Molecular Biology and Pooling Design

Weili Wu[1], Yingshu Li[2], Chih-hao Huang[2], and Ding-Zhu Du[1]

[1] Department of Computer Science,
University of Texas at Dallas,
Richardson, TX 75083, USA
{weiliwu,dzdu}@utdallas.edu
[2] Department of Computer Science and Engineering,
University of Minnesota,
Minneapolis, MN 55455, USA
{yili,huang}@cs.umn.edu

Summary. The study of gene functions requires a high-quality DNA library. A large amount of testing and screening needs to be performed to obtain a high-quality DNA library. Therefore, the efficiency of testing and screening becomes very important. Pooling design is a very helpful tool, which has developed a lot of applications in molecular biology. In this chapter, we introduce recent developments in this research direction.

1 Molecular Biology and Group Testing

One of the recent important developments in biology is the success of Human Genome Project. This project was done with a great deal of help from computer technology, which made molecular biology a hot research area conjugated with computer science. Bio-informatics is a new born research area that grows very rapidly from this conjugation.

The technology for obtaining sequenced genome data is getting more developed as and more and more sequenced genome data is available to the scientific research community. Based on those data, the study of gene functions has become a very important research direction. This requires high-quality gene libraries. The high-quality gene libraries are obtained from extensive testing and screening of DNA clones, that is, identifying clones used in the libraries. Therefore, the efficiency of DNA screening is very important. For example, in 1998, the Life Science Division of Los Alamos National Laboratories [14] was dealing with a dataset of 220,000 clones. Individual testing those clones requires 220,000 tests. However, they used only 376 tests with a technology called *group testing*.

The group testing takes advantage of small percentage of clones containing target probes. It tests subsets of clones called *pools*, instead of testing each

of them individually. For example, in the above mentioned testing at Los Alamos National Laboratories, each pool contained about 5,000 clones. The technology of group testing was started from Wassernan-type blood test in World War II. A very simple design that was used in the earlier stage is as follows: Divide each blood sample into two parts. First, mix all first parts into a pool and test the pool. If the outcome is positive, i.e., there is a presence of syphilitic antigen, then test the second part individually. Otherwise, all men in the pool passed the test. During the past 60 years, more efficient designs have been developed. These designs have gained more and more attention due to significant applications in the study of genome.

A typical application of pooling designs is DNA library screening. A DNA library is a collection of cloned DNA segments usually taken from a specific organism. Those cloned DNA segments are called *clones*. Given a DNA library, the problem is to identify whether each clone contains a probe from a given set of probes. A *probe* is a piece of DNA labeled with radioisotope or fluorescence. The probe is often used to detect specific DNA sequences by hybridization. A clone is said to be *positive* if it contains a given probe and *negative* otherwise. A pool is *positive* if it contains a positive clone and *negative* otherwise. In a group testing algorithm a clone may appear in two or more pools. Therefore, making copies is a necessary preprocessing procedure.

Hybridization is one of the techniques to reproduce clones or perform *DNA cloning*. To better understand the concept of hybridization, let us describe the composition of DNA. DNA is a large molecule with double helix structure that consists of two nucleic acids which in turn are strings of nucleotides. There are four types of nucleotides A (adenine), T (thymine), G (guanine) and C (cytosine). Thus, each nucleic acid can be seen as a string of four symbols A, T, G, C. When two nucleic acids are joined into a double helix, A must bond with T and G must bond with C. Heating can break the DNA into two separated nucleic acids. Through the action of an enzyme each nucleic acid may be jointed with a probe and consequently the probe would grow into a dual nucleic acid. This process is referred to as *hybridization*.

By repeating hybridization we can clone unlimited number of copies of any piece of DNA. This approach is called Polymerase Chain Reaction (PCR). It is a cell-free, fast, and inexpensive technique. Another technique for DNA cloning is cell-based. It contains four steps:

(1) Insert the DNA fragment (to be cloned) into an agent called *vector*. This step results in a recombinant.

