Computational Biology
Lecture 14: Physical mapping by hybridization (cont.)
Saad Mneimneh

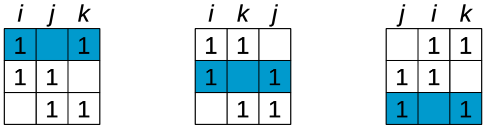
We have seen how mapping by hybridization can be abstracted into the shortest covering string problem SCS which is NP-hard when the probes are non unique. However, we showed that even when the probes are non unique, a polynomial time algorithm exists for computing a shortest covering string for a given permutation π of the clones. This polynomial time algorithm was used to develop a heuristic algorithm for the general SCS problem with non unique probes. Now we look at the SCS problem when probes are unique and we show that it can be solved in polynomial time.

SCS with unique probes

As for the case of non unique probes, we will assume that we have no hybridization errors. Under this assumption, (1) the length of the shortest covering string in equal to the number of probes (assume a probe hybridizes with at least one clone), and (2) the shortest ocvering string is therefore a permutation of the probes. Both (1) and (2) imply that the matrix D satisfy the consecutive 1's property $C1P$:

$C1P$: There exists a permutation of the columns of D (probes), such that all 1s in each row occur in consecutive positions.

Note that in the presence of hybridization errors, the $C1P$ property might not be satisfied, below is an example where we cannot permute the columns of D to make it $C1P$:



i	j	k
1		1
1	1	
	1	1

i	k	j
1	1	
1		1
	1	1

j	i	k
	1	1
1	1	
1		1

Figure 1: 3 possible permutations up to reversal $3!/2 = 6/2 = 3$

Therefore, assuming no hybridization errors, the shortest covering string reduces to finding a permutation of the columns of D to make it $C1P$. We will not try to explicitly construct the $C1P$ permutation, but we will find it by repeatedly identifying neighboring probes.

First, we list our assumptions. Let C_i be the set of clones that probe i hybridizes with (note that this is different from our previous notation where C_i denoted a clone). We have four assumptions:

- No hybridization errors: $C1P$ permutation exists
- Non-inclusion: No clone X contains another clone Y (i.e. the set of probes hybridized with X does not strictly contain the set of probes hybridized with Y)
- Connectedness (no gaps): for every partition of the set of probes into two non-empty sets A and B, there exist probes $i \in A$ and $j \in B$ such that $C_i \cap C_j \neq \emptyset$
- Distinguishability: $C_i \neq C_j$ for $i \neq j$

If connectedness is not satisfied, we can treat each connected subset of probes separately. If distinguishibility is not satisfied, we can treat undistinguishable probes as one. In non-inclusion is not satisfied, we can remove the clones that are completely contained in other clones. These can then be added by possibly performing some local permutations on the probes (we do not look at this detail here).

Now we reformulate the $C1P$ property in terms of the C_i's: Let $1...m$ be the correct order of the probes (i.e. the $C1P$ permutation).

Lemma: If $1 \leq i < j < k \leq m$ then $C_i \cap C_k \subseteq C_i \cap C_j$ and $C_i \cap C_k \subseteq C_j \cap C_k$.

Proof: If a clone $c \in C_i \cap C_k$ then $c \in C_j$ since the unique probe j lies between i and k and we have no hybridization errors; therefore $c \in C_i \cap C_j$ and $c \in C_j \cap C_k$.

Now we define the notion of a closest probe j to a given probe i.

For a given probe i and a set of probes P, a probe $j \in P$ is closest to i iff no other probe $k \in P$ lies between i and j.

Note that a given probe i can have up to two closest probes in P (either from left or from right). We will show that:

Theorem: Given a set of probes P and a probe i such that $C_i \cap C_j \neq \emptyset$ for some $j \in P$, probe $j \in P$ is closest to probe i if j has a minimum $|C_j|$ that maximizes $|C_i \cap C_j|$.

