al elemento genoma2
                     s++;
                     if(s>1)//si s>1 , es decir hay más de dos columnas
                        s=0;
                  }
            }
         }
      }
   }
```

Método DCJ1

En este método lo que se hace es intercambiar la posición de los genes del genoma 1 a la posición donde se encuentran en el genoma 2, es decir, para el genoma 1 se tiene {p,l}{q,m}estan relacionados con el genoma {p,q} del genoma 2, pot lo tanto se intercambian los elementos del genoma 1 , en este caso se intercambia l por q , es decir, el resultado sería {p,q},{l,m}, en la siguiente figura se muestra un ejemplo, donde {1,-2} del genoma B, está conectado con {1,2} y {-2,-3} del genoma A, para que uno de los dos genes de A quede igual que el de B, intercambio un gen del genoma A que haga que {gj,gi} sea igual a {1,-2} del genoma de B, es decir intercambio el -2 con el 2.



genome A: (o 1 -2 3 4 o) (5 6 7)
genome B: (o 1 2 3 4 o) (o 5 6 7 o)
DCJ1: {1,2} {-2,-3} → {1,-2} {2,-3}

Para esto se hace uso de la matriz de adyacencia la cual nos indica las filas de la matriz del genoma 1 en donde se encuentran los genes iguales al genoma2, entonces se toman esas filas y se compara si el elemento de la columna cero del genoma 1 es igual al elemento de la columna cero o uno del genoma 2, de no ser así entonces el elemento que es igual es el de la columna 1 del genoma 2, tomaremos la variable p1 para una fila, que indicara el elemento que no es igual , en este caso 0, de lo contrario p1 valdra 1 y para la otra fila se tomara p2. Una vez que se sabe que elementos no son los que son iguales, se sabe entonces los elementos del genoma 1 que son iguales a los del genoma 2, por lo tanto, se intercambian el elemento p2 que es un elemento del genoma 1 que no se encuentra en el genoma 2, por el elemento 1 de la fila v1 que si se encuentra en el genoma 2 si p1=0

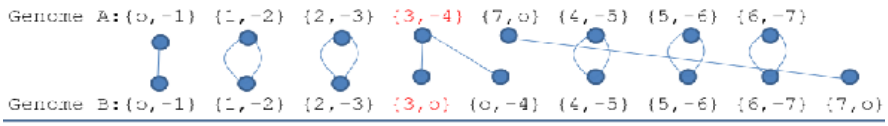
```
/* Método que implementa el algoritmos de DCJ1*/
static void DCJ1()
{
    int intercambia=0;//0,1 -> no intercambia;2->intercambia
    int v1=0,v2=0,aux=0,i=0,p1=0,p2=0;
    for(i=0;i<adyacencias.length;i++){//Se obtienen los elementos de la matriz de adyacencia,es decir, las filas de la matriz del genoma 1 que tienen un elemento común con 2
      v1=adyacencias[i][0];
      v2=adyacencias[i][1];
      if(v1!=-1&&v2!=-1&&v1!=v2)//Si los elementos v1 y v2 son diferentes de -1, quiere decir que existen un elemento igual en el genoma 2 y si los elementos son diferentes entre si
      {
        if(rea1[v1][0]!=rea2[i][0]&&rea1[v1][0]!=rea2[i][1])//si el elemento de la columna 0 y la fila v1 de la matriz del genoma 1 no es ni el elemento de la columna 0 y 1 de la matriz del genoma 2.
          p1=0;//se asigna a p1=0
        else if(rea1[v1][1]!=rea2[i][1]&&rea1[v1][1]!=rea2[i][1])//si el elemento de la columna 0 y la fila v1 de la matriz del genoma 1 no es ni el elemento de la columna 0 y 1 de la matriz del genoma 2.
          p1=1;//se asigna a la variable p1=1
        if(rea1[v2][0]!=rea2[i][0]&&rea1[v2][0]!=rea2[i][1])//si el elemento de la columna 0 y la fila v2 de la matriz del genoma 1 no es ni el elemento de la columna 0 y 1 de la matriz del genoma 2.
          p2=0;//se asigna a la variable p2=0
        else if(rea1[v2][1]!=rea2[i][1]&&rea1[v2][1]!=rea2[i][1])//si el elemento de la columna 1 y la fila v2 de la matriz del genoma 1 no es ni el elemento de la columna 0 y 1 de la matriz del genoma 2.
        //el elememto p2 representa la columna que no es igual
          p2=1;//se asigna a la variable p2=1
        aux= rea1[v2][p2];
```

```
            if(p1==0)//si p1 es igual a cero entonces la columna 0 de la matriz del genoma 1 no es igual al elemento del genoma 2
            {
            //se intercambian el elemento p2 que es un elemento del genoma 1 que no se encuentra en el genoma 2, por el
elemento 1 de la fila v1 que si se encuentra en el genoma 2
              rea1[v2][p2]= rea1[v1][1];
              rea1[v1][1]=aux;
            }
            else{
            //se intercambian el elemento p2 que es un elemento del genoma 1 que no se encuentra en el genoma 2, por el
elemento 0 de la fila v1 que si se encuentra en el genoma 2
              rea1[v2][p2]= rea1[v1][0];
              rea1[v1][0]=aux;
            }
            System.out.println("DCJ1:");
            imprime_genomas(rea1,rea2);//Imprime los genomas 1 y 2
            adyacente(rea1,rea2);//genera la matriz adyacente
          }
        }

