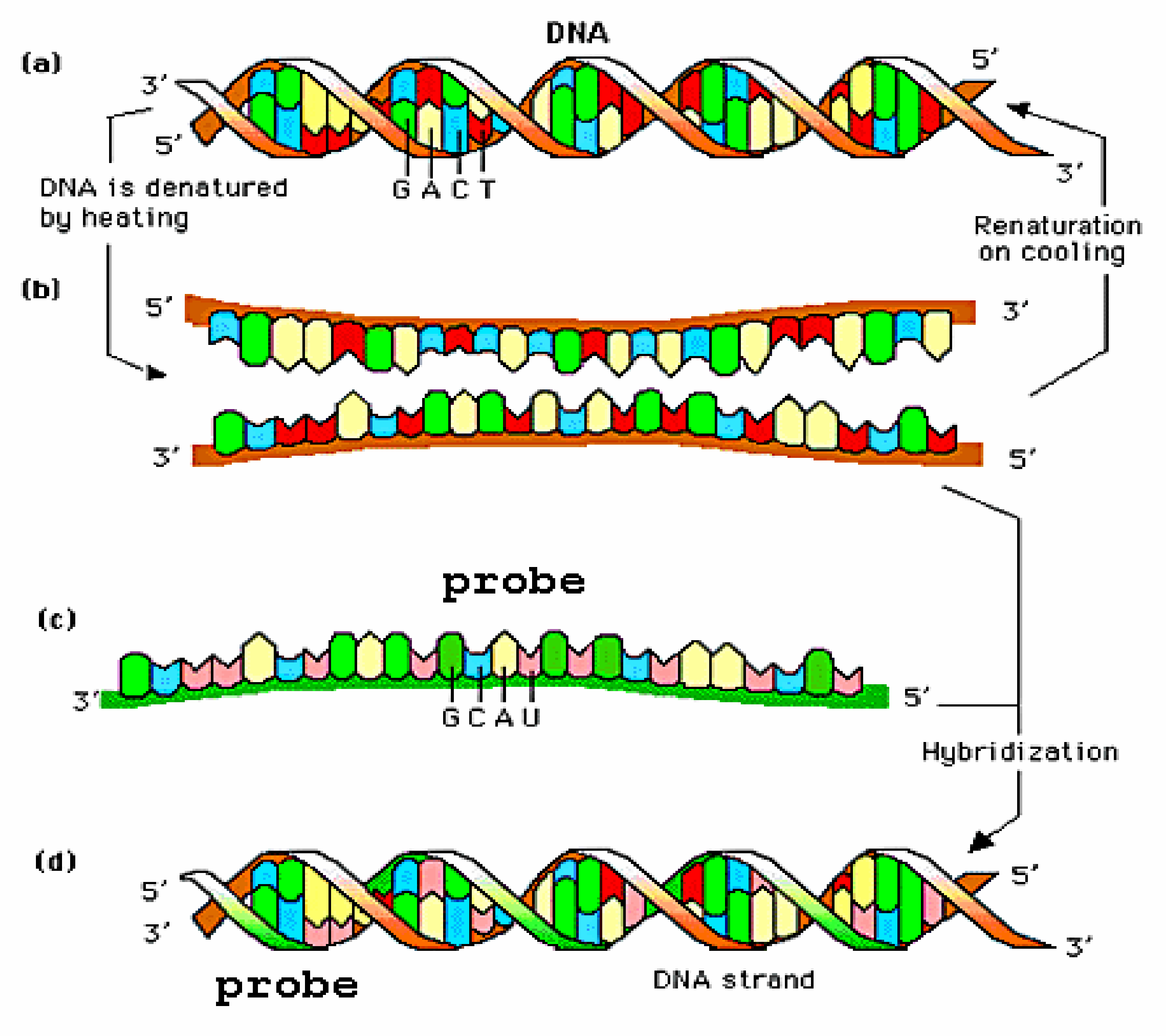
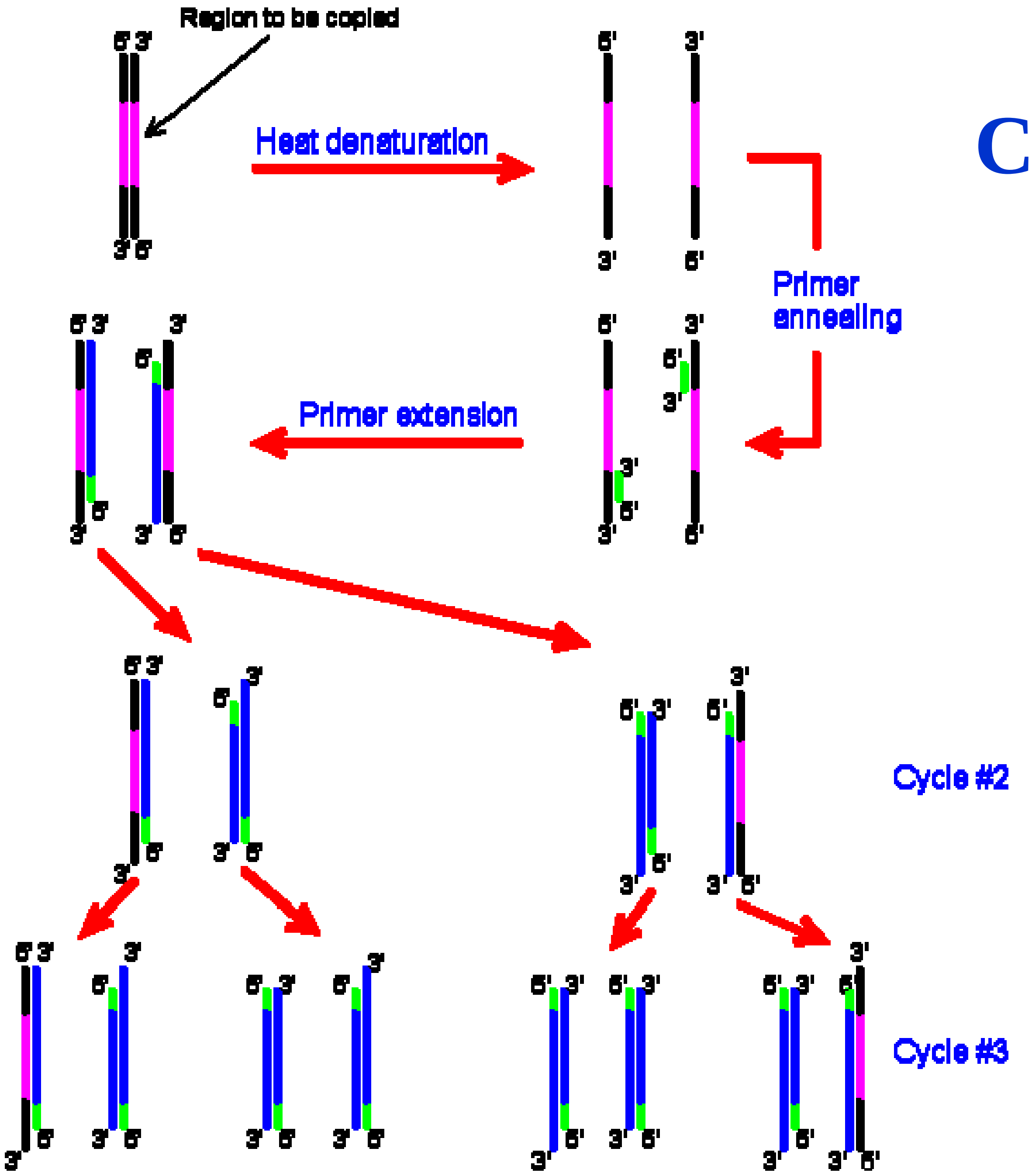
Nonunique Probe Selection and Group Testing

