Mapeo de Secuencias:

Mapeo con Unique Probes

Mapeo de Secuencias:

Una “Probe” es una sonda o secuencia de oligonucleotidos de longitud 8 a 25 que se usa para identificar objetivos en una muestra biologica através de hibridaciones usando Microarrays de ADN. Una sonda se llama unique probe si hibridiza solamente a un objetivo especifico de otro modo, se denomina una non-unique probe.

Ya que las unique probes separan los objetivos de manera determinante, identificar la presencia de objetivos en una muestra usando unique probes is directa.
Sinembargo, encontrar unique probes para cada objetivo es una tarea dificil por la gran similitud que puede existir entre objetivos relacionados.

En ese caso se utilizan Non -Unique Probes

Mapeo de Secuencias:

Mapeo con Unique Probes:

Un STS (Secuence Tagged Site) Probe es un filtro que puede determinar de forma unica si una secuencia corta de ADN aparece a lo largo de otra secuencia más larga. El filtro puede identificar la la existencia de la secuencia corta, pero no da información sobre su posición. Las secuencias cortas son de 200 a 300 bp para minimizar probabilidades de error.

Problema: Ordenar los clones para cubrir la secuencia original

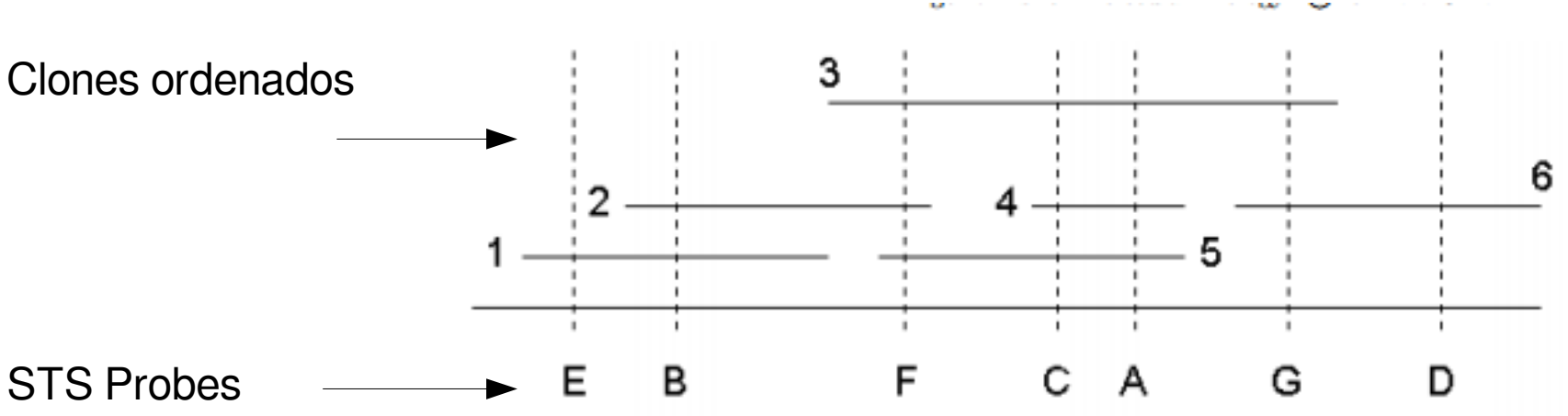


Figure 9.3: Ordered clones and several STS probes.

Mapeo de Secuencias:

Mapeo con Unique Probes:

El Problema de Mapeo con Unique Probes:

Entrada: Un conjunto de elementos U (Probes o sondas) y una colección de subconjuntos $\varphi = \{ S_1, S_2 \ldots . , S_n \}$ tal que $S_i \subset U$

Encontrar: el conjunto de $\Pi(\varphi)$ todas las permutaciones en U en las cuales cada S_i es contiguo.

Mapeo de Secuencias:

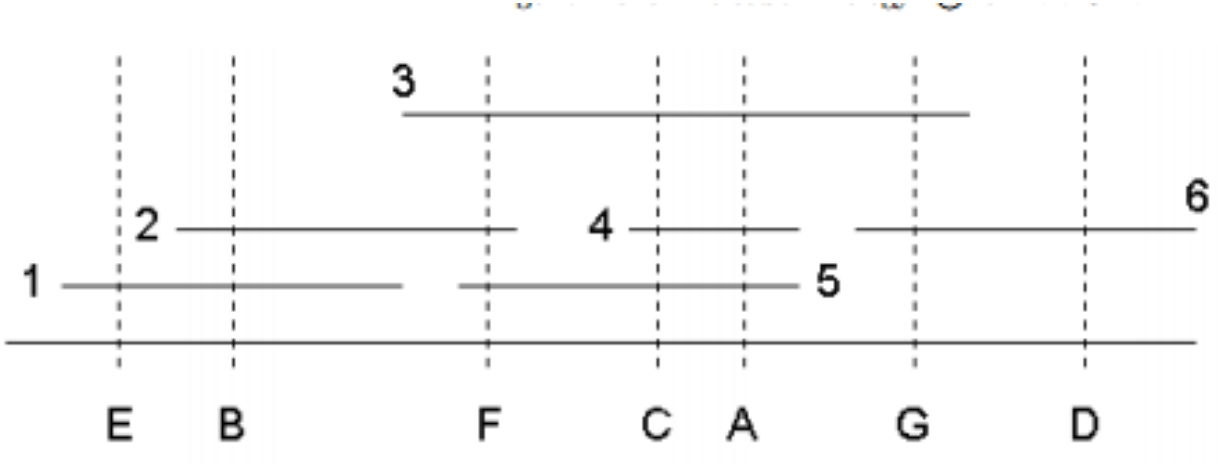


Figure 9.3: Ordered clones and several STS probes.

Clone / Probe	A	B	C	D	E	F	G
1	0	1	0	0	1	0	0
2	0	1	0	0	0	1	0
3	1	0	1	0	0	1	1
4	1	0	1	0	0	0	0
5	1	0	1	0	0	1	0

La matriz resultante de STS para el ejemplo. Un uno en la fila i y columna j significa que el clon i contiene la sonda (Probe) j

Mapeo de Secuencias:

La matriz resultante de STS para el ejemplo

Clone / Probe	A	B	C	D	E	F	G
1	0	1	0	0	1	0	0
2	0	1	0	0	0	1	0
3	1	0	1	0	0	1	1
4	1	0	1	0	0	0	0
5	1	0	1	0	0	1	0
6	0	0	0	1	0	0	1

Table 9.1: Resulting STS matrix. 1 in line *i* column *j* denotes that clone *i* contains probe *j*.

Clone / Probe	E	B	F	C	A	G	D
1	1	1	0	0	0	0	0
2	0	1	1	0	0	0	0
3	0	0	1	1	1	1	0
4	0	0	0	1	1	0	0
5	0	0	1	1	1	0	0
6	0	0	0	0	0	1	1

Table 9.2: Solved STS matrix. The set of solved matrixes is equivalent to the PQ-Tree Algorithm solution.

La matriz de STS resuelta, para el ejemplo

Mapeo de Secuencias:

El Problema de Mapeo con Unique Probes:
Es equivalente al problema de redistribuir las columnas de la matriz resultante de STS de tal manera que los “1” en las columnas de la matriz sean consecutivos.

Esta propiedad se llama la propiedad de los 1 consecutivos - “C1P”

Una solucion lineal para este problema fue presentada por Booth y Lueker, la cual se basa en la representacion de todas las permutaciones con Arboles PQ (PQTrees)

Mapeo de Secuencias:

El algoritmo de los Arboles PQ (PQTrees):

El Arbol PQ es un árbol con raiz y ordenado. Las hojas del Arbol son los elementos de U (probes/ sondas).
Los nodos internos son de 2 tipos: P-nodes y Q-Nodes.
Un P-Node tiene sub-nodos T1...Tk para k >= 2 y representa a k subconjuntos de U (cada uno de los sub-nodos). Se sabe que cada subconjunto es un bloque consecutivo de elementos, pero el orden de los bloques es desconocido.
Un Q-Node Tiene subnodos T1...Tk para k >= 3 . representa un conjunto de k bloques que se sabe que aparecen en ese orden (o en orden totalmente invertido)

Mapeo de Secuencias:

El algoritmo de los Arboles PQ (PQTrees):

Como los nodos P y Q representan el orden permitido de sus hojas, algunos movimientos en el arbol son permitidos

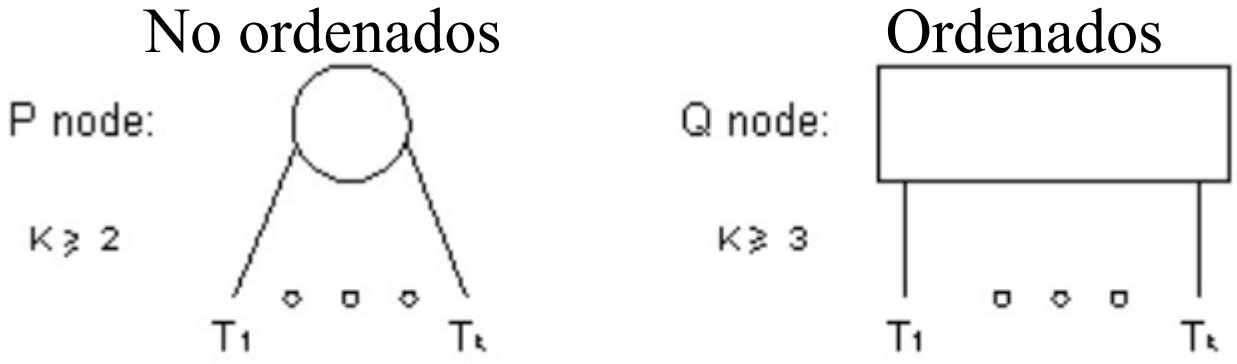


Figure 9.4: PQ-tree node types: we use circles and bars to denote P-nodes and Q-nodes, respectively.