    }
```

Método DCJ3

En esté método, si existe en el genoma 2 una tupla de la siguente manera {l,p}, donde p=0, entonces se verifica con que elemento es adyacente en la matriz de adyacencia supongamos {l,m}, y se construyen los siguientes genes {l,0},{m.0}



genome A: (o 1 -2 3 4 o) (5 6 7)
genome B: (o 1 2 3 4 o) (o 5 6 7 o)
DCJ3: {3,-4} {o,o} → {3,o} {o,-4}

Debido a que quizá se agreguen más elementos al genoma, se crea una nueva matriz con la longitud del genoma 2, y se hace una copia de los elementos de la matriz del genoma 1 en la nueva matriz llamada 'rea1_or'. Se recorre la matriz del genoma 2 en busca de un elemento que cuente con las siguientes caracteristicas {l,0}, una vez que se encuentra un elemento, se busca en la matriz de adyacencia la fila con la cual se relaciona con el genoma 1. Ya que se debe dejar un espacio para el nuevo elemento que se va a agregar , se coloca en el último elemento de la nueva matriz del genoma 1 la fila que se va a remplazar de la matriz del genoma 1. Si en el gen {p,q} del genoma 1 con el que se relaciona no tiene ningun elemento igual a cero , entonces en la nueva matriz colocamos p en la columna 0 y un cero en la columna 1 y en la fila siguiente de la nueva matriz colocamos en la columna 0 un cero y en la columna 1 colocamos q. En la siguiente imagen se muestra el código en donde se implementa el DCJ3

```
  /* Método que implementa el DCJ3*/
  static void DCJ3()
  {
    int i=0,v1=0,v2=0,s=0;
      for(i=0;i<adyacencias.length;i++){
        if(i<m_a){//Se copia la matriz del genoma 1 en una nueva matriz, mientras el valor de 1 sea menor a los bloques
indicados del genoma 1
          rea1_or[s][0]=rea1[i][0];
          rea1_or[s][1]=rea1[i][1];
        }
        if(rea2[i][0]!=0&&rea2[i][1]==0)//si encuentra un cero en la columna 1 de la matriz del genoma 2
        {
          v2=adyacencias[i][1];//busco la fila de la adyacencia con el genoma 1
          // ya que se debe dejar un espacio para el nuevo elemento que se va a agregar , se coloca en el último elemento de la
nueva matriz del genoma 1, la fila que voy a remplazar de la matriz del genoma 1
          rea1_or[m_a-1][0]=rea1[v2+1][0];
          rea1_or[m_a-1][1]=rea1[v2+1][1];
          if(rea1[v2][0]!=0&&rea1[v2][1]!=0)//si los dos elementos de la fila v2 a los que se les agregara 0 son diferentes de 0
          {
            rea1_or[s][0]=rea1[v2][0];//coloco en la nueva matriz el elemento de la columna 0 de la fila v2 de la matriz del
genoma1
            rea1_or[s][1]=0;//en la columna 1 de la nueva matriz coloco un cero
            s++;
            rea1_or[s][0]=0;
            rea1_or[s][1]=rea1[v2][1];//coloco en la nueva matriz el elemento de la columna 1 de la fila v2 de la matriz del
genoma 1
          }
        }
        s++;
      }
```

Hasta este punto ya se tendran los pares {gj,gi} iguales en los genomas 1 y 2 aunque quizá aún no esten ordenados igual, por lo tanto, el siguiente paso es ordenar el conjunto de genes, para esto se hace uso de la matriz de adyacencia para saber en que fila se encuentra del genoma 1 nuestro gen igual al genoma 2, y entonces colocar este gen en la misma fila que se encuentra el genoma 2 , entonces se recorre la matriz de adyacencia y si la fila es diferente al elemento v1(que es la fila de la matriz del genoma 1) se hace un intercambio, a continuación se muestra el código donde se implementa esta parte.

```
    /*ordena los genes del genoma1*/
    for(i=0;i<adyacencias.length;i++){
        v1=adyacencias[i][0];//toma la fila de la matriz del genoma 1 con la que está relacionado
        v2=adyacencias[i][1];//toma la fila de la matriz del genoma 2 con la que está relacionado
        if(v1==v2)//debe de estar relacionado con ambas v1 y v2 entonces deben de ser iguales
          if(i!=v1)//si la fila de la matriz del genoma 2 es diferente a la fila de la matriz del genoma1
          {
            rea1_or[v1][0]=rea1_or[i][0];//asignamos a esa fila el elemento
            rea1_or[v1][1]=rea1_or[i][1];
            rea1_or[i][0]=rea2[i][0];//asignamos a la fila el elemento que es igual al de la columna 0 de la matriz del
genoma 2
            rea1_or[i][1]=rea2[i][1];//asignamos a la fila el elemento que es igual al de la columna 1 de la matriz del
genoma 2
            adyacente(rea1_or,rea2);//construimos la matriz adyacente,porque habremos modificado el orden
          }
      }
```

Método imprime_genoma(int [][],int [][])

Este método recibe como parámetros dos matrices de genomas, recorre las matrices e imprime en pantalla los valores de la matriz con la representación {gj,gi}, a continuación se muestra el código que implementa esta parte