Proof: If j is the only probe in P that maximizes $|C_i \cap C_j|$, then by the lemma above, no probe $k \in P$ can lie between i and j since otherwise $C_i \cap C_j \subseteq C_i \cap C_k \Rightarrow |C_i \cap C_j| \leq |C_i \cap C_k|$ and j is not the only probe in P that maximizes $|C_i \cap C_j|$. Therefore, $j \in P$ is closest to i. If $|C_i \cap C_j|$ is maximized for two (or more) probes, then pick probe j with with minimum $|C_j|$. We will prove that j is closest by contradiction. Assume the opposite, i.e. j is not closest to i. Therefore, another probe k is closest to i. By the condition of the theorem, the maximum $|C_i \cap C_j| = |C_i \cap C_k| \neq 0$. The situation is depicted below without loss of generality (also consider one side of i without loss of generality).

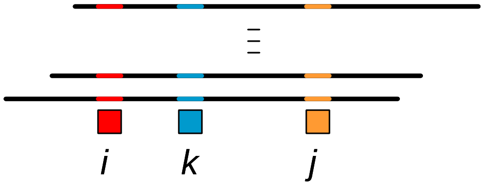


Figure 2: Assume for the sake of contradiction that j is not closest

By the **distinguishibility** assumption, $C_j \neq C_k$; therefore, there must be a clone c that

- hybridizes with j but not with k, or
- hybridizes with k but not with j

In the latter case, c can hybridize neither with i (otherwise $|C_i \cap C_k| > |C_i \cap C_j|$) nor with j, and hence will be contained in some other clone, which cannot be true by the **non-inclusion** assumption. Therefore, the former case is true and $|C_k| < |C_j|$, a contradiction to the choice of j.

The theorem above suggests an algorithm for determining the correct order of probes (i.e. the $C1P$ permutation). We maintain a correct partial permutation π of the probes, initially the permutation consists of an arbitrary probe. Let P be the set of probes that are not part of the permutation. Consider the first and last probes of the permutation π_{first} and π_{last}. By the connectedness assumption, there must be a probe $j \in P$ such that $C_{\pi_{first}} \cap C_j \neq \emptyset$ or $C_{\pi_{last}} \cap C_j \neq 0$.

Therefore, if $C_{\pi_{last}} \cap C_j \neq \emptyset$ for some $j \in P$, we find the closest $j \in P$ to π_{last} by the method of the theorem above, i.e. the j with the minimum $|C_j|$ that maximizes $|C_{\pi_{last}} \cap C_j|$. Once found, we know that no other probe $k \in P$ can lie between j and π_{last}. This means that j either follows π_{last} (closest from right), or preceeds π_{first} (closest from left and partial permutation built so far is correct). To choose among the two options, we let $P' = \{\pi_{first}, \pi_{last}\}$ and find the closest $k \in P'$ to j, i.e. find the $k \in P'$ with the minimum $|C_k|$ that maximizes $|C_j \cap C_k|$ and place j accordingly.

On the other hand, if $C_{\pi_{last}} \cap C_j = \emptyset$ for all $j \in P$, reverse the permutation and continue (making π_{first} the last probe of the permutation).

Algorithm

```
$\pi = \pi_{first} = \pi_{last} = i$ for any probe $i$
$P = \{$ all probes expect $i\}$
**repeat**
    **if** $C_{\pi_{last}} \cap C_j = \emptyset$ for all $j \in P$
      **then** reverse $\pi$
    choose $j \in P$ with minimum $|C_j|$ that maximizes $|C_{\pi_{last}} \cap C_j|$
    choose $k \in \{\pi_{first}, \pi_{last}\}$ with minimum $|C_k|$ that maximizes $|C_j \cap C_k|$
    **if** $k = \pi_{last}$
      **then** $\pi = \pi, j$
    **else** $\pi = j, \pi$
    $P = P - \{j\}$
**until** $P = \emptyset$
```

References

Pevzner P., Computational Molecular Biology, Chapter 3.