Ding-Zhu Du

DNA Hybridization



Polymerase Chain Reaction (PCR)



- cell-free method of DNA cloning

Advantages

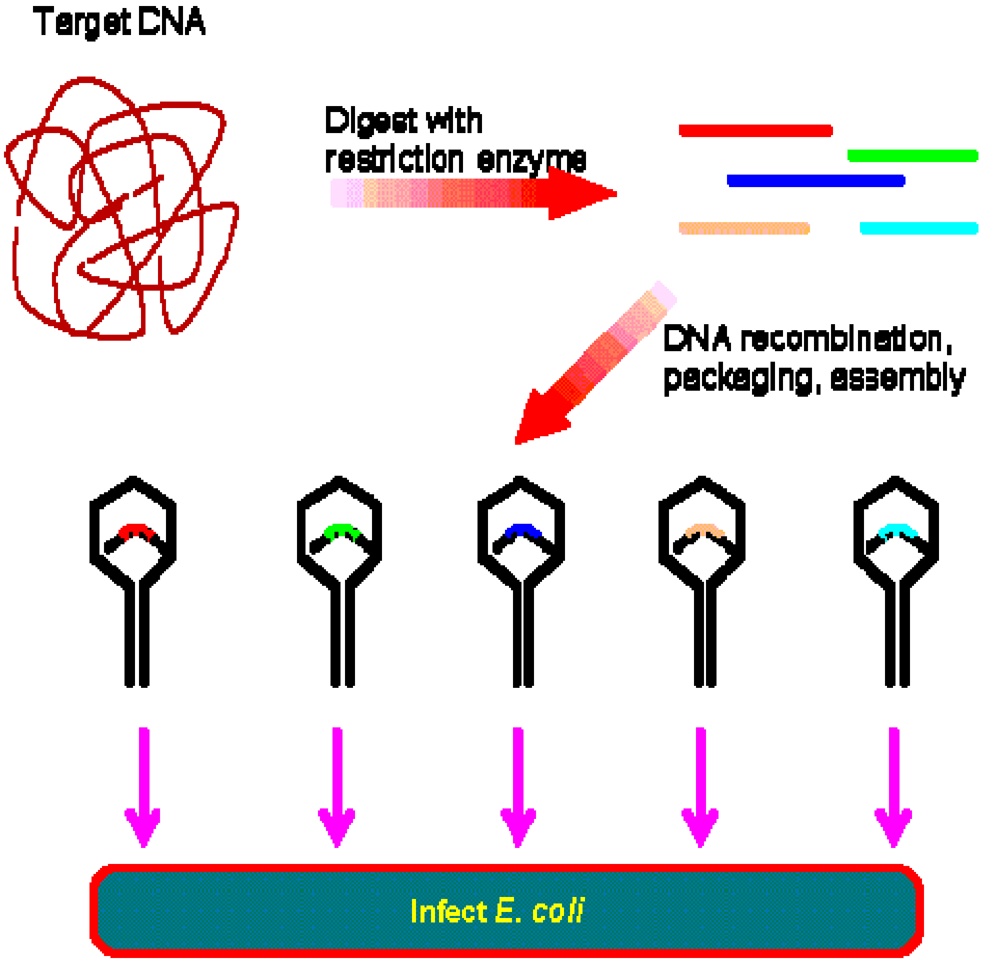
- much faster than cell based method
- need very small amount of target DNA

