
Learning a Hidden Graph

Example of Group Testing

- In screening clone library the goal is to determine which clones in the clone library hybridize with a given probe in an efficient fashion. A clone is said to be **positive** if it hybridize with the **probe(探針)**, and **negative** otherwise.
-

Various Models

- *Errors* occurred.
 - There are *inhibitors*.
 - There is a *threshold*.
 - Defective items are sets: *complex* model.
 - *Competitive* model: the number of defectives is unknown.
 - The defectives are *mutually obscuring*.
-

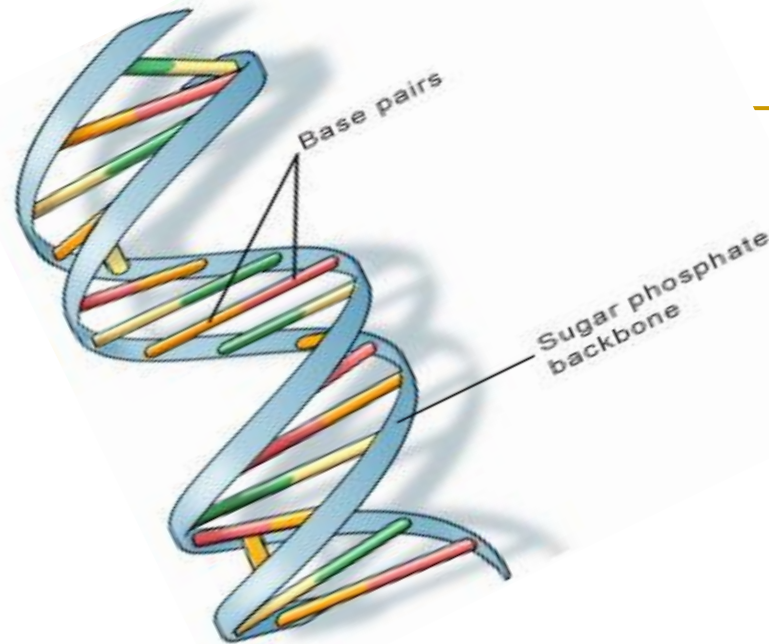
Competitive Group Testing

- In classical group testing, the number of positive substances is usually assumed *bounded* while for *competitive group testing*, *no information* on the number of positive substances is given.
- A *learning graph problem* can be viewed as a competitive group testing on complex model with each complex of *size two*.
- If we are learning *hypergraphs*, then the size of complex is getting larger.

Graph Learning Problem

- Let H be a family of labeled graphs (*hidden*) defined on the set $[n] = \{1, 2, \dots, n\}$.
- Referring to the information of “*belonging to H* ”, we wish to identify G by *edge-detecting queries*, each of which tells whether a subset (with some constraints) of $[n]$ induces an edge (or more info.) of G .
- **Who is hiding there?**

Motivated by applications in DNA physical mapping



Whose Idea?

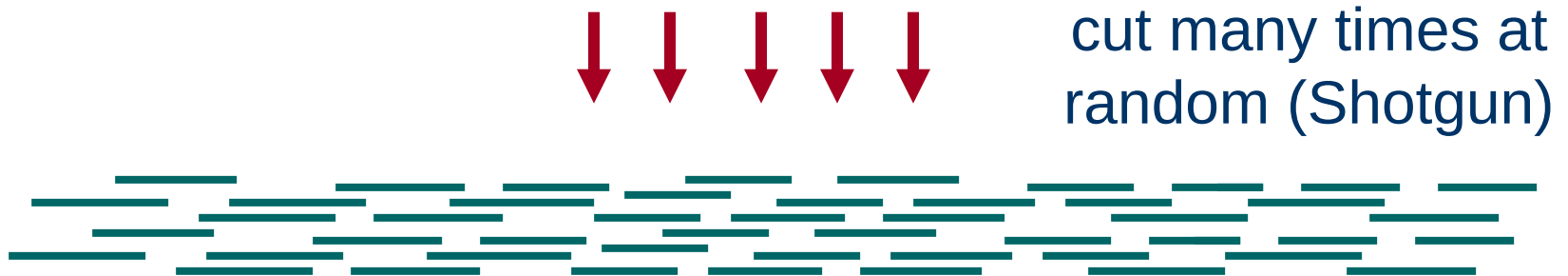
- In 1996, *A. Sorokin et al.* proposed the following idea in *Genome Research*: For physical mapping, some cloned fragments are produced to cover the whole DNA molecule with *some gaps* and then some of them are assembled to form longer continuous fragments (contigs). However, the information about the *mutual placement* of the contigs on the DNA sequence is lost.