Mapeo de Secuencias:

El algoritmo de los Arboles PQ (PQTrees):
Movimientos (Acciones) permitidas en el arbol:
1. Reordenamiento arbitrario de los sub-nodos de un P-node
2.Invertir el orden de los sub-nodos de un Q-Node.

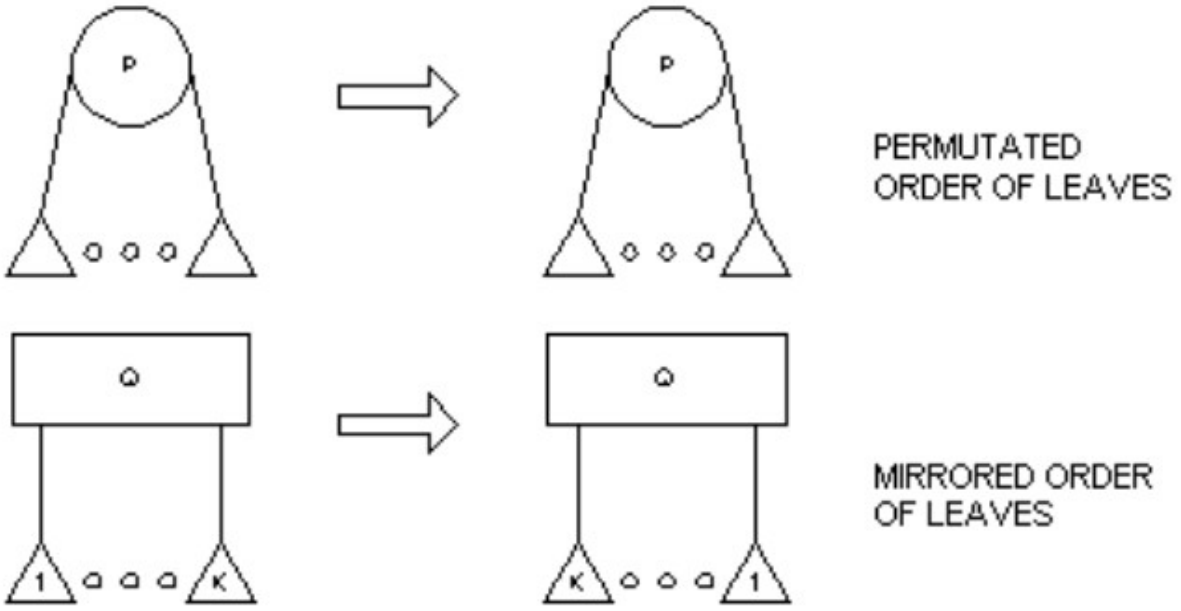
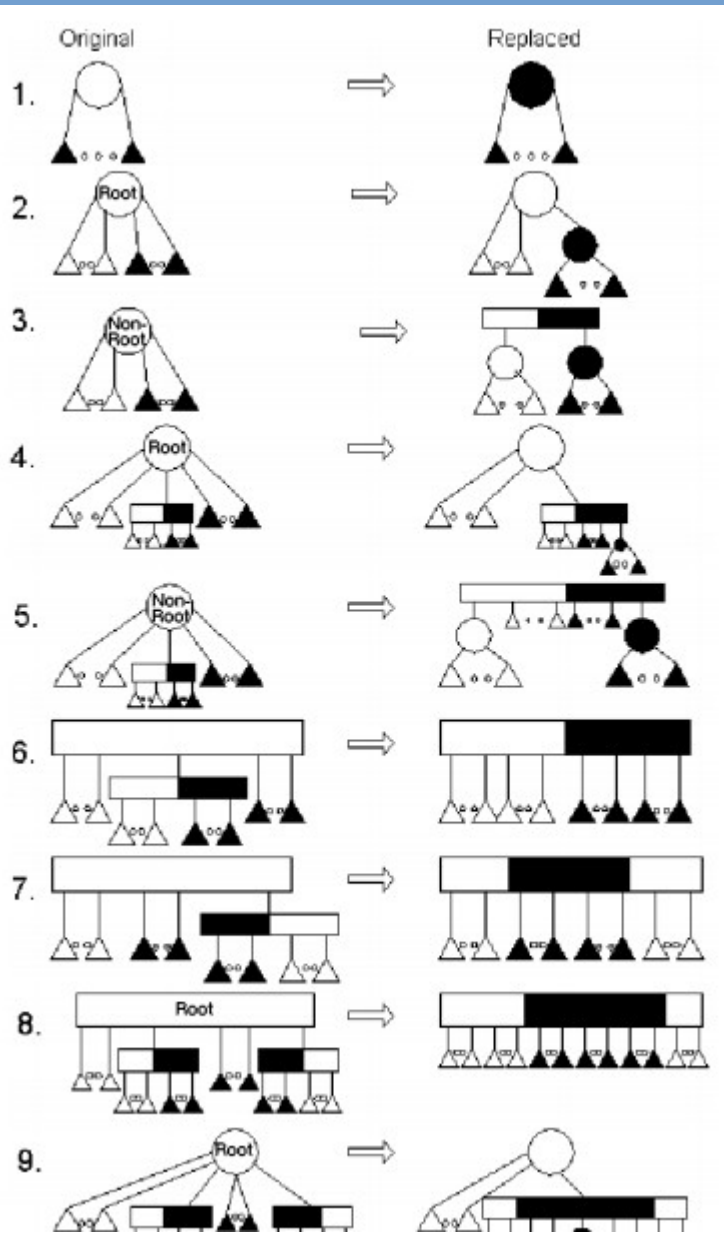


Figure 9.6: Legal transformations of a PQ-tree.

Mapeo de Secuencias:

De las acciones permitidas se desprende un set de reglas que se pueden utilizar para reordenar el PQ-Tree.

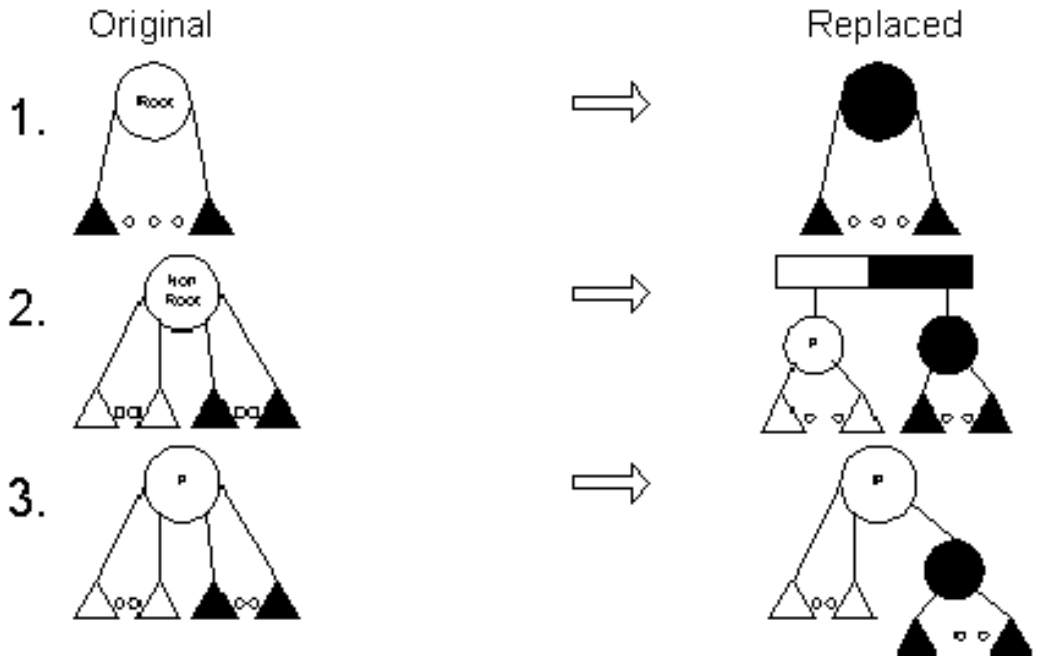
En la figura, las acciones de transformacion se deben realizar en el sub-arbol coloreado, afin de conseguir un arbol PQ valido.



Mapeo de Secuencias:

De las acciones permitidas se desprende un set de reglas que se pueden utilizar para reordenar el PQ-Tree.