Disadvantages

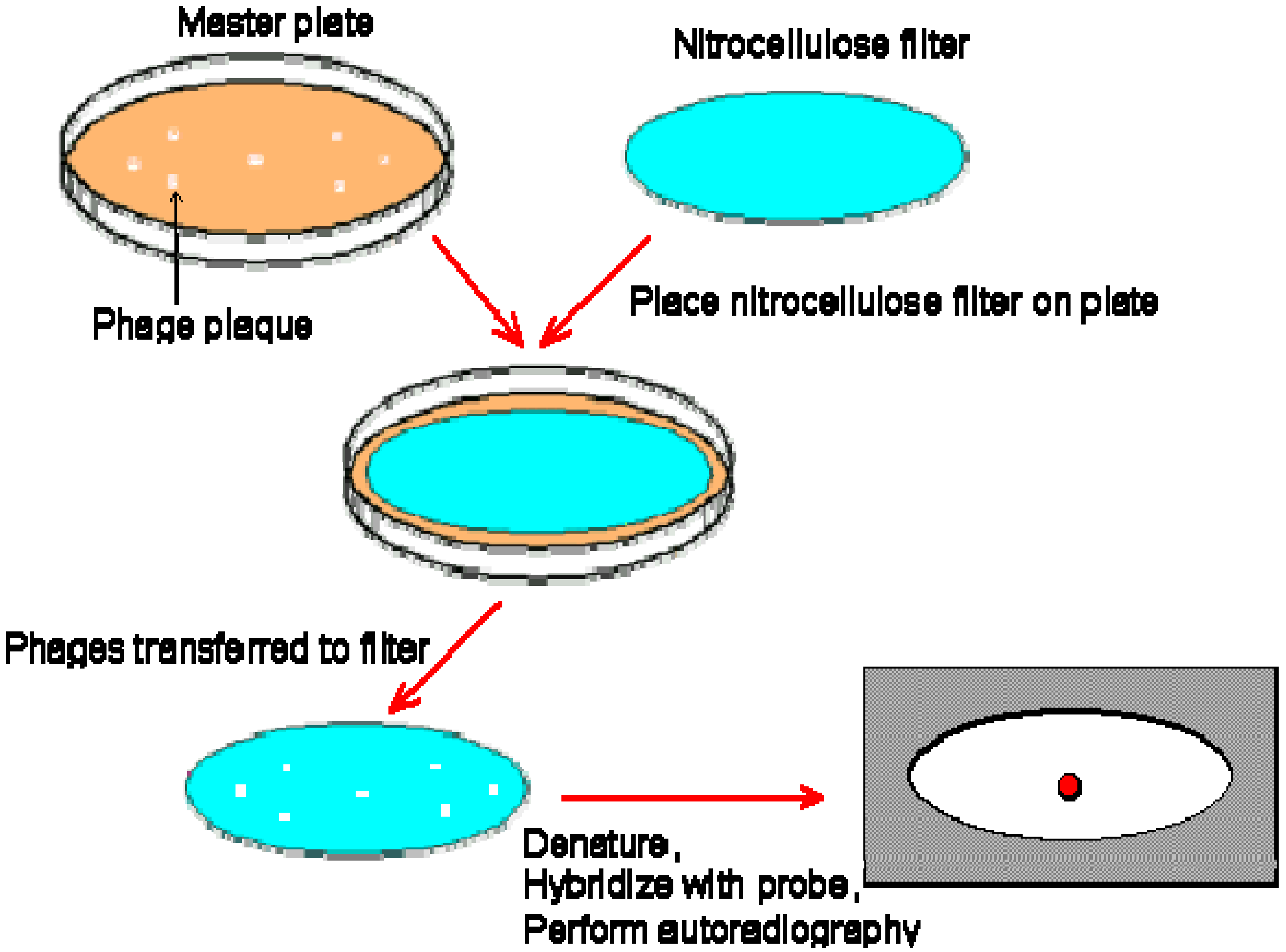
- need to synthesize primers
- applies only to short DNA fragments(<5kb)

Preparation of a DNA Library

- DNA library: a collection of cloned DNA fragments
- usually from a specific organism


Target DNA
Digest with restriction enzyme
DNA recombination, packaging, assembly
Infect *E. coli*

DNA Library Screening



Problem

- If a probe doesn’t uniquely determine a virus, i.e., a probe determine a group of viruses, how to select a subset of probes from a given set of probes, in order to be able to find up to *d* viruses in a blood sample.

Binary Matrix

viruses

$$
\begin{array}{llccccccccccccccccc}
 & & c_1 & c_2 & c_3 & & & & & & c_j & & & & & & & & c_n \\
 & p_1 & 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & p_2 & 0 & 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & p_3 & 1 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
\textbf{probes} & & 0 & 0 & 1 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & & . & & & & & & & & & & & & & & & & \\
 & & . & & & & & & & & & & & & & & & & \\
 & p_i & 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 1 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & & . & & & & & & & & & & & & & & & & \\
 & & . & & & & & & & & & & & & & & & & \\
 & p_t & 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0
\end{array}
$$

The cell (*i*, *j*) contains 1 iff the *i*th probe hybridizes the *j*th virus.

Binary Matrix of Example

virus

$$\begin{array}{llccccccccc}
 & & c_1 & c_2 & c_3 & & & & c_j & & \\
 & p_1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & p_2 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
 & p_3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
\text{probes} & & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
 & & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
 & & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1
\end{array}$$

Observation: All columns are distinct.

To identify up to *d* viruses, all unions of up to *d* columns should be distinct!

$\bar{d}$-Separable Matrix

viruses

$$
\begin{array}{cccccccccccccccccccc}
 & & c_1 & c_2 & c_3 & & & & & & c_j & & & & & & & & c_n \\
 & p_1 & 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & p_2 & 0 & 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & p_3 & 1 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
\text{probes} & & 0 & 0 & 1 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & & . & & & & & & & & & & & & & & & & \\
 & & . & & & & & & & & & & & & & & & & \\
 & p_i & 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 1 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & & . & & & & & & & & & & & & & & & & \\
 & & . & & & & & & & & & & & & & & & & \\
 & p_t & 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0
\end{array}
$$

All unions of up to *d* columns are distinct.