Shotgun Sequencing

- Shotgun sequencing is a throughput technique resulting in the sequencing of a large number of bacterial genomes, mouse genomes and the celebrated human genomes. In all such projects, we are left with a collection of contigs that for special reasons cannot be assembled with general assembly algorithms. Continued ...
-

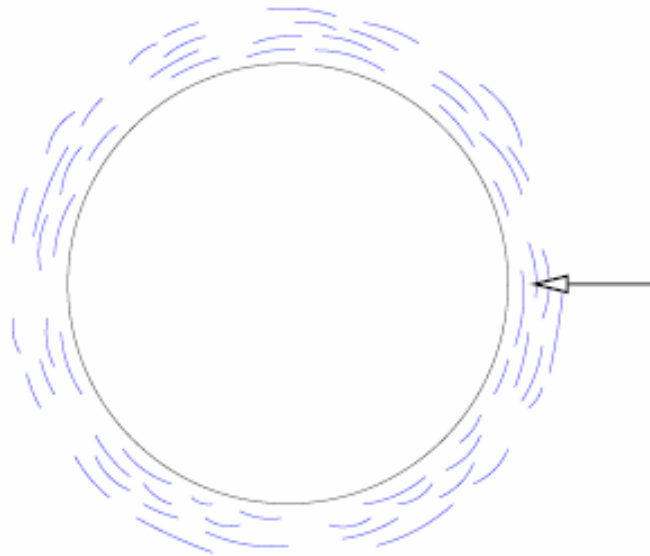
Random shotgun approach

genomic segment



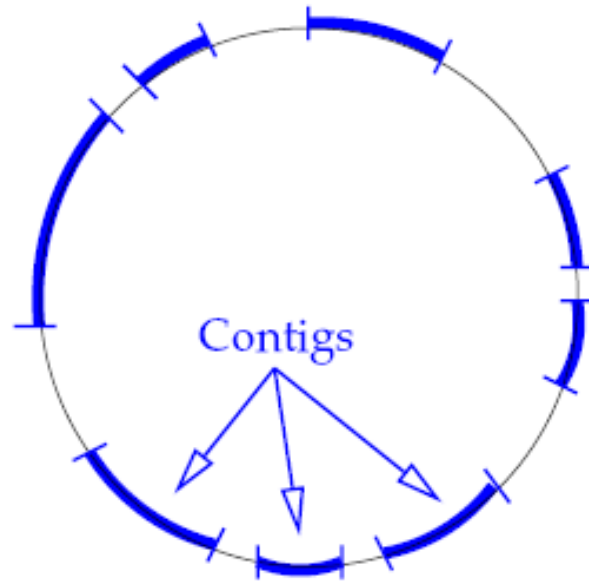
■ Whole-genome shotgun sequencing

- Short reads are obtained and covering the genome with redundancy and possible gaps.

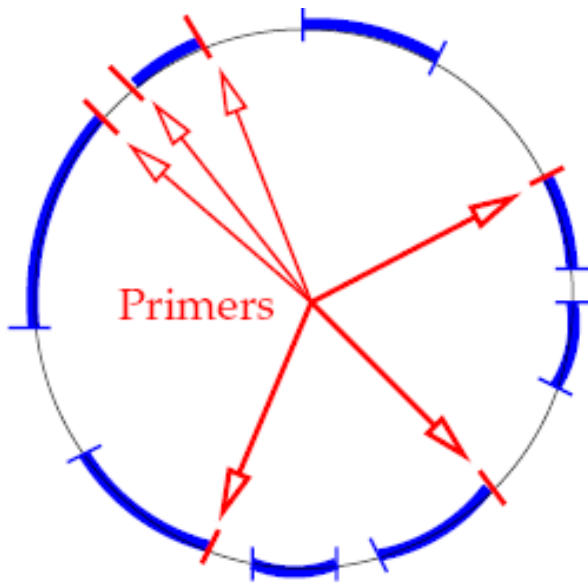


Circular genome

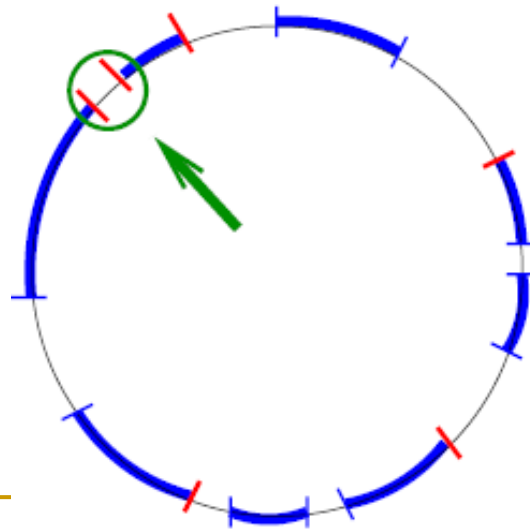
- ❑ Reads are assembled into contigs with unknown relative placement.

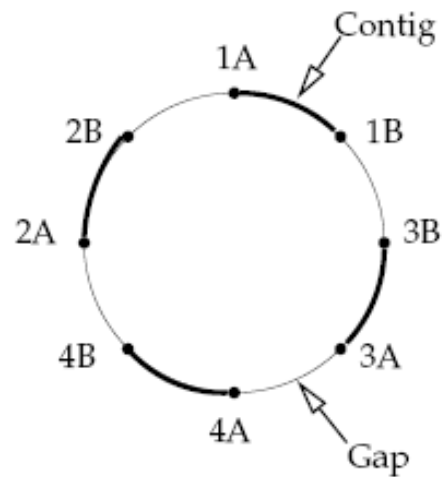


- **Primers** : (short) fragments of DNA characterizing ends of contigs.