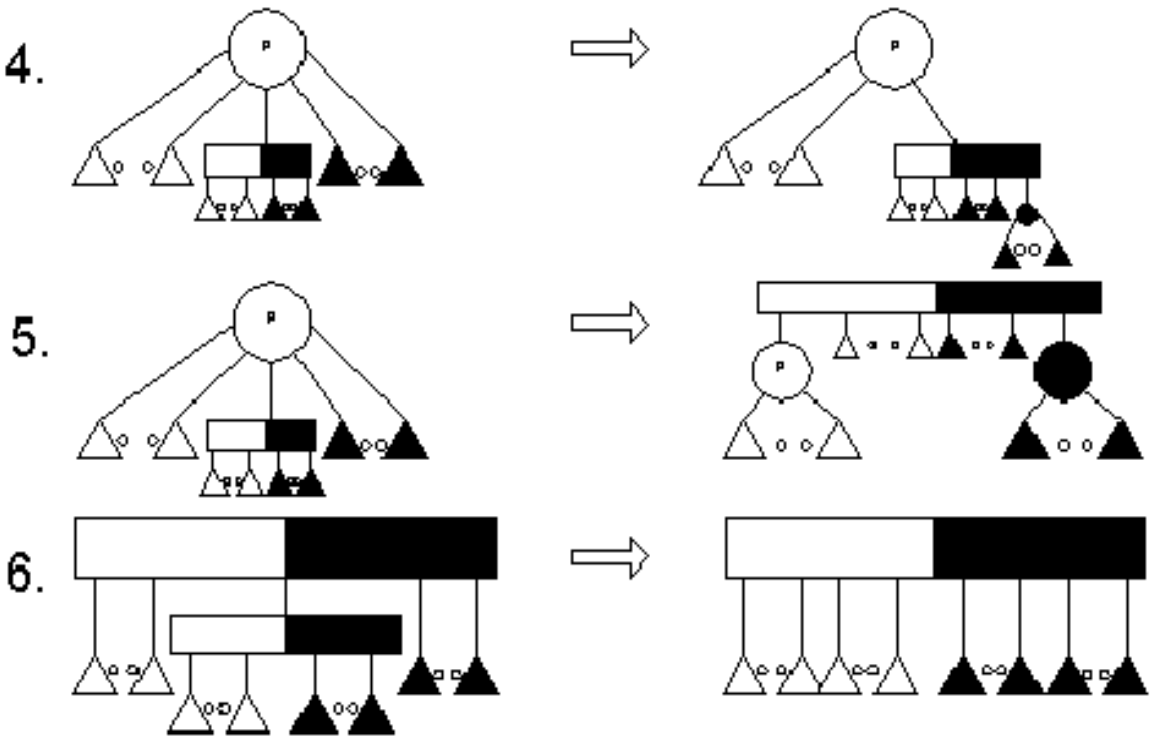
En la figura, las acciones de transformacion se deben realizar en el sub-arbol coloreado, afin de conseguir un arbol PQ valido.



Mapeo de Secuencias:

De las acciones permitidas se desprende un set de reglas que se pueden utilizar para reordenar el PQ-Tree.

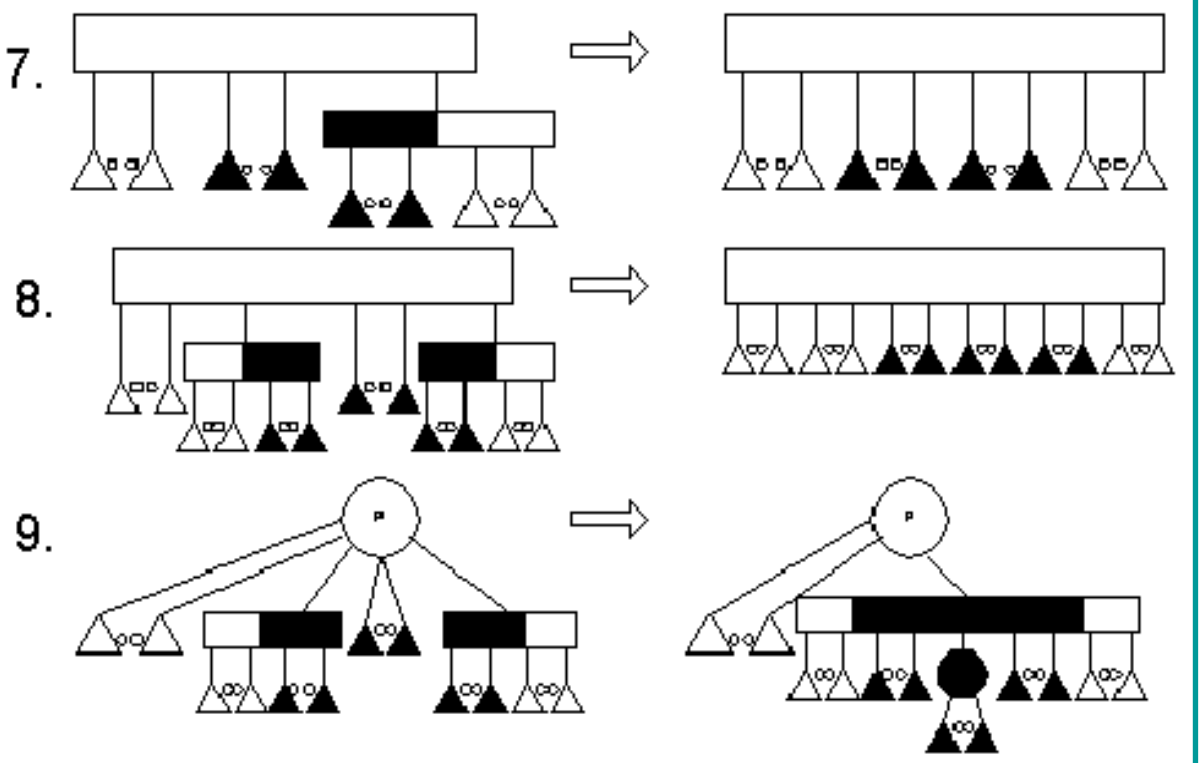
En la figura, las acciones de transformacion se deben realizar en el sub-arbol coloreado, afin de conseguir un arbol PQ valido.



Mapeo de Secuencias:

De las acciones permitidas se desprende un set de reglas que se pueden utilizar para reordenar el PQ-Tree.

En la figura, las acciones de transformacion se deben realizar en el sub-arbol coloreado, afin de conseguir un arbol PQ valido.



Mapeo de Secuencias:

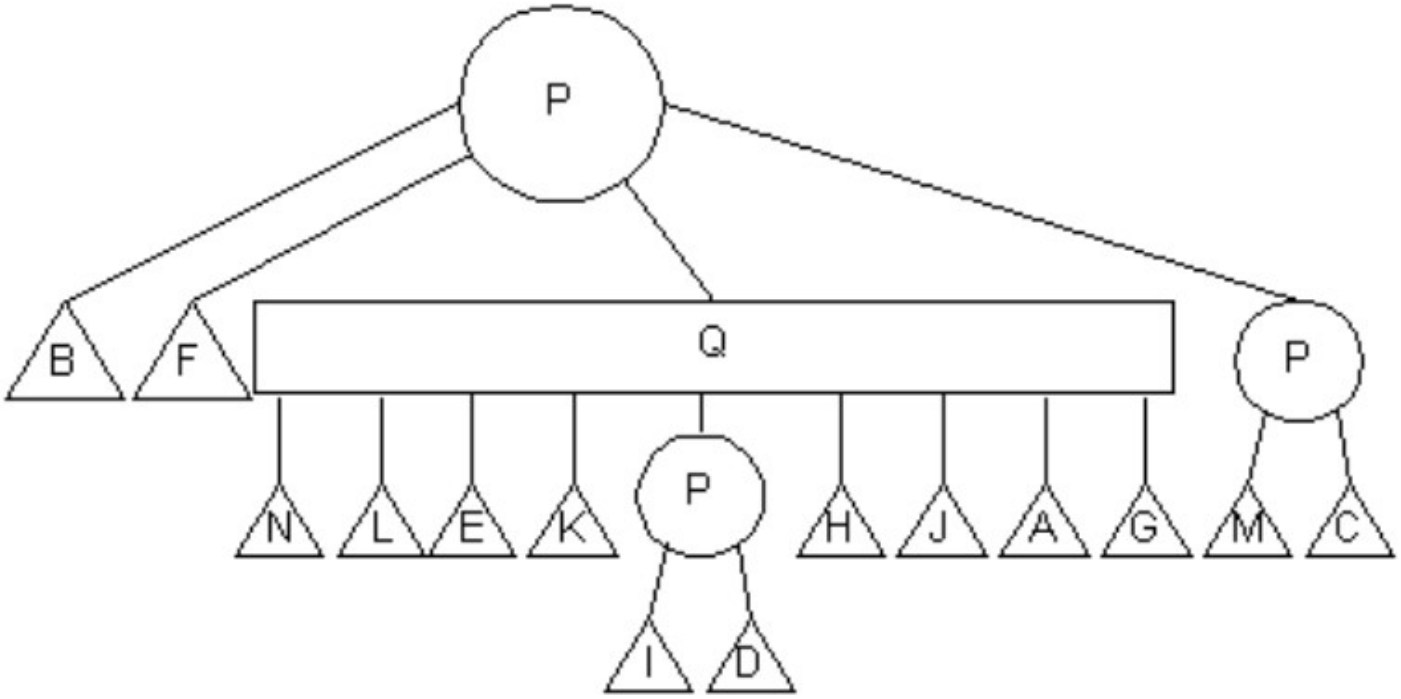
La Frontera de un arbol PQ es el conjunto de todas las hojas, en orden de izquierda a derecha.