Decoding: *$O(n^d)$*

d-Disjunct Matrix

$$
\begin{array}{ll ccccccccccccccccc}
 & & \multicolumn{17}{c}{\text{viruses}} \\
 & & c_1 & c_2 & c_3 & & & & & & c_j & & & & & & & & c_n \\
 & p_1 & 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & p_2 & 0 & 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & p_3 & 1 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
\text{probes} & & 0 & 0 & 1 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & & \vdots & & & & & & & & & & & & & & & & \\
 & p_i & 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 1 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 & & \vdots & & & & & & & & & & & & & & & & \\
 & p_t & 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0
\end{array}
$$

Each column is different from the union of every d other columns

Decoding: O(n)

Remove all clones in negative pools. Remaining clones are all positive.

es

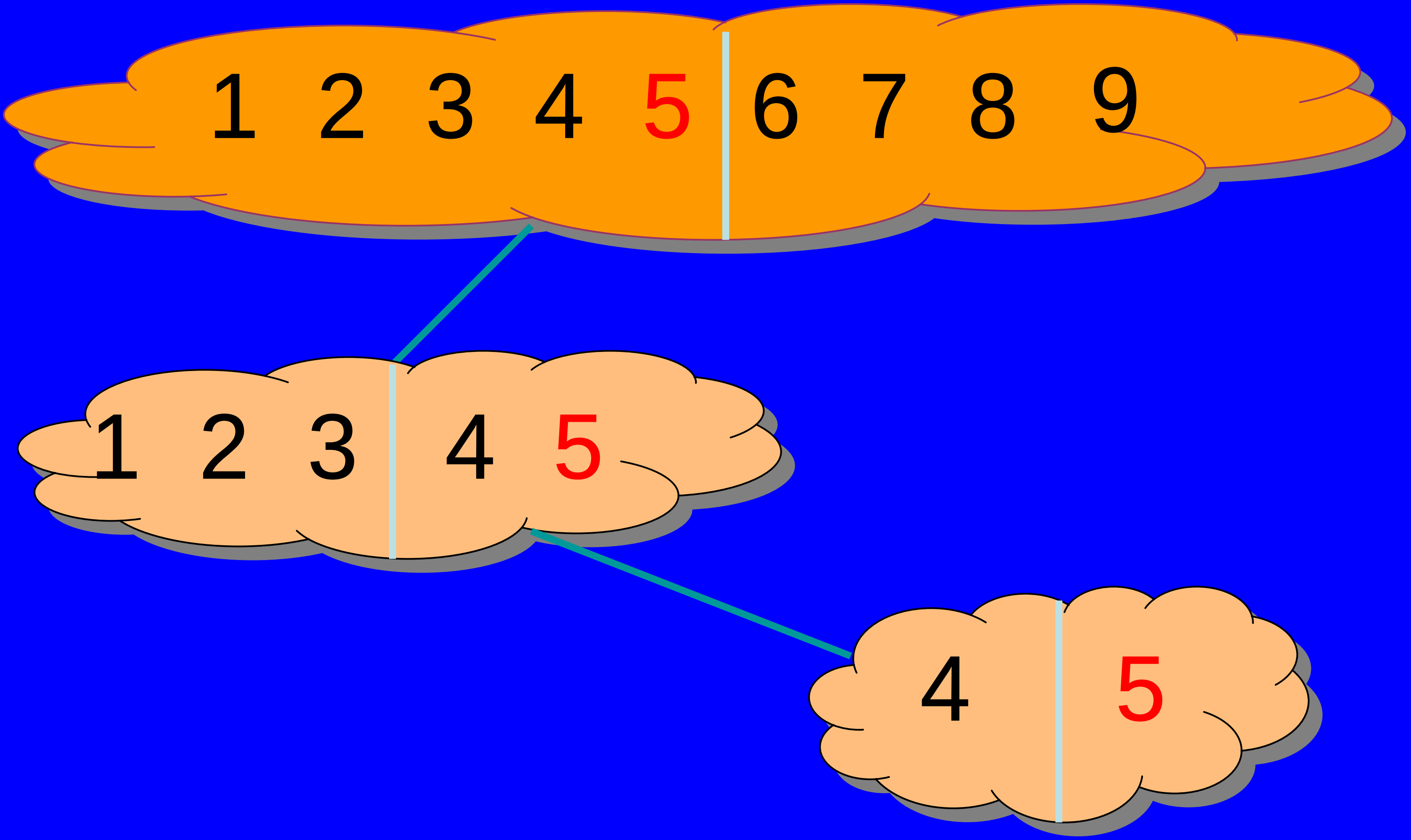
Nonunique Probe Selection

- Given a binary matrix, find a $\bar{d}$-separable submatrix with the same number of columns and the minimum number of rows.
- Given a binary matrix, find a d-disjunct submatrix with the same number of columns and the minimum number of rows.
- Given a binary matrix, find a d-separable submatrix with the same number of columns and the minimum number of rows

Classical Group Testing Model

- Given *n* items with some positive ones, identify all positive ones by less number of tests.
- Each test is on a subset of items.
- Test outcome is positive iff there is a positive item in the subset.

Example 1 - Sequential



Example 2 – Non-adaptive

p1	1	2	3
p2	4	5	6
p3	7	8	9
	p4	p5	p6

O($\sqrt{n}$) tests for *n* items

General Model about Nonadaptive Group Testing

- Classical: no restriction on pools.
- Complex model: some restriction on pools
- General model: Given a set of pools, select pools from this set to form a d-separable (d\bar-separable, d-disjunct) matrix.

Minimum d-Separable Submatrix

- Given a binary matrix, find a d-separable submatrix with minimum number of rows and the same number of columns.