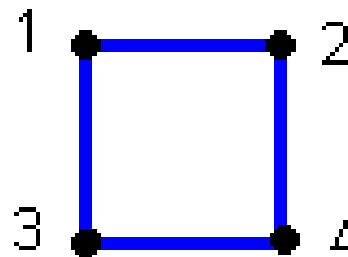
- ❑ A **PCR (Polymerase Chain Reaction)** reaction reveals if two primers are proximate (adjacent to the same gap).
- ❑ **Multiplex PCR** can treat multiple primers simultaneously and outputs if there is a pair of adjacent primers in the input set and even sometimes the number of such pairs.

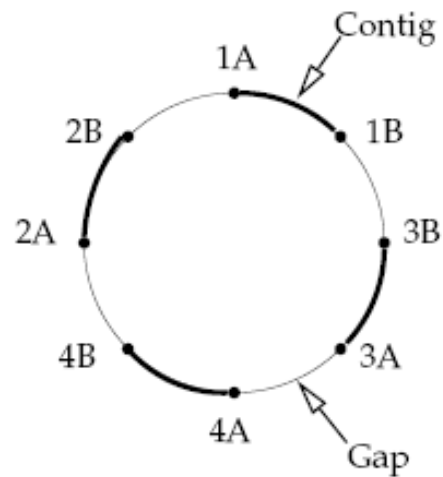




- Two primers of each contig are “mixed together”
 - Find a Hamiltonian cycle by PCRs!

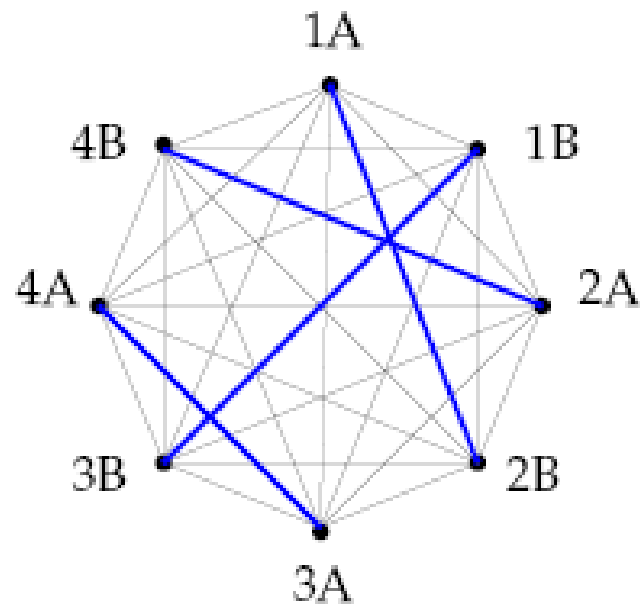
1A ——— 1B
 2A ——— 2B
 3A ——— 3B
 4A ——— 4B





1A ——— 1B
 2A ——— 2B
 3A ——— 3B
 4A ——— 4B

- Primers are treated independently.
- Find a perfect matching by PCRs.



Goal

- Our goal is to provide an experimental protocol that identifies all pairs of adjacent primers with as few PCRs (queries) (or multiplex PCRs respectively) as possible.

Mathematical Models

Hidden Graphs
(Reconstructed)

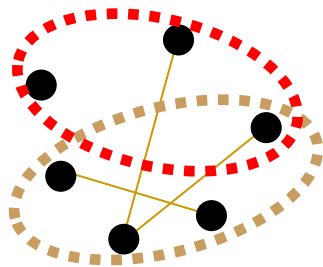


• Models

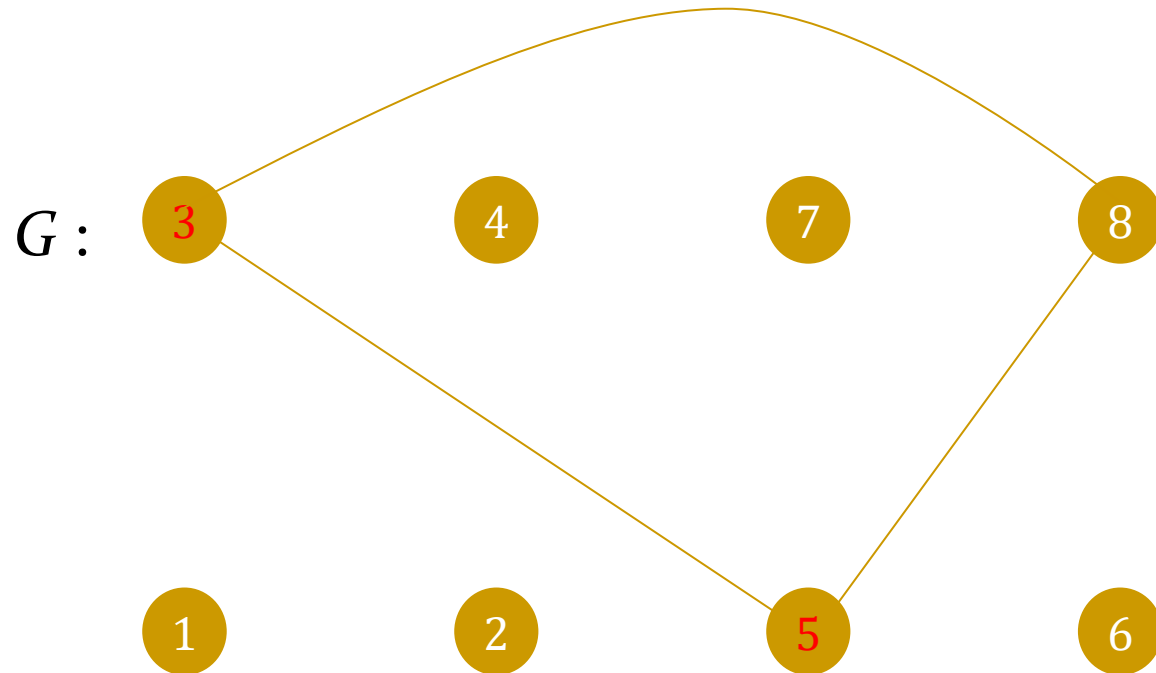
- 1) *Multi-vertex model*
- 2) *Quantitative multi-vertex model*
- 3) *k-vertex model*
- 4) *Quantitative k-multi-vertex model*