2 Arboles PQ son Equivalentes (T ≡ T') si existe un conjunto de transformaciones permitidas que llevan de un arbol al otro

El conjunto de todas las fronteras de árboles equivalentes a **T se denomina Consistent(T),** donde

Consistent(T) = {Frontera(T') | T' ≡ T}

Mapeo de Secuencias:



Tree's frontier is: BFNLEKIDHJAGMC

Mapeo de Secuencias:

INPUT: *A set of elements U (probes) and a collection of subsets $\varphi = \{S_1, S_2, \ldots, S_n\}, \forall i : S_i \subseteq U$*

QUESTION: *Find the set $\Pi(\varphi)$ of all permutations over U along which every S_i is continuous.*

Booth y Lueker demostraron que Para cada U y φ (subconjunto de probes o sondas) existe un PQ-Tree donde consistent(T) = Π(φ)

Es decir, el problema de permutar las Probes sondas para conseguir la propiedad de los “1” consecutivos de la matriz de STS es equivalente a encontrar el arbol PQ que representa a la permutación Π(φ)

Mapeo de Secuencias:

Algoritmo del PQ -Tree para el mapeo de ADN con Unique probes (con sondas unicas)

1. Inicializar el árbol como un nodo P raiz con todos los elementos de **U** como sub nodos – hojas
2. Para i=1, ..., n : **reducir(T, S_i)**

El procedimiento **reducir(T,S_i)** retorna un árbol para cualquier permutación en **consistent(T)** en la cual S_i es continua

Mapeo de Secuencias:

Algoritmo del PQ -Tree para el mapeo de ADN con Unique probes (con sondas unicas)

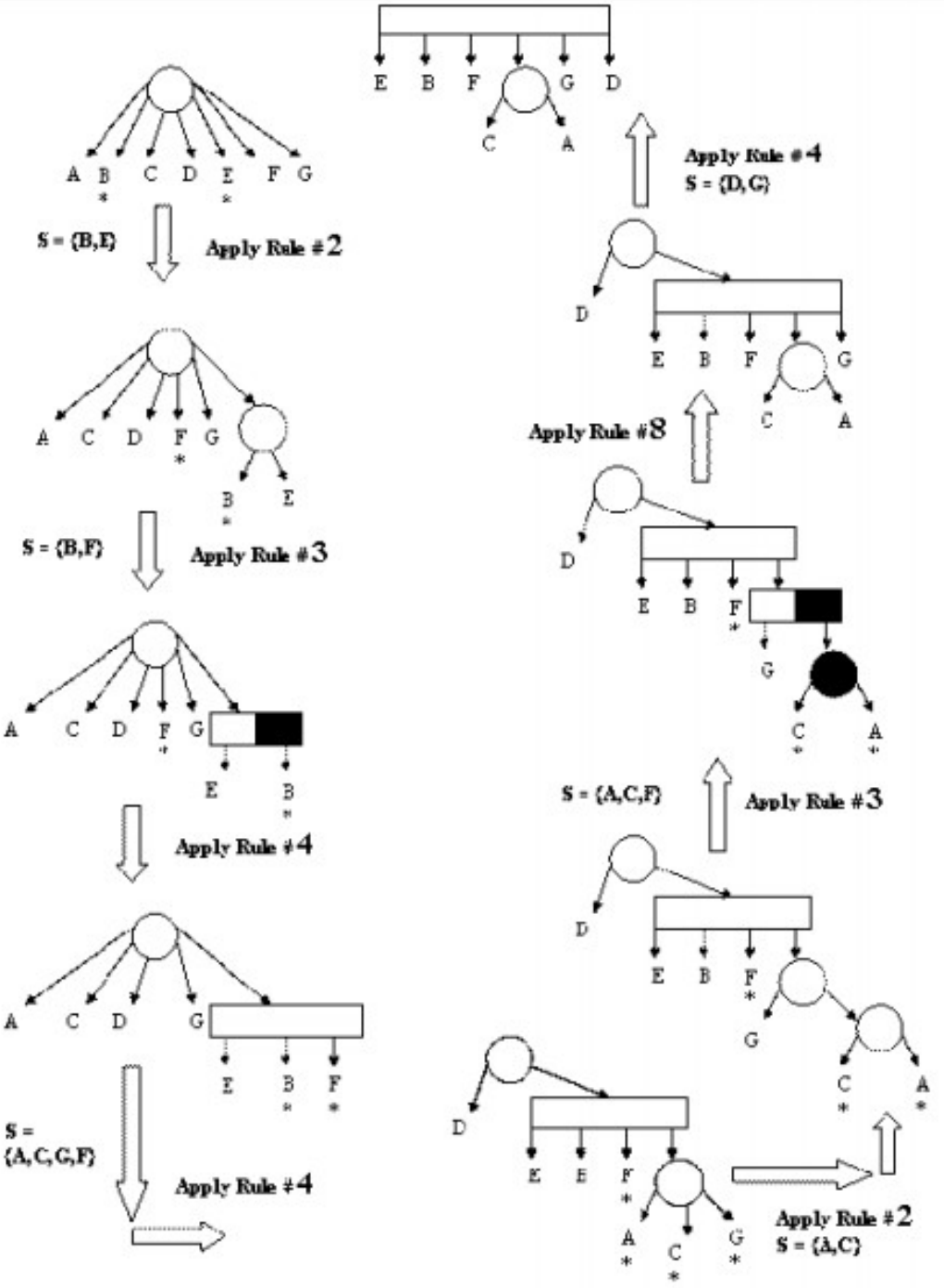
reducir(T, S_i):

1. Colorear todas las hojas S_i
2. Aplicar transformaciones para reemplazar T con un arbol PQ-tree equivalente en cuyas fronteras todas las hojas coloreadas son consecutivas
3. Identificar el nodo mas profundo **Root(T,S_i)** cuyo sub arbol contiene todas las hojas coloreadas
4. Aplicar las reglas de reemplazo permitidas en este sub arbol hasta llegar al nodo **Root(T,S_i)**

Mapeo de Secuencias:

Clone / Probe	A	B	C	D	E	F	G
1	0	1	0	0	1	0	0
2	0	1	0	0	0	1	0
3	1	0	1	0	0	1	1
4	1	0	1	0	0	0	0
5	1	0	1	0	0	1	0
6	0	0	0	1	0	0	1

Clone / Probe	E	B	F	C	A	G	D
1	1	1	0	0	0	0	0
2	0	1	1	0	0	0	0
3	0	0	1	1	1	1	0
4	0	0	0	1	1	0	0
5	0	0	1	1	1	0	0
6	0	0	0	0	0	1	1



Mapeo de Secuencias:

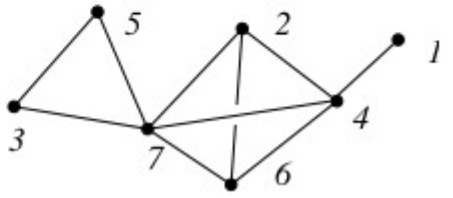
Se puede determinar si una matriz tiene la propiedad de los “1” consecutivos en tiempo polinomial.
El problema de los 1 Consecutivos tambien se puede expresar en funcion de matrices binarias y grafos de intervalos.
Ej: Dada una matriz binaria M podemos encontrar una permutacion de columnas de M tal que en cada fila todos los “1” ocurran en un intervalo consecutivo (un Bloque)?

1. Matrix:

$$\begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \left(\begin{array}{ccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{array}\right) \end{array}$$

2. Set system: $\{1,4\},\{3,5,7\},\{2,4,6,7\}$

3. Graph:



Mapeo de Secuencias:

Desafortunadamente, los experimentos de hibridizacion son propensos a generar errores de tipo falsos positivos y falsos negativos:
Falsos Positivos: Reportar que un clon contiene una secuencia especifica cuando en realidad no la contiene.
Falsos Negativos: Reportar que un clon no contiene una secuencia especifica cuando en realidad si la contiene
Quimeras: Son falsos clones construidos por diferentes piezas de ADN que vienen de diferentes partes (distintas) del genoma, juntando segmentos (probes) que son en realidad distantes.
Para estos casos la matriz de STS podria no tener una solución.

Clone / Probe	E	B	F	C	A	G	D
1	1	1	0	0	0	0	0
2	0	1	1	0	0	0	0
3	0	0	1	1	1	1	0
4	0	0	0	1	1	0	0
5	0	0	1	1	1	0	0
6	0	0	0	0	0	1	1

	probe						
clone	E	B	F	C	A	G	D
1	1	1	0	0	1	0	0
2	0	1	1	0	0	0	0
3	0	0	1	0	1	1	0
4	0	0	0	1	1	0	0
5	0	0	1	1	1	0	0
6	1	0	0	0	0	1	1

La matriz correcta y una matriz ordenada con un falso negativo en el clon 3, un falso positivo en el clon 1, y una posible quimera en el clon 6

Mapeo de Secuencias:

En la práctica, como es muy probable que este tipo de errores ocurran se considera el problema de los “1” optimos consecutivos.