- For any fixed d >0, the problem is NP-hard.
- In general, the problem is conjectured to be Σ_2^p–complete.

d-Separable Test

- Given a matrix M and d, is M d-separable?
- It is co-NP-complete.

$\bar{d}$-Separable Test

- Given a matrix M and d, is M d\bar-separable?
- This is co-NP-complete.

(a) It is in co-NP.

Guess two samples from space S(n,d\bar). Check if M gives the same test outcome on the two samples.

d-Disjunct Test

- Given a matrix M and d, is M d-disjunct?
- This is co-NP-complete.

Complexity of Sequential Group Testing

- Given n items, d and t, is there a group testing algorithm with at most t tests for n items with at most d positives?
- In PSPACE
- Conjectured to be PSPACE-complete.

Complexity of Nonadaptive Group Testing

- Given n items, d and t, is there a t x n d-separable matrix?
- Given n items, d and t, is there a t x n d\bar-separable matrix?
- Given n items, d and t, is there a t x n d-disjunct matrix?

Approximation

- Greedy approximation has performance

  1+2d ln n
- If NP not= P, then no approximation has performance o(ln n)
- If NP is not contained by DTIME(n^{log log n}), then no approximation has performance (1-a)ln n for any a > 0.

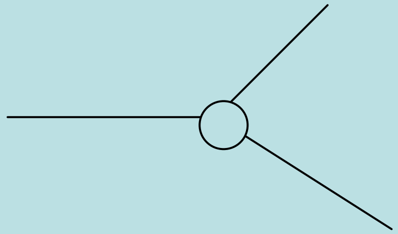
Pool Size = 2

- The minimum 1-separable submatrix problem is also called the minimum test set (the minimum test cover, the minimum test collection).
- The minimum test cover is APX-complete (story was complicated).
- The minimum 1-disjunct submatrix is really polynomial-time solvable.

Lemma

- Consider a collection *C* of pools of size at most 2. Let *G* be the graph with all items as vertices and all pools of size 2 as edges. Then
- *C* gives a *d*-disjunct matrix if and only if every item not in a singleton pool has degree at least *d+1* in *G*.

Proof



Suppose there exists an item a_0 not in any singleton pool of C and its degree in G is at most d. Let (a_0,a_1), (a_0, a_2), $\ldots$, (a_0,a_k) $(k < d)$ be all edges of G at a_0. Then column with label a_0 is contained in the union of columns with labels a_1, a_2, $\ldots$, a_k. Therefore, C does not form a d-disjunct matrix.