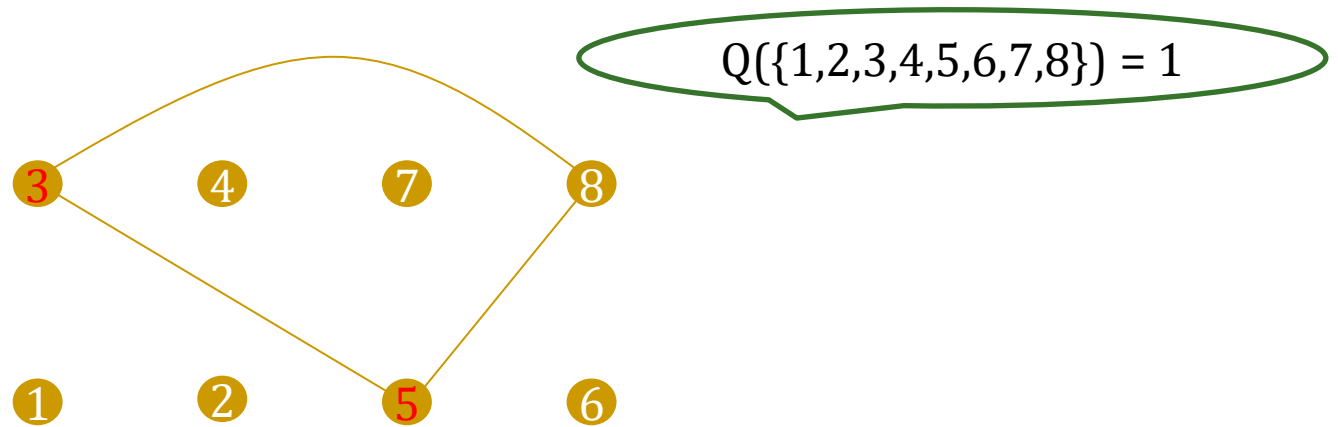
Learning a hidden graph by edge-detecting queries:

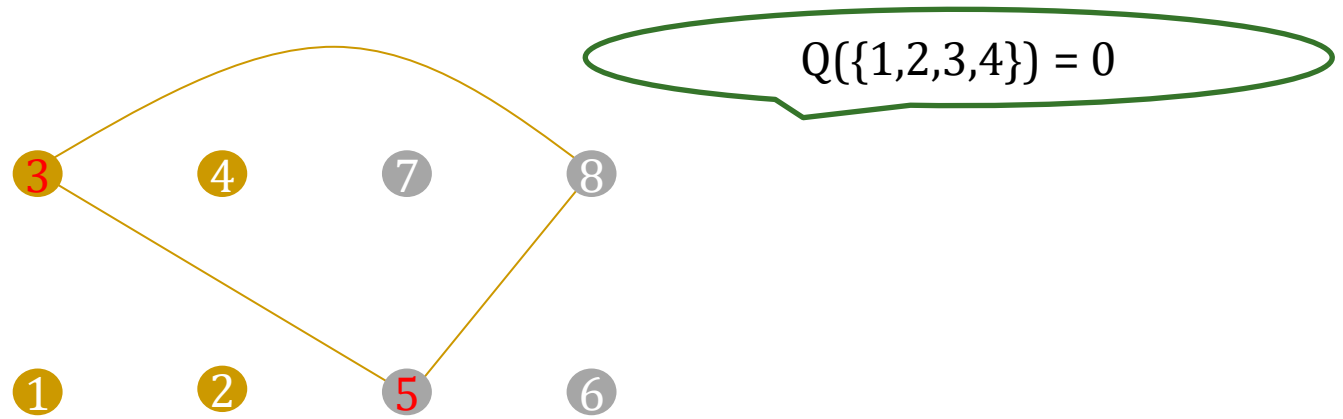
$Q_G(S)$: does S include at least one edge of G ?