(2) Put the recombinant DNA into a host cell to proliferate. This step is called *transformation*.

(3) Reproduce the transformed cell.

(4) Isolate the desired DNA clones from the cells obtained from (3).

In general, there are two conditions that need to be satisfied for group testing: (a) copies of items are available and (b) testing on a subset of items is

available. In DNA library screening both conditions are available due to DNA cloning, especially hybridization.

2 Pooling Design

There are two types of group testing, sequential and non-adaptive. To explain them, let us look at two examples of group testing algorithms.

Consider a set of nine clones with one positive clone. In the first example, the method is sequential. At each iteration, bisect the positive pool into two equal or almost equal pools and test each of the obtained two pools until only one positive clone is found in the pool. In the worst case, this method takes at most six tests to identify the positive clone. In general, for a set of n clones with one positive clone, the bisection would take at most $2\lceil \log_2 n \rceil$ tests to identify the positive one.

In the second example, the method is to put the nine clones into a 3×3 matrix. Each row and each column represent a test. Since there is only one positive clone, there is exactly one positive row and one positive column. Their intersection is the positive clone. In general, for n clones that include a positive one, this method takes $O(\sqrt{n})$ tests. For large n, this method needs more tests than the first one. However, all tests in this method are independent. They can be performed simultaneously. This type of group testing is called *non-adaptive* group testing.

Group testing in molecular biology is usually called *pooling design*. The pooling design is often non-adaptive [3, 8]. This is due to the time consuming nature of tests in molecular biology. Therefore, we may simply refer to the pooling design as the non-adaptive group testing. Hence, every pooling design can be represented as a binary matrix by indexing rows with pools and columns with clones and assigning 1 to cell (i, j) if and only if the ith pool contains the jth clone.

A positive clone would imply the positivity of all pools containing it. Therefore, d positive clones would result in the positivity of all pools containing any of them. If we consider each column (clone) as a set of pools with 1-entry in the column, then the union of the d columns represents the testing outcome when those d clones form the set of all positive clones. Therefore, if a binary matrix representing a pooling design can identify up to d positive clones, all unions of up to d columns should be distinct. A binary matrix with this property is called $\bar{d}$*-separable*.

For a $\bar{d}$-separable matrix, a naive way for decoding a given testing outcome vector to find all positive clones is to compare it with all unions of up to d columns. This takes $O(n^d)$ time. Is it possible to do better? The following result of Li mentioned in [18] gives a negative answer.

Theorem 1 *Decoding for $\bar{d}$-separable matrix can be done in polynomial time with respect to n and d if and only if the hitting set problem is polynomial-time solvable, i.e., if and only if P=NP.*

Indeed, decoding is equivalent to finding a subset of at most d clones hitting every positive pool. By a set hitting another set, we mean that the intersection of two sets is nonempty. Note that every clone in a negative pool is negative. Therefore, the input size of this hitting problem is controlled by the union of negative pools. The following result gives an interesting condition on the size of this union.

Theorem 2 *For a $\bar{d}$-separable matrix, the union of negative pools always contains at least $n-d-k+1$ clones if and only if no d-union contains a k-union, where a d-union means a union of d columns.*

When $k = 1$, the union of negative pools contains at least $n - d$ clones. Thus, the number of clones that are not in any negative pool is at most d, and hence they form a hitting set of at most d clones, which should be the solution. The binary matrix with the property that no column is contained in any d-union is said to be *d-disjunct*. For any d-disjunct matrix, decoding can be done in $O(n)$ time.

3 Simplicial Complex and Graph Properties

Finding the best d-disjunct matrix is an intractable problem for computer science. So far, its computational complexity is unknown. Therefore, we can only make approximate designs with various tools, including classical combinatorial designs, finite geometry, finite fields, etc. Recently, the construction of pooling designs using simplicial complexes was developed. A simplicial complex is an important concept in geometric topology [15, 18].