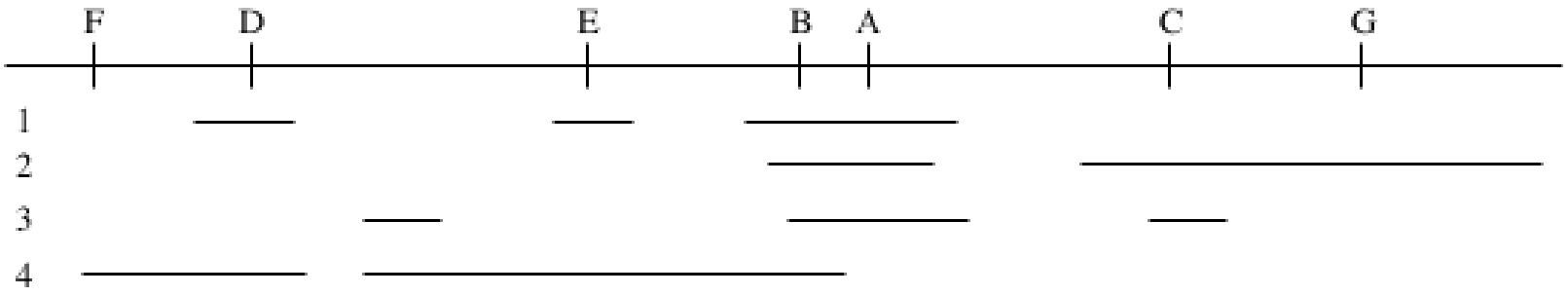
Ej : Dada una matriz binaria M Encontrar una permutación que minimiza el número de bloques de unos consecutivos.

Este problema es de solución NP- hard , sinembargo es similar a resolver un conocido problema de grafos, el problema del “Traveling salesman”, para el cual existen algoritmos eficientes de aproximación.

Veamos el ejemplo de mapeo de hibrido por radiacion:
En este experimento, un cromosoma (p.ej humano) es irradiado y partido en un pequeño numero de fragmentos. Estos fragmentos que no se traslapan se implantan en una celula (p.Ej. De un hamster) y la replicación produce una linea celular. Cada celula, tendrá entonces un grupo de 5 a 10 grandes fragmentos que no se traslapan del ADN objetivo. Este procedimiento se repite varias veces, con diferentes resultados. Al final se determina que linea de celulas se hibrida a que secuencias (probes).

Mapeo de Secuencias:

Mapeo de hibrido por radiacion:



	probe						
cell line	E	B	F	C	A	G	D
1	1	1	0	0	1	0	1
2	0	1	0	1	1	1	0
3	0	1	0	1	1	0	0
4	1	1	1	0	0	0	1

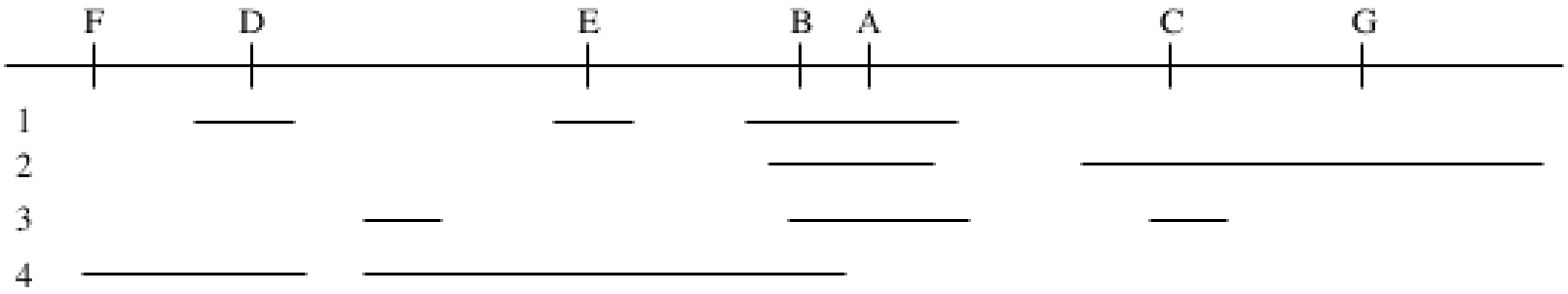
Como encontrar la correcta permutacion de secuencias (probes)?
Si asumimos que las probes que estan cerca la una a la otra en la secuencia (genoma) objetivo, tienen mas probabilidad de que se encuentren en el mismo segmento (dentro del conjunto dado). Entonces, tambien se busca minimizar el numero total de bloques de unos consecutivos.

Mapeo de Secuencias:

Solucion del Problema del vendedor viajero para minimizar el número de bloques de unos cosecutivos:

1. Se define un Grafo etiquetado G=(V,E) con V={s, p_1,...,p_k } donde p_i es un nodo para cada secuencia (probe) “i” y “s” es un nodo especial.
2. E contiene una arista desde “s” hasta cada “p_i” y una rama para cada par de secuencias (probes)
3. El peso de las aristas de “s” a los p_i es el número de unos en la columna correspondiente de la matriz.
4. El peso de cualquier otra arista (p_i ,p_j) es la distancia de Hamming entre las columnas que corresponden a p_i y a p_j

Mapeo de Secuencias:



	probe						
cell line	E	B	F	C	A	G	D
1	1	1	0	0	1	0	1
2	0	1	0	1	1	1	0
3	0	1	0	1	1	0	0
4	1	1	1	0	0	0	1

Para este caso tomaremos una sub matriz del ejemplo anterior mostrado, su matriz de distancias:

c/p	E	B	C	A
1	1	1	0	1
2	0	1	1	1
3	0	1	1	1
4	1	1	0	0

$\Rightarrow$

H	E	B	C	A
E	0	2	4	3
B		0	2	1
C			0	1
A				0

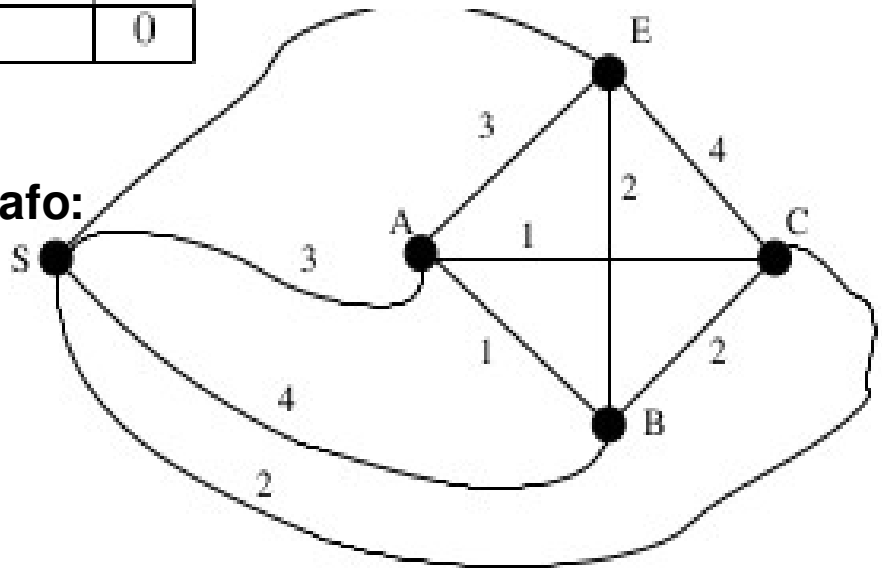
Mapeo de Secuencias:

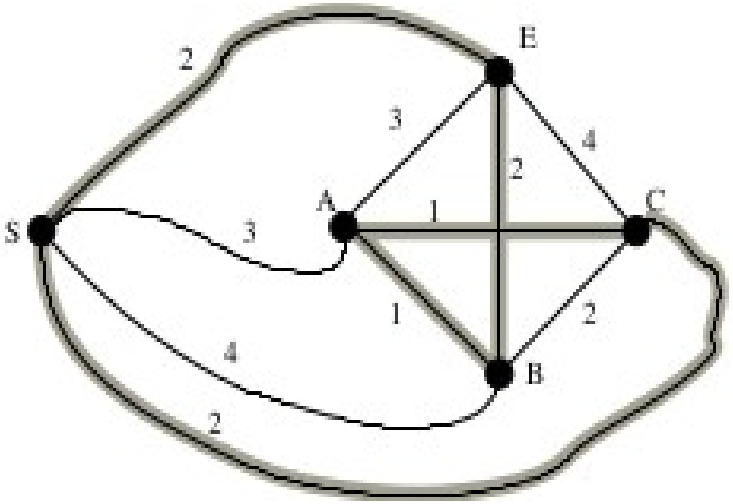
c/p	E	B	C	A
1	1	1	0	1
2	0	1	1	1
3	0	1	1	1
4	1	1	0	0

$\Rightarrow$

H	E	B	C	A
E	0	2	4	3
B		0	2	1
C			0	1
A				0

La matriz se traduce en el siguiente grafo:





Una ruta optima seria: S,C,A,B,E,S:

ALGORITMO DE DIJKTRA'S

Encuentra la ruta mas corta de un nodo de la red (nodo origen) a cualquier otro nodo, cuando los costos en los arcos (distancias) son no negativos.Los nodos se marcan con marcas Temporales y Permanentes, comenzando por el nodo origen. Un nodo tiene una marca Permanente si se ha encontrado la menor distancia a ese nodo. Un nodo j tiene marca temporal si existe el arco (i, j) y el nodo i tiene marca Permanente.