```java
/*Imprime los genomas*/
static void imprime_genomas(int [][] rea1, int [][] rea2)
{
  int i=0;
  /*recorre la matriz del genoma y lo imprime como la representación {gj,gi}*/
  for(i=0;i<rea1.length;i++)
    System.out.print(" ("+rea1[i][0]+","+rea1[i][1]+") ");
  System.out.println("");
  /*recorre la matriz del genoma y lo imprime como la representación {gj,gi}*/
  for(i=0;i<rea2.length;i++)
    System.out.print(" ("+rea2[i][0]+","+rea2[i][1]+") ");
  System.out.println("");
}
```

Datos de entrada del Programa

El programa recibe primero el número de genes del genoma 1, después se pide que los ingreses, en caso de que tu gen contenga cromosomas lineales o telomeros, debes de agregar un cero al inicio y uno al final, y tu cantidad de genes será el número de genes más uno por cada par de ceros que se agregará. Después se debe ingresar el número de genes del genoma 2 e igual que se ingresen de la misma manera que el genoma 1.

Datos de salida

Se imprimen el orden de la representación de los genes conforme ocurren las operaciones CDJ1 y CDJ3

Pruebas del código

Suponiendo los genomas con cromosomas lineales g1=(0,4,1,3,5,2,0) y g2=(0,1,2,3,4,5,0) se tiene

```
                              CDJ
 Un genoma puede estar compuesto de cromosomas lineales (telomero ) y/o circulares, el cromosoma line
al  comienza con un cero y finaliza con un cero ya que no hay forma de conectar el primer elemento de
l cromosoma con el último como sería el caso de un cromosoma circular
 Número de genes del genoma1(si es lineal o telomero agrega uno más por cada par de ceros):
Número de bloques de los genoma1:
6
Escribe el genoma 1 (Si es un telomero ,comience con un 0 y finalicelo con un 0):
Escribe el gen 1 :
0
Escribe el gen 1 :
4
Escribe el gen 2 :
1
Escribe el gen 3 :
3
Escribe el gen 4 :
5
Escribe el gen 5 :
2
Escribe el gen 6 :
0
Número de bloques de los genoma2:
6
```

```
Número de bloques de los genoma2:
6
Escribe el genoma 2 (Si es un telomero ,comience con un 0 y finalicelo con un 0)
:
Escribe el gen 1 :
0
Escribe el gen 1 :
1
Escribe el gen 2 :
2
Escribe el gen 3 :
3
Escribe el gen 4 :
4
Escribe el gen 5 :
5
Escribe el gen 6 :
0
Genomas:
 (0,-4)  (4,-1)  (1,-3)  (3,-5)  (5,-2)  (2,0)
 (0,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)  (5,0)
DCJ1:
 (0,-4)  (4,-1)  (1,-2)  (3,-5)  (5,-3)  (2,0)
 (0,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)  (5,0)
DCJ1:
 (0,-4)  (4,-1)  (1,-2)  (3,-5)  (5,0)  (2,-3)
 (0,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)  (5,0)
DCJ1:
 (0,-5)  (4,-1)  (1,-2)  (3,-4)  (5,0)  (2,-3)
 (0,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)  (5,0)
DCJ1:
 (0,-1)  (4,-5)  (1,-2)  (3,-4)  (5,0)  (2,-3)
 (0,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)  (5,0)
DCJ3:
 (0,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)  (5,0)
 (0,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)  (5,0)
Resultado:
 (0,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)  (5,0)
 (0,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)  (5,0)
```

Suponiendo los genomas con cromosomas circulares g1=(3,5,-1,4,-2) y g2=(1,2,3,4,5) se tiene

```
Genomas:
 (-2,-3)  (3,-5)  (5,1)  (-1,-4)  (4,2)
 (5,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)
DCJ1:
 (-2,-3)  (3,-5)  (-4,1)  (-1,5)  (4,2)
 (5,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)
DCJ1:
 (-2,1)  (3,-5)  (-4,-3)  (-1,5)  (4,2)
 (5,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)
DCJ1:
 (-2,1)  (3,-5)  (2,-3)  (-1,5)  (4,-4)
 (5,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)
DCJ1:
 (-2,1)  (4,-5)  (2,-3)  (-1,5)  (3,-4)
 (5,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)
DCJ3:
 (5,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)
 (5,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)
Resultado:
 (5,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)
 (5,-1)  (1,-2)  (2,-3)  (3,-4)  (4,-5)
```

Suponiendo los genomas como cromosomas lineales y circulares g1=(0,1,-2,3,4,0) (5,6,7) y g2=(0,1,2,3,4,0) (0,5,6,7,0)