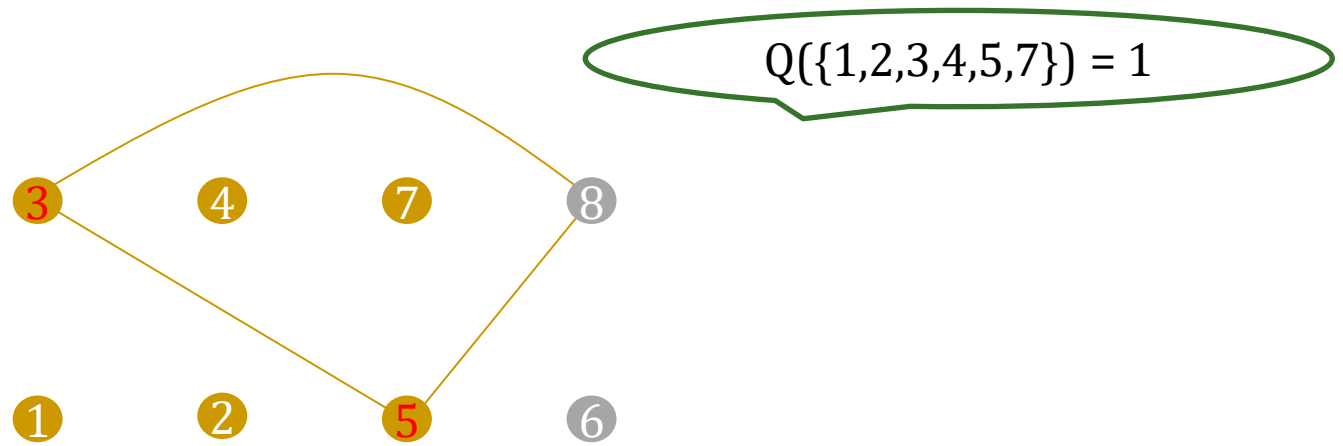


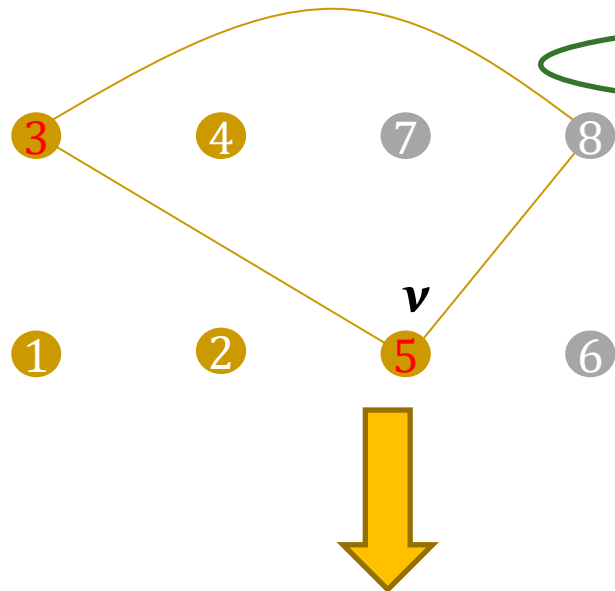
Example



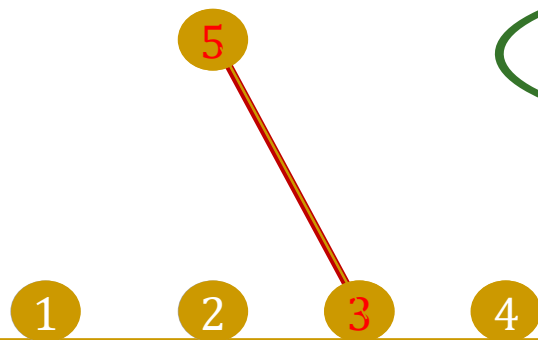








$v = \{5\}, S \setminus \{v\} = \{1, 2, 3, 4\}$



$Q(\{5,3\}) = 1$

Then, ...

- We can delete the edge and start it over again to find another edge until all the edges are found.
- Clearly, if the hidden graph we are looking for is of large size compare to n , then this algorithm is not going to be a good one.
- *We may simply ask all the $n(n-1)/2$ 2-subsets of $[n]$.*

Lower bound (Information theoretic)

- **Theorem.** For any $0 < \varepsilon < 2$, $\varepsilon m(\log n - 2)$ edge-detecting queries are required to identify a graph G drawn from the class of all graphs with n vertices and $m = n^{2-\varepsilon}$ edges.

Proof.

$$\binom{\binom{n}{2}}{m} \geq \left(\frac{\binom{n}{2}}{m} \right)^m$$

$$\log \left(\frac{\binom{n}{2}}{m} \right)^m = m \log \left(\frac{\binom{n}{2}}{m} \right) = m \log \left(\frac{\binom{n}{2}}{n^{2-\varepsilon}} \right) \geq m \log \left(\frac{n^\varepsilon}{4} \right) = \varepsilon m(\log n - 2)$$

Better Lower Bound!

- In reconstructing a graph with n vertices and m edges, all graphs of order n and size less than m must be considered as graphs to be reconstructed. (Using computation trees.)
- *Theorem.* $\log(\sum_{0 \leq i \leq m} (n(n-1)/2 \text{ chooses } i))$ edge-detecting queries are required to identify a graph drawn from the family of all graphs with n vertices and m edges.

Main Ideas of search

- We start with finding the maximal matching!
 - Then, **PARTITION_OF_VERTEX_SET (Identify-Bipartite)**
 - If there are edges between two independent sets A and B , we may find all of the edges by using (a, B) -algorithm, $a \in A$.
 - Finally, find **HIDDEN_GRAPH**.
-

5 Steps Algorithm (How?)



Using Designs (Affine Planes)

- *An affine plane of order q is a balanced incomplete block design, $2 - (q^2, q, 1)$ design, such that the set of blocks can be partitioned into parallel classes.*
- A Steiner triple system of order 9, STS(9), is an example of an affine plane of order 3.
- It is well-known that the existence of an affine plane of order q is equivalent to the existence of *a complete set of mutually orthogonal Latin squares of order q which exists when q is a prime power.*

Continued ...