ALGORITMO DE DIJKTRA'S

La marca del nodo j es de la forma $[u_j,i]=[u_i+c_{ij},i]$, donde u_i es la distancia mas corta del nodo origen al nodo i con marca Permanente y c_{ij} el costo del arco (i,j). Los nodos que no pueden alcanzarse directamente a partir de un nodo con marca Permanente tendrán marca Temporal igual a ∞.

ALGORITMO DE DIJKTRA'S

Sea i=1 el nodo origen

Paso 0: marcar el nodo origen con [0,0], i=1, P={1}, T={2,3, …n}.

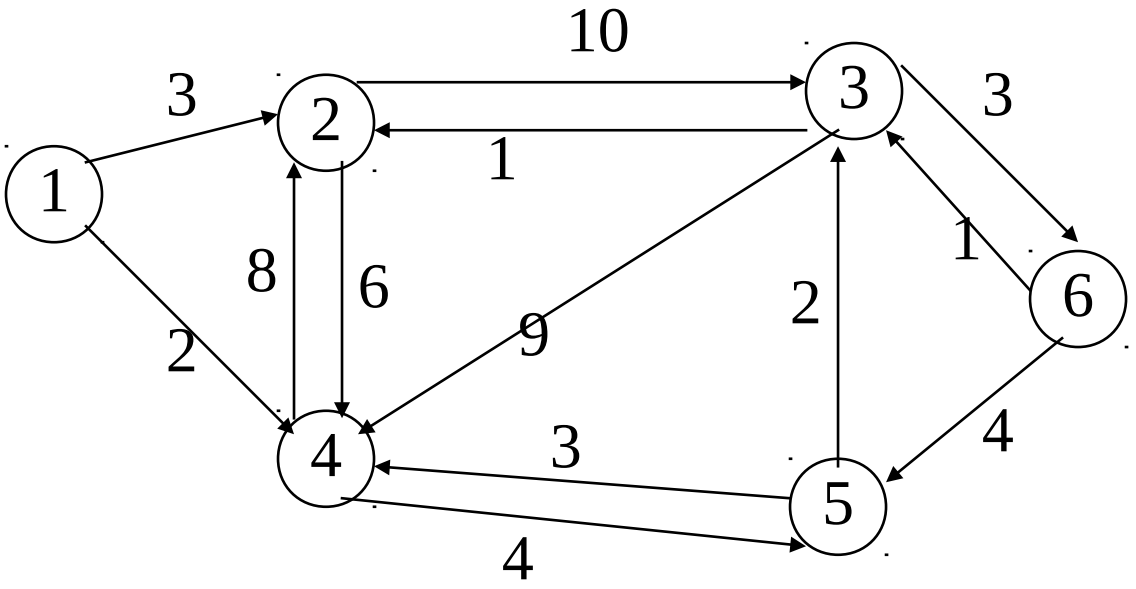
Paso 1: $\forall\, j \in T$ marcar $[u_j,,i]=[u_i+c_{ij},i]$. Si el nodo j tiene marca temporal $[u_j,k]$ y $u_i+c_{ij}<u_j$ reemplazar $[u_j,k]$ por $[u_i+c_{ij},i]$.

Paso 2:hallar $k \in T$ tal que $c_{ik}=\min\{c_{ij}, j \in T\}$, hacer, T=T-{k}, P=P+{k}. Marcar el nodo k en forma permanente. Si T=Ø parar, sino pasar al Paso 1.

EJEMPLO

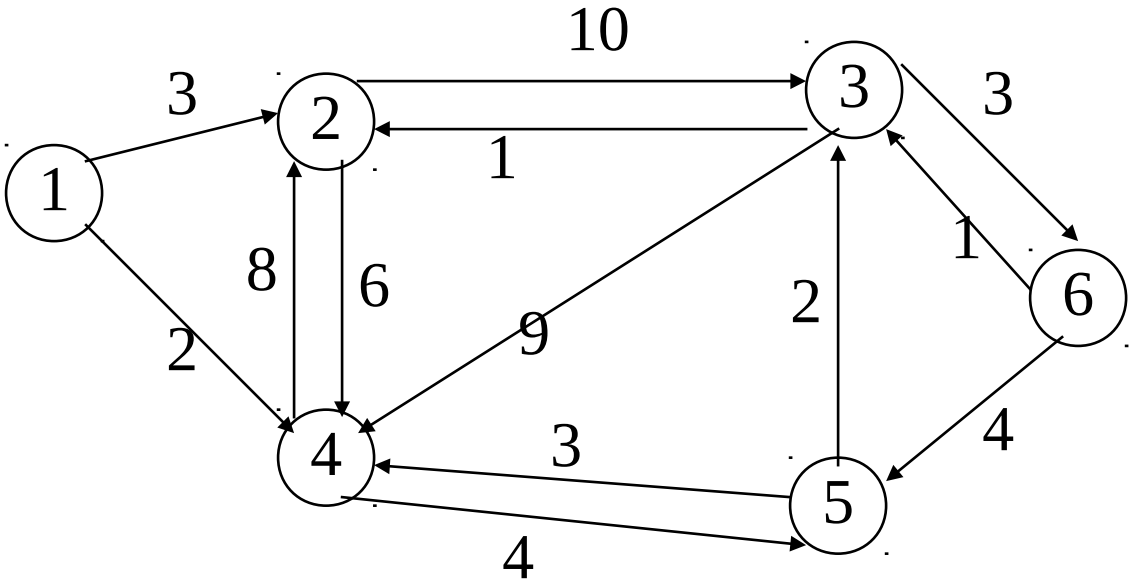
Los nodos de la red representa las estaciones de transbordo de un sistema de transporte en una ciudad. Los arcos representan las rutas posibles y las distancias representan el tiempo de recorrido que depende de las paradas. El origen está en el nodo 1 y en el nodo 6 se encuentra el final del recorrido. Se quiere encontrar la ruta mas corta del origen a cada nodo de transbordo y en particular la ruta mas corta al destino final.

RED



SOLUCIÓN

NODO	1	2	3	4	5	6	T={1,2,3,4,5,6},P={∅}
Iter 1	$[0,0]_p$	∞	∞	∞	∞	∞	T={2,3,4,5,6},P={1}
Iter 2	$[0,0]_p$	[3,1]	∞	$[2,1]_p$	∞	∞	T={2,3,5,6},P={1,4}
Iter 3	$[0,0]_p$	$[3,1]_p$	∞	$[2,1]_p$	[6,4]	∞	T={3,5,6},P={1,4,2}
Iter 4	$[0,0]_p$	$[3,1]_p$	[13,2]	$[2,1]_p$	$[6,4]_p$	∞	T={3,6},P={1,4,2,5}
Iter 5	$[0,0]_p$	$[3,1]_p$	$[8,5]_p$	$[2,1]_p$	$[6,4]_p$	∞	T={6},P={1,4,2,5,3}
Iter 6	$[0,0]_p$	$[3,1]_p$	$[8,5]_p$	$[2,1]_p$	$[6,4]_p$	$[11,3]_p$	T={∅},P={1,4,2,4,3}



SOLUCIÓN

Para determinar la ruta mas corta desde el nodo origen a cualquier otro nodo se procede como sigue:

Partiendo del nodo terminal escogido (k) buscar en la marca el nodo adyacente $[u_k,j]$, es decir el nodo j.Proceder de igual manera hacia atrás en la red. La distancia mínima es u_k

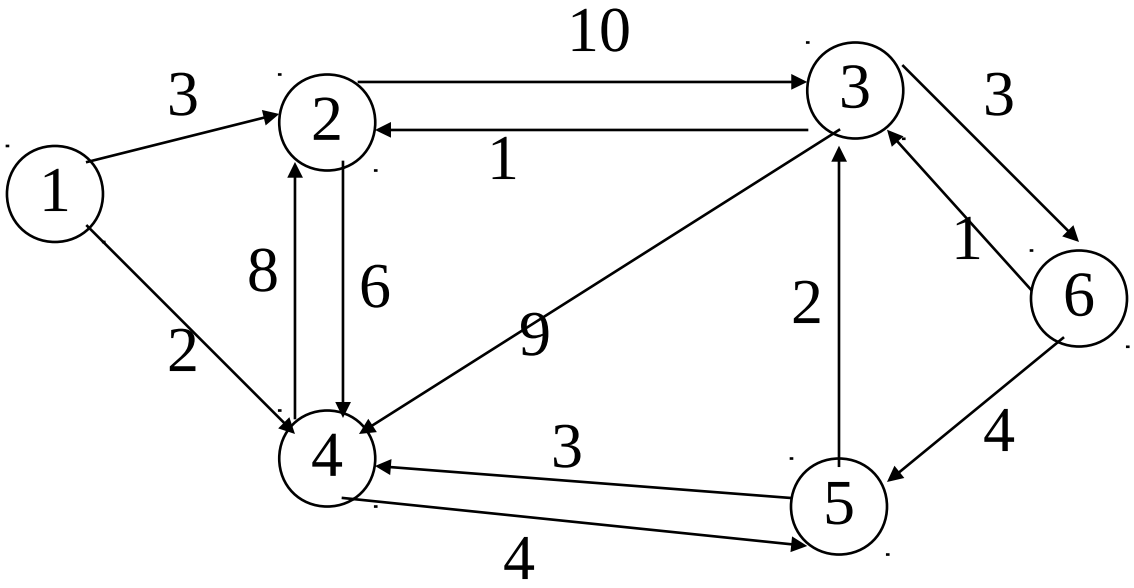
SOLUCIÓN

En el ejemplo, la ruta más corta del nodo origen al nodo 6 tiene una distancia igual a 11 y la ruta es:

1,4,5,3,6.