Conversely, if no such an item a_0 exists, then every item is either in a singleton pool or of degree at least $d+1$. In the former case, the singleton pool does not contain any other item, and in the latter case, for any d other items $a_1, a_2, \ldots, a_d$, there is a pool of size 2 contains a_0 and does not contain anyone of $a_1, \ldots, a_d$. Hence, C form a d-disjunct matrix.

Theorem

- Min-d-DS is polynomial-time solvable in the case that all given pools have size exactly 2

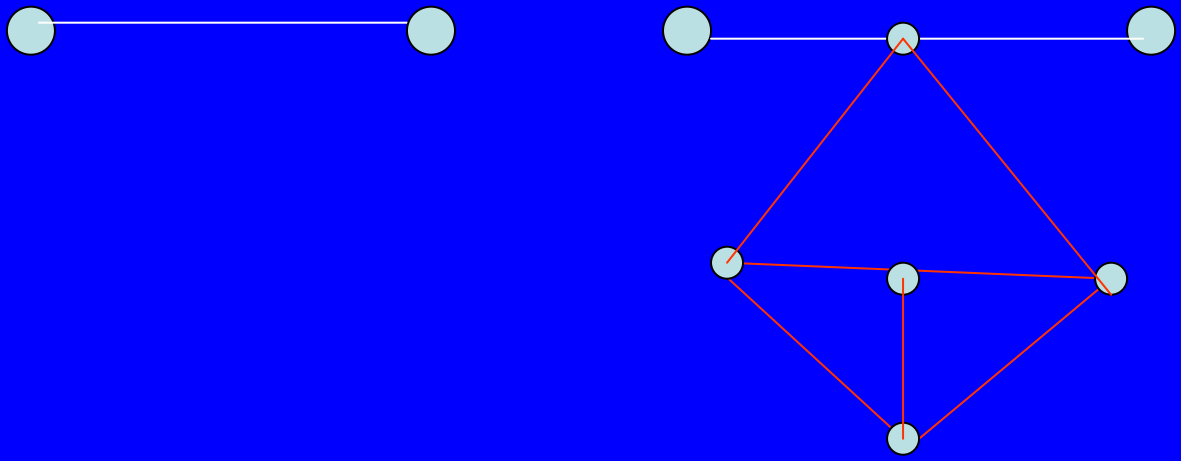
Let H be the graph with all items as vertices and all given pools as edges. By Lemma, Min-d-DS is equivalent to find a subgraph G, with minimum number of edges, such that every vertex has degree at least $d+1$ in G. It is equivalent to maximize the number of edges in H-G such that every vertex v has degree at most $d_{H(v)}$-d-1 in H-G where $d_{H(v)}$ is the degree of v in H. The letter maximization problem has been known to be polynomial-time solvable for a long time.

Theorem 2

- Min-2-DS is NP hard in the case that all given pools have size at most 2.

Proof

- Vertex-Cover



Theorem 2’

- Min-2-DS is MAX SNP-complete in the case that all given pools have size at most 2.

Lemma 2

- Suppose all given pools have size at most 2. Let s be the number of given singleton pools. Then any feasible solution of Min-d-DS contains at least s+ $(n$-$s)(d$+$1)/2$ pools.

Proof

Suppose C is a feasible solution of Min - d - DS. By Lemma 1, every item is either in a singleton pool or in at least $d+1$ pools of size 2. Suppose C contains s singleton pools. Then C contains at least $s+(n\text{-}s)(d+1)/2$ pools.

Step 1

- Compute a minimum solution of the following polynomial-time solvable problem: Let *G* be the graph with all items as vertices and all given pools of size 2 as edges. Find a subgraph *H*, with minimum number of edges, such that every item not in a singleton pool has degree at least *d+1*.

Step 2

- Suppose *H* is a minimum solution obtained in Step 1. Choose all singleton pools at vertices with degree less than *d+1* in *H*. All edges of *H* and chosen singleton pools form a feasible solution of Min-*d*-DS.