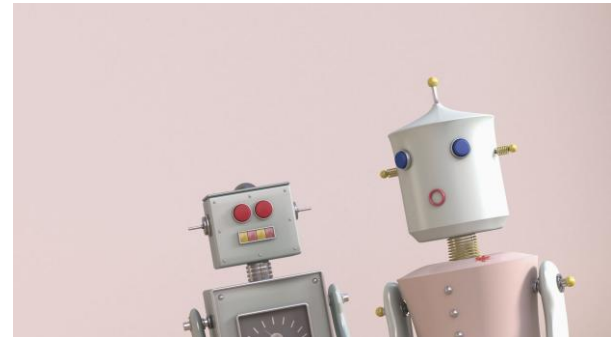
- **Theorem.** *For each positive integer n there exists a prime p power such that $2n \geq p^2 \geq n$.*
- There exists a pairwise balanced design (PBD), $2 - (n, K, 1)$ design, such that K contains positive integers not greater than p and the design has at most $p(p+1)$ blocks. (?)
- *All edges of a hidden graph defined on $[n]$ are hiding in the blocks!*

How many queries?

- The number of queries is at most $m(\log n + 3) + \alpha'(G)(\log n + 5) + 1$ where m is the number of edges in G and $\alpha'(G)$ is the matching number.
- By testing the $p(p+1)$ blocks of the PBD obtained above first and then reconstructing the edges in each block, we need at most $m(\log p + 3) + m(\log p + 5) + p^2 + p$ queries which is about $m(\log n) + 9m + 3n$ queries.

Another Set UP

If all the possible edges (complex model) are known to be in a bipartite graph, how do we find a positive edge? By the way, the queries we are asking is the same as previous model:
whether there is a positive edge in the graph induced by a set of vertices?