A *simplicial complex* Δ is a family of subsets of a finite set E such that $A \in \Delta$ and $B \subset A$ imply $B \in \Delta$. Every element in E is called a *vertex*. Every member in the family Δ is called a *face* and furthermore called a *k-face* if it contains exactly k vertices. Motivated by the work of Macula [12, 13], Park *et al.* [15] construct a binary matrix $M(\Delta, d, k)$ for a simplicial complex Δ by indexing rows with all d-faces and columns with all k-faces ($k > d$) and assigning 1 to cell (i, j) if and only if the ith d-face is contained in the jth k-face. They proved the following theorem.

Theorem 3 *$M(\Delta, d, k)$ is d-disjunct.*

An important family of simplicial complexes is induced by monotone graph properties. A graph property is *monotone increasing* if every graph containing a subgraph having this property also has this property. Similarly, a graph property is *monotone decreasing* if every subgraph of a graph with this property has this property. If one fixes a vertex set and considers edge sets of all graphs satisfying a monotone decreasing property, they will form a simplicial complex. Since graphs not satisfying a monotone increasing property form a monotone decreasing property, every monotone increasing property is also associated with a simplicial complex.

Matching is an example of a monotone decreasing property. Let Δ_m be the the simplicial complex consisting of all matchings in a complete graph of order m. Then k-matching (a matching of k edges) is a k-face of Δ_m. There is an error tolerance result for matching [7].

Theorem 4 *If k-matching is perfect, then $M(\Delta_m, d, k)$ is a d-error detecting d-disjunct matrix.*

Here, by a *d-error detecting* matrix, we mean that if there exist at most d erroneous tests, the matrix is still able to identify all positive clones.

Park *et al.* [15] also generalized this result to the case of a simplicial complex.

Theorem 5 *If for any two k-faces A and B $|A \setminus B| \geq 2$, then $M(\Delta, d, k)$ is a d-error detecting d-disjunct matrix.*

Huang and Weng [10] generalized Theorem 3 to a class of partial ordering sets, including lattices.

4 Error-Tolerant Decoding

Error-tolerant decoding is a very interesting issue in various pooling design models. To see it, let us study a so-called inhibitor model.

In fact, in some situations, a clone can be negative, positive or anti-positive. An *anti-positive* clone can cancel the positivity of a pool, that is, a test outcome on a pool containing an anti-positive clone must be negative, even if the pool contains a positive clone. An anti-positive clone is also called an *inhibitor*. If we know a positive clone, then all inhibitors can be identified by testing all pairs of clones consisting of the known positive clone and all clones in negative pools. However, if no positive clone is known, it is not so easy to identify inhibitors. Therefore, it is an interesting problem to decode all positive clones without knowing inhibitors.

Du and Hwang [4] developed the following method.

For each clone j and a possible subset I of inhibitors, compute $t(j, I)$, the number of negative pools containing j, but disjoint from I. Set $T(j) = \min t(j, I)$ over all possible subsets I.

They proved the following theorem.

Theorem 6 *For a $(d+r+e)$-disjunct matrix, if the input sample contains at most r inhibitors and at most d positive clones, and testing contains at most e erroneous tests, then $T(j) < T(j')$ for any positive clone j and any negative clone j'.*

Consequently, the following results can be formulated.

Theorem 7 (Du and Hwang [4]) *A $(d+r+e)$-disjunct matrix can identify all positive clones for every sample with d positive clones and at most r inhibitors subject to at most e erroneous tests.*

Theorem 8 (Hwang and Liu [9]) *A $(d+r+2e)$-disjunct matrix can identify all positive clones for every sample with at most d positive clones and at most r inhibitors subject to at most e erroneous tests.*

The inhibitor model was proposed by Farach *et al.* [6]. De Bonis and Vaccaro [1] developed a sequential algorithm for this model and raised an open problem of finding non-adaptive algorithm in this model. While D'yachkov *et al.* [5] solved the error-free case, Hwang and Liu [9] gave a general solution.