La ruta mas corta al nodo 3 es:

1, 4,5,3 con distancia igual a 8



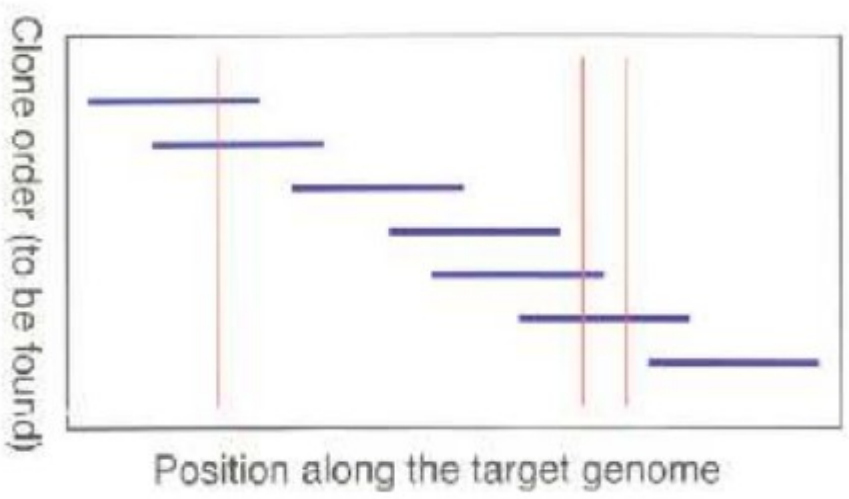
Mapeo de Secuencias:

Mapeo con Non -Unique Probes

Mapeo de Secuencias: Mapeo con Non -Unique Probes

Podemos utilizar non-unique probes para el problema de identificacion de secuencias. Dado que cada non-unique probe puede hibridar mas de un objetivo el problema de identificacion con non-unique probes se complica.

Es un problema NP-hard



Los clones son las lineas horizontales, la aparicion de una Non unique probe, es aleatoria, y en este caso aparece 3 veces

Mapeo de Secuencias: Mapeo con Non -Unique Probes

Ejemplo de un “Buen mapa” Las lineas horizontales son los clones las coordenadas x corresponden a su posicion en el genoma objetivo. La coodenada y indica el orden del clone en el mapa. Cada posicion en x esta cubierta por varios clones.

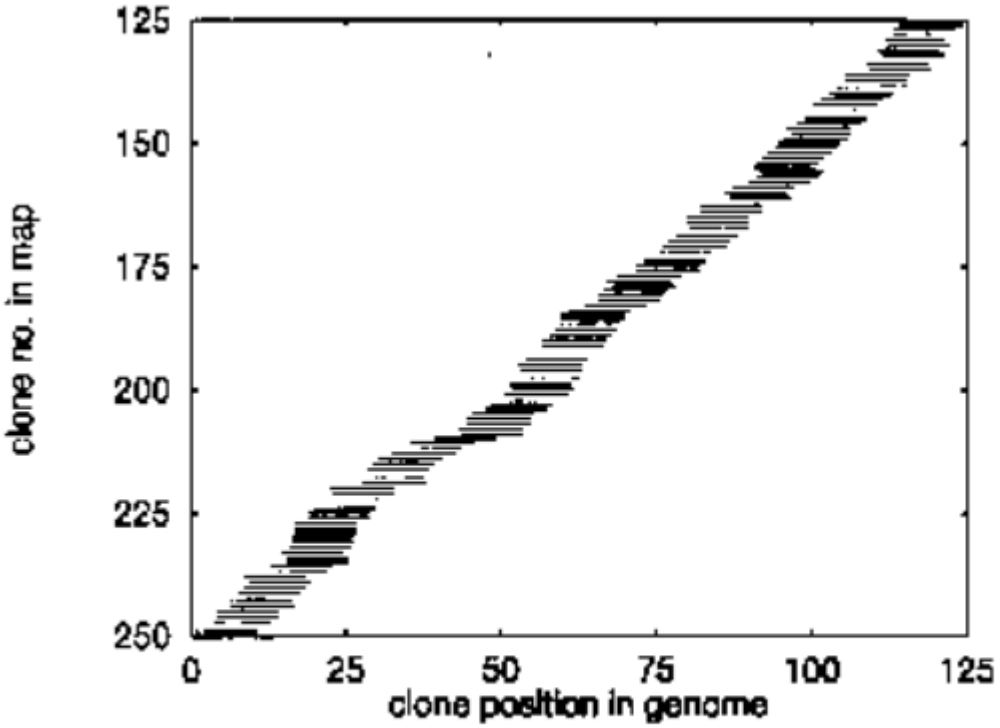


Figure 9.12: Source: [5].Physical (good) map example. The short horizontal lines are the clones with their x coordinates corresponding to the position on the target genome. The y coordinates correspond to the clone order in the constructed map. Note that each point on the target genome is covered by many clones.

Mapeo de Secuencias: Mapeo con Non -Unique Probes

Ejemplo de un “ mapa no tan bueno”. Debido a poca redundancia o errores o ruido en la informacion de las sondas (probes)

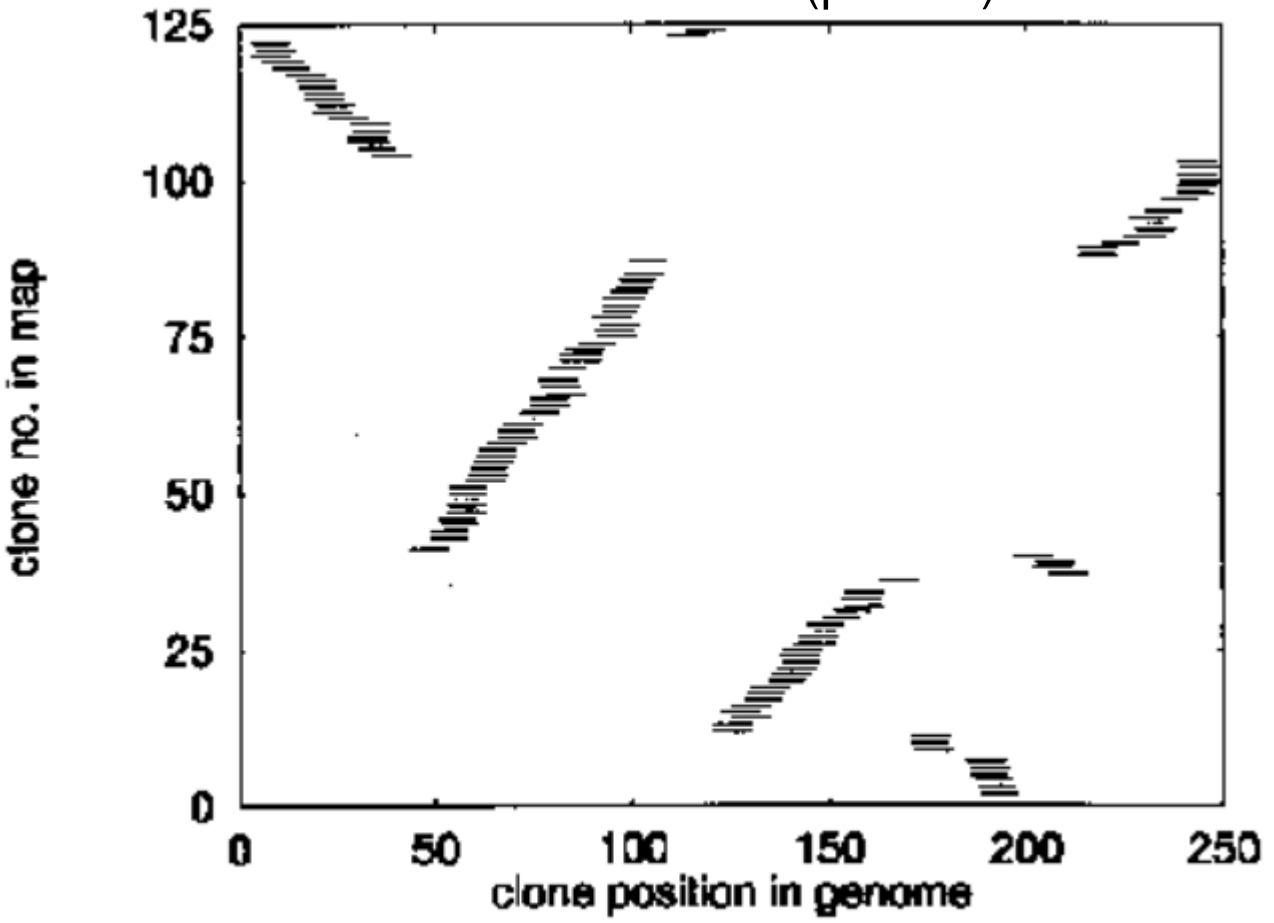


Figure 9.13: Source: [5].Physical (not so good) map example. Due to low redundancy factor (R) or noise with the probes data, we might produce this kind of maps. Most of the data within the eight contigs is correct, but still there are some contigs inversions.