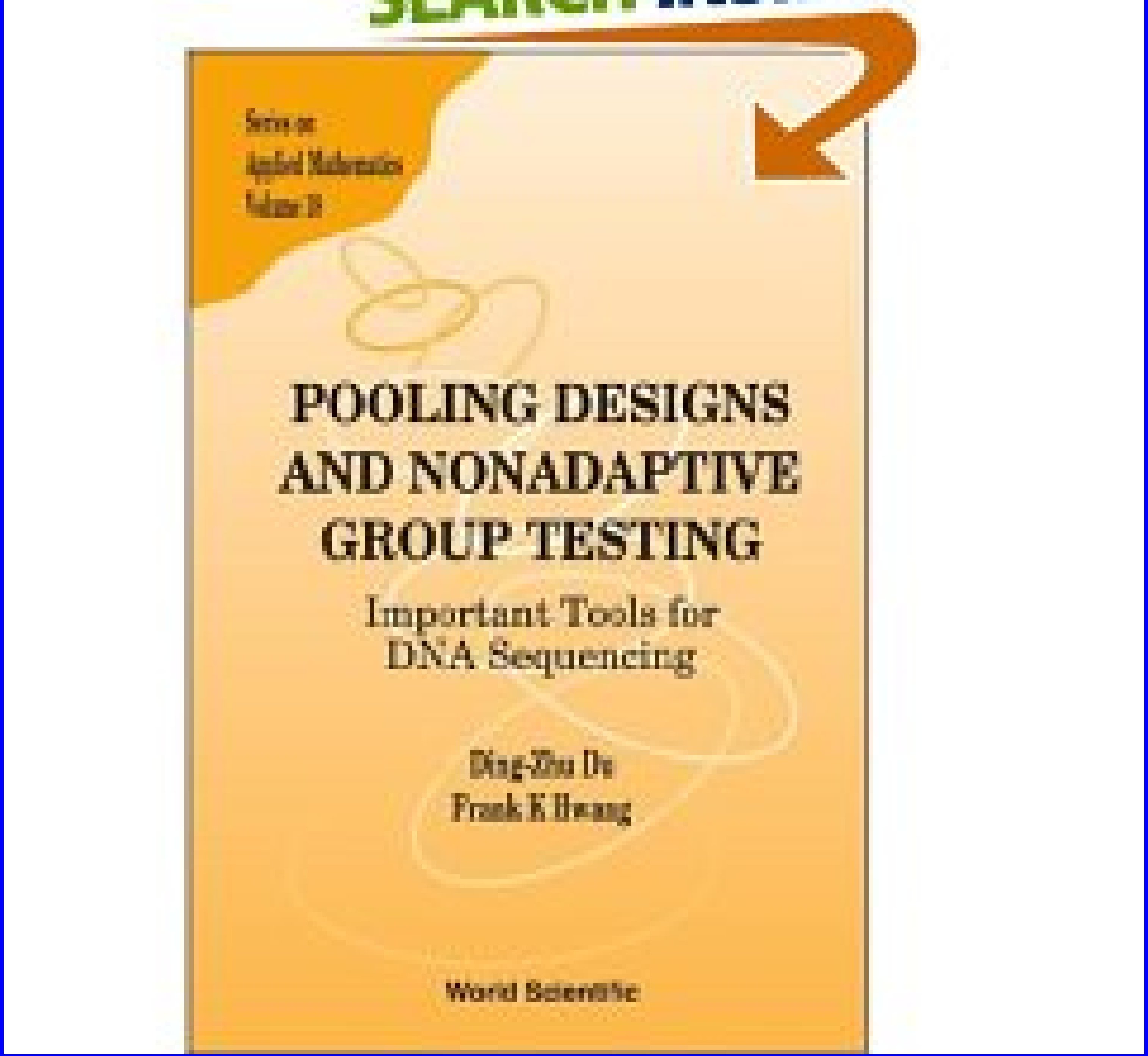
Theorem 3

- The feasible solution obtained in the above algorithm is a polynomial-time approximation with performance ratio $1+2/(d+1)$.

Proof

- Suppose *H* contains *m* edges and *k* vertices of degree at least $d+1$.
- Suppose an optimal solution containing s^* singletons and m^* pools of size 2.
- Then $m \leq m^*$ and $(n-k)-s^* \leq 2m^*/(d+1).$
- $(n-k)+m \leq s^*+m^*+ 2m^*/(d+1)$

$$< (s^*+m^*)(1+2/(d+1)).$$



Series on
Applied Mathematics
Volume 18

POOLING DESIGNS AND NONADAPTIVE GROUP TESTING

Important Tools for DNA Sequencing

Ding-Zhu Du
Frank K Hwang

World Scientific