5 Future Research

The development of error-tolerant pooling designs is very important in practice. Theorems 3 and 4 established connections between error-tolerant designs and simplicial complexes. Since all monotone graph properties induce simplicial complexes, these connections may open a new research direction joint with graph theory to develop efficient designs.

There are many issues that we need to consider when constructing a pooling design. For example, after receiving test outcomes on all pools, the question to be addressed is how to decode this data to obtain information on each clone. The different designs have different computational complexity for decoding. One can find some interesting contributions and open problems in this area in [17].

In practice, DNA screening is closely related to information retrieval and data mining. In fact, database systems have already employed the technique of group testing. This opens an opportunity to attack some problems in data processing by applying our new designs. Therefore, our research work can be widely extended into different areas of computer science.

References

1. A. De Bonis and U. Vaccari. Improved algorithms for group testing with inhibitors. *Information Processing Letters*, 65: 57-64, 1998.
2. D.-Z. Du and F. K. Hwang. *Combinatorial Group Testing and Its Applications (2nd ed.)*, World Scientific, Singapore, 1999.
3. D.-Z. Du and F. K. Hwang. Pooling Designs: Group Testing in Biology, manuscript.
4. D.-Z. Du and F. K. Hwang. Identifying d positive clones in the presence of inhibitors, manuscript.
5. A. G. D'ychkov, A. J. Macula, D. C. Torney, and P. A. Vilenkin. Two models of nonadaptive group testing for designing screening experiments. In *Proceedings of the 6th International Workshop on Model-Oriented Designs and Analysis*, pages 63-75, 2001.
6. M. Farach, S. Kannan, E. Knill, and S. Muthukrishnan. Group testing problem with sequences in experimental molecular biology. In *Proceedings of the Compression and Complexity of Sequences*, pages 357-367, 1997.
7. H. Q. Ngo and D.-Z. Du. New constructions of non-adaptive and error-tolerance pooling designs. *Discrete Mathematics*, 243: 161-170, 2002.
8. H. Q. Ngo and D.-Z. Du. A survey on combinatorial group testing algorithms with applications to DNA library screening. In D.-Z. Du, P.M. Pardalos, and J. Wang, editors, *Discrete Mathematical Problems with Medical Applications*, pages 171–182. American Mathematical Society, Providence, RI, 2000.
9. F. K. Hwang and Y. C. Liu. Error tolerant pooling designs with inhibitors. *Journal of Computational Biology*, 10: 231-236, 2003.
10. T. Huang and C.-W. Weng. A note on decoding of superimposed codes. *Journal of Combinatorial Optimization*, 7: 381-384, 2003.
11. F.K. Hwang. On Macula's error-correcting pooling design, to appear in *Discrete Mathematics*, 268: 311–314, 2003.
12. A.J. Macula. A simple construction of d-disjunct matrices with certain constant weights. *Discrete Mathematics* 162: 311–312, 1996.
13. A. J. Macula. Error correcting nonadaptive group testing with d^e-disjunct matrices. *Discrete Applied Mathematics*, 80: 217–222, 1997.
14. M. V. Marathe, A. G. Percus, and D. C. Torney. Combinatorial optimization in biology, manuscript, 2000.
15. H. Park, W. Wu, Z. Liu, X. Wu, and H. Zhao, DNA screening, pooling designs, and simplicial complex. *Journal of Combinatorial Optimization*, 7(4): 389–394, 2003.
16. W. W. Paterson. *Error Correcting Codes*, MIT Press, Cambridge, MA, 1961.
17. W. Wu, C. Li, X. Wu, and X. Huang. Decoding in pooling designs. *Journal of Combinatorial Optimization*, 7(4): 385–388, 2003.
18. W. Wu, C. Li, and X. Huang. On error-tolerant DNA screening, submitted to *Discrete Applied Mathematics*.