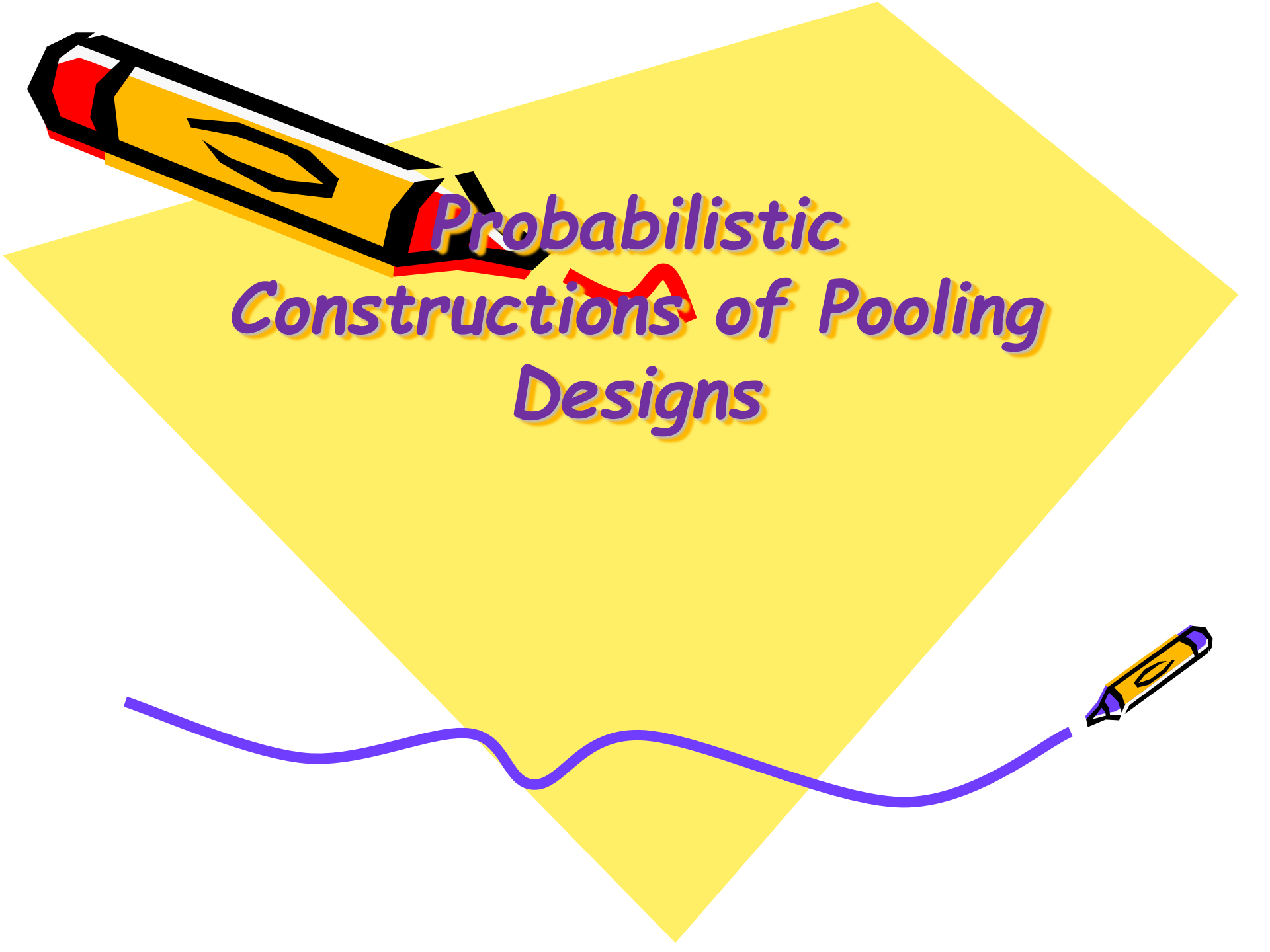
Bipartite Graph Conjecture

Conjecture

Given a bipartite graph G of size 2^k , there exists an induced subgraph of G whose size is 2^{k-1} .

(*) If this conjecture is true, then we have a binary splitting algorithm to find a positive edge with $\{\log_2 m\}$ tests where m is the size of G .

Bonus: Prove the conjecture! (萬元徵答)



Probabilistic Constructions of Pooling Designs

Set Notation (Design)

- Let $M = [m_{i,j}]$ be a $t \times n$ matrix mentioned above. Then we can use n sets (ordered) S_i 's to represent the matrix where

$$S_k = \{i : m_{i,k} = 1, i = 1, 2, \dots, t\}, k = 1, 2, \dots, n.$$

- The following sets represent the (0,1)-matrix of the last 9×12 matrix:

$\{1,2,3\}, \{4,5,6\}, \{7,8,9\}, \{1,4,7\}, \{2,5,8\}, \{3,6,9\},$
 $\{1,5,9\}, \{2,6,7\}, \{3,4,8\}, \{1,6,8\}, \{2,4,9\}, \{3,5,7\}.$

(Projective plane of order 3.)

d-separable and d-disjunct matrices

- A matrix is *d-separable* if $\cup D \neq \cup D'$ for any two distinct d -sets D and D' (columns), i.e. no two unions of d columns are the same.
- A matrix is *d-disjunct* if no column is contained in the union of any other d columns.

Important Facts

- A d -disjunct matrix is also a d -separable matrix.
- A d -separable matrix can be applied to find exactly d positives.

Proof. The union of d columns corresponds to distinct *outcome* vector.

- d -disjunct matrices have a simple decoding algorithm, namely, a column is positive if and only if it does not appear in a negative row.
(The above matrix is 2-disjunct.)

Disjunct Matrices

- Disjunct matrices can be considered as special designs or special codes.
- **Extra constraints** are needed in order to handle various models in group testing.
- We are aiming at finding all the defectives with comparatively **easier decoding algorithms**.
- Of course, also **minimize** the number of tests.

$(d; z)$ -disjunct

- Let A be a $t \times n$ $(0, 1)$ -matrix in which the rows are indexed by $[1, t]$ and the columns are indexed with the supports of columns. (Columns are subsets of $[1, t]$.)
- A is $(d; z)$ -disjunct if for any $d+1$ columns $C_0, C_1, \dots, C_d$, $|C_0 / (C_1 \cup C_2 \cup \dots \cup C_d)| \geq z$.
- If $z \geq 2e + 1$, then this matrix can be applied to find d defectives with e errors in outcome vectors.

Complex Model

- An extension of classical group testing problem.
- A set of *complexes*, each of which is a subset of basic items, is given and the property (*positive or negative*) of each complex is waiting to be determined.
- The *corresponding group test* determines whether a subset contains at least one positive complex (*instead of just one positive item*).

$(d, r]$ -disjunct

- A is $(d, r]$ -disjunct if the **union** of any d columns does not contain the **intersection** of any other r columns.
- If the defectives are r -subsets instead of items (**complex model**), this matrix can be utilized to find d defective complexes.
- In case of errors occurred, we define a
- **$(d, r; z]$ -disjunct** matrix accordingly.

(d, h)-disjunct

- A is called (d, h) -disjunct if the union of any d columns does not contain the union of the other h columns.
- A can be applied to find d defectives in an inhibitor model with h inhibitors.
- We need extra effort to handle this model with errors!

$(d, s \text{ out of } r]\text{-disjunct}$

- A is $(d, s \text{ out of } r]\text{-disjunct}$ if for any d columns and any other r columns, there exists a row (index) in which none of the d columns appear and at least s of r columns do.
- If $s = 1$, then A is $(d, r)\text{-disjunct}$ and if $s = r$, then A is $(d, r]\text{-disjunct}$.

(k, m, n) -selector

- For $m \leq k$, A is a (k, m, n) -selector if any $t \times k$ submatrix of A contains at least m rows of the identity matrix I_k .
- A (k, m, n) -selector corresponds to well-known combinatorial objects such as superimposed code and k -selectors.
- Superimposed codes and k -selectors are very basic combinatorial structures and find application in a variety of areas such as cryptography, data security, computational molecular biology etc.

Construction of disjunct-matrices

- There are known constructions, deterministic and probabilistic.
- Here, we shall use the well-known **Lovász Local Lemma** to construct such matrices.
- We shall present one of them and the others are similar in some sense.