Mapeo de Secuencias: Mapeo con Non -Unique Probes

Un algoritmo heuristico para el problema de mapeo con non-unique probes.

Dada una permutación de los clones Π encontrar la secuencia más corta S_{Π} que cubra los clones en el orden (segun puntos terminales de la izquierda) y dada por la permutacion Π es un algoritmo de tiempo polinomial. En este caso S_{Π} no es, necesariamente la secuencia mas optima.

Por ejemplo, dada la permutación (1,3,2) de los clones C_1={A,B,C}; C_2={B,C,D}; C_3={C,D,E}, la secuencia mas corta que ecubre la permutación es: S=ABCEDBC, La cual es más larga que S'=ABCDE que cubre la permutacIón (1,2,3)

Mapeo de Secuencias: Mapeo con Non -Unique Probes

Dada una permutación de los clones Π encontrar la secuencia más corta S_{Π} que cubra los clones en el orden (segun puntos terminales de la izquierda) y dada por la permutacion Π es un algoritmo de tiempo polinomial. En este caso S_{Π} no es, necesariamente la secuencia mas optima.

Por ejemplo, dada la permutación (1,3,2) de los clones C_1={A,B,C}; C_2={B,C,D}; C_3={C,D,E}, la secuencia mas corta que cubre la permutación es: S=ABCEDBC, La cual es más larga que S'=ABCDE que cubre la permutacIón (1,2,3)

La secuencia más corta que cubra los clones se puede definir como la secuencia más corta en { S_{Π} | Π es una permutación } Buscar todas las permutaciones dá un algoritmo con tiempo exponencial – Por las posibles n! permutaciones

Mapeo de Secuencias: Mapeo con Non -Unique Probes

Algoritmo Heuristico: - No retorna necesariamente la secuencia más corta.

1. comenzar con una permutación Π arbitraria. Y encontrar S_{Π} en tiempo polinomial, entonces cambiamos Π para obtener un número de permutaciones vecinas y para cada una de esas permutaciones Π' computamos S_{Π} ' en tiempo polinomial. Si S_{Π} ' es mas corta que S_{Π} para algun vecino Π' de Π, entonces hacemos Π = Π' y repetimos el proceso.

De otro modo, paramos.

Obviamente, despues de cada iteración obtenemos una secuencia que cubra los clones más corta que la anterior.

Mapeo de Secuencias: Mapeo con Non -Unique Probes

Algoritmo Heuristico: - No retorna necesariamente la secuencia más corta.

start with an arbitrary permutation of clones π
$length \leftarrow \infty$
compute s_π in polynomial time
while $|s_\pi| < length$
    $length \leftarrow |s|$
    compute a set of neighbors of π (polynomially many)
    for each neighbor π' of π
        compute $s_{\pi'}$ in polynomial time
        if $|s_{\pi'}| < |s_\pi|$
            then $\pi \leftarrow \pi'$
return s_π

Mapeo de Secuencias: Mapeo con Non -Unique Probes

Y como calculamos S_{Π} a partir de Π en tiempo polinomial?

Asumimos 3 proposiciones:

- Ningun Clone contiene a otro.
- El ADN esta completamente cubierto por los Clones
- Sin perdida de generalidad se asume que Π=(1,2,...,n), por lo que asumimos que las superposiciones de C1, C2,..., Cn aparecen en orden (de sus terminales izquierd

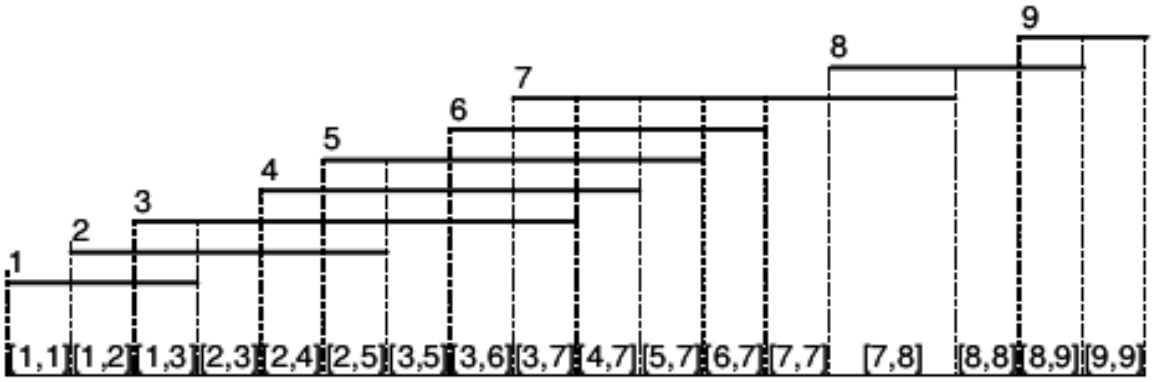


Figure 1: Intervals defined by an overlap

Mapeo de Secuencias: Mapeo con Non -Unique Probes

Cada intervalo [i,j] significa que los clones Ci y Cj se superponen.

Otro supuesto: La hibridacion esta libre de errores.

Los intervalos son libres de conflictos – si No hay clones contenidos uno en otro.

- Cada superposicion de C1,..., Cn define un conjunto de intervalos libres de conflictos
- Para cada cada conjunto de intervalos I libres de conflictos, existe un conjunto de intervalos libre de conflictos I' tal que I $\subseteq$ I' que define una superposicion de C1,..., Cn

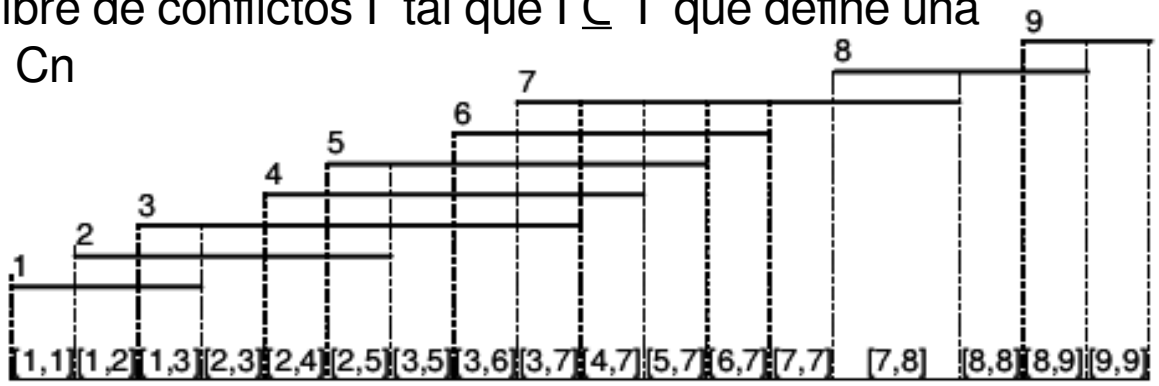


Figure 1: Intervals defined by an overlap

Mapeo de Secuencias: Mapeo con Non -Unique Probes

Los intervalos libres de conflictos en la matriz D
son:
[1,2],[6,8], [2,4] , [7,8], [1,1], [3,5], [7,9], [2,3], [3,6], [3,8]

Después de ordenar tenemos lo siguiente:
[1,1], [1,2], [2,3], [2,4], [3,5], [3,6], [3,8] , [6,8] [7,8], [7,9]
C A E B CD F G A B D

La secuencia CAEBCDFGABD es una secuencia que cubre C1, ..., Cn en orden ... pero será la la secuencia más corta? - Si

En cualquier secuencia que cubra C1,..Cn cada probe – tiene que aparecer un número de veces igual al número de sus intervalos.

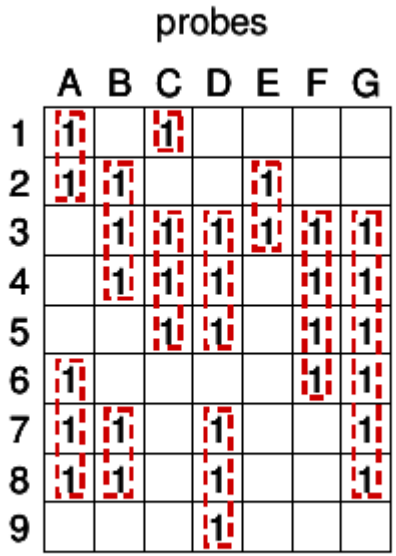


Figure 2: Conflict free intervals of D