Lovász Local Lemma

■ (Symmetric Case)

Let $A_1, A_2, \dots, A_m$ be events in an arbitrary probability space. Suppose that each event A_i is mutually independent of a set of other events A_j but at most μ of them, and that $\Pr(A_i)$ is at most $0 < p < 1$ for all $1 \leq i \leq m$.

If $e \cdot p \cdot (\mu + 1) \leq 1$, then the product of m probabilities $\Pr(B_i)$ is larger than 0 where B_i is the complement of the event A_i .

(d, r]-disjunct Case

- Let $t(n, d, r]$ denote the **minimum number of rows for a $(d, r]$ -disjunct matrix with n columns.**
- For convenience of “typing”, we shall use $((n, d))$ to denote the combination number: n chooses d , i. e. $n!/d! \cdot (n-d)!$, and $(([n], k))$ to denote all k -subsets of $\{1, 2, \dots, n\}$.
- Notice that **$-\ln(1-x) \geq x$ for $0 \leq x < 1$.**

Upper Bound

■ $t(n, d, r] \leq (1 + d/r)^r (1 + r/d)^d \cdot$
 $\{1 + \ln[((n, d))((n-d, r)) - ((n-d-r, d)) \cdot$
 $((n-2d-r, r))]\}.$

Proof. (Sketch)

Let M be a $t \times n$ $(0, 1)$ -matrix with

$\Pr(m_{i,j} = 1) = p$ and $\Pr(m_{i,j} = 0) = 1 - p$. Note that all entries are chosen independently.

- Let $D \in ([n], d)$, $R \in ([n], r)$,
 $D \cap R = \emptyset$ and $A_{D,R}$ be the event such that
 $\cap R \subseteq \cup D$. Hence, the complement of $A_{D,R}$ says that $\cap R$ is not
contained in $\cup D$.
- $\Pr(A_{D,R}) = (1 - p^r(1-p)^d)^t$.
- $(\mu + 1) = ([n], d) \cdot ([n-d], r) - ([n-d-r], d) \cdot$
 $([n-2d-r], r)$.
- Hence, $e \cdot (1 - p^r(1-p)^d)^t \cdot [([n], d) \cdot ([n-d], r) - ([n-d-r], d) \cdot$
 $([n-2d-r], r)] \leq 1$ is required to make sure that the
intersection of the complements of all events $A_{D,R}$ does
have nonnegative probability and hence we have a $(d, r]$ -
disjunct matrix.

- Solving the inequality, we have

$$t \geq (1 + \ln[(\binom{n}{d} \cdot \binom{n-d}{r} - \binom{n-d-r}{d}) \cdot \binom{n-2d-r}{r}]) / (-\ln((1 - p^r(1-p)^d))). \dots (*)$$

By the fact $-\ln(1-x) \geq x$ for $0 \leq x < 1$, if

$$t \geq (1 + \ln[(\binom{n}{d} \cdot \binom{n-d}{r} - \binom{n-d-r}{d}) \cdot \binom{n-2d-r}{r}]) / p^r(1-p)^d, \text{ then } (*) \text{ holds. } \dots (**)$$

- Let $p = r/(d+r)$. Then the RHS of (**) has the minimum value. Hence, as long as we have this number of tests, a $(d, r]$ -disjunct matrix does exist. This implies that $t(n, d, r] \leq \text{RHS of } (**)$.

References

- A new construction of 3-bar-*separable* matrices via improved decoding of Macula construction (with F. K. Hwang), **Discrete Optim.**, 5(2008), 700-704.
- An upper bound of the number of tests in pooling designs for the *error-tolerant complex model* (with Hong-Bin. Chen and F. K. Hwang), **Optimization Letters**, 2(2008), no. 3, 425-431.
- The minimum number of e-vertex-cover among *hypergraphs* with e edges of given rank (with F. H. Chang, F. K. Hwang and B. C. Lin), **Discrete Applied Math.**, 157(2009), 164-169.
- Non-adaptive algorithms for *threshold* group testing (with Hong-Bin Chen), **Discrete Applied Math.**, 157(2009), 1581-1585.

Continued

- Identification and classification problem on pooling designs for *inhibitor* models (with Huilan Chang and Hong-Bin Chen), **J. Computational Biology**, 17(2010), No. 7, 927-941.
- Reconstruction of *hidden graphs and threshold* group testing (with Huilan Chang, Hong-Bin Chen and Chih-Huai Shih), **J. Combin. Optimization**, 22(2011), no. 2, 270 – 281.
- Group testing with multiple *mutually-obscuring positives* (with Hong-Bin Chen), **Lecture Notes Computer Science**, 7777(2013), 557 – 568.
- *Threshold* group testing on *inhibitor* model (Huilan Chang and Chih-Huai Shih), **J. Computational Biology**, Vol. 20, No. 6, 2013, 1 – 7.

Continued

- New bound on *2-bar-separable* codes of length 2 (with Minquan Cheng, J. Jiang, Yuan-Hsun Lo and Ying Miao), **Des. Code Cryptography**, to appear.
- Learning a *hidden graph* (with Huilan Chang and Chie-Huai Shih), under review of Optimization Letters.
- Kuo-An Yu, Applications of the *Lovász Local Lemma* to pooling designs, M.S. thesis, 2007, National Chiao Tung University.