Pacific Symposium on Biocomputing 6:348-359 (2001)

QUALITY CONTROL IN MANUFACTURING OLIGO ARRAYS: A COMBINATORIAL DESIGN APPROACH

RIMLI SENGUPTA and MARTIN TOMPA
Department of Computer Science and Engineering
University of Washington, Box 352350
Seattle, WA 98195-2350
{rimli,tompa}@cs.washington.edu

Abstract

The advent of the DNA microarray technology has brought with it the exciting possibility of simultaneously observing the expression levels of all genes in an organism. One such microarray technology, called "oligo arrays", manufactures short single strands of DNA (called *probes*) onto a glass surface using photolithography. An altered or missed step in such a manufacturing protocol can adversely affect all probes using this failed step, and is in general impossible to disentangle from experimental variation when using such a defective array. The idea of designing special quality control probes to detect a failed step was first formulated by Hubbell and Pevzner. We consider an alternative formulation of this problem and use a combinatorial design approach to solve it. Our results improve over prior work in guaranteeing coverage of all protocol steps and in being able to tolerate a greater number of unreliable probe intensities.

1. Introduction

Recent advances in DNA microarray technology have allowed biologists to obtain expression profiles of the genes in an organism in a quantitative and high throughput fashion. An important class of DNA microarray technology, called "oligo arrays", manufactures short single strands of DNA (called *probes*) onto a glass surface using photolithography [1]. The glass surface (or *array*) has a well-defined set of addresses (or *spots*) where the probes are grown. The manufacturing *protocol* is a sequence of steps $N_1 N_2 \ldots N_n$, each with an associated nucleotide $N_i \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. Conceptually, at the i^{th} step of the protocol a *mask* is placed on the glass array and the array is exposed to a solution containing the nucleotide N_i. This causes the probes at the positions on the array that are not masked to be extended by one base, N_i. The rest of the probes do not change during this step. The process is repeated with a new mask at each step, to build a diverse assortment of probes.

An altered or missed step in the array's manufacturing protocol can adversely affect all probes using the failed step, and thus their hybridization behavior with targets. The error ensuing from a faulty manufacturing step may well be impossible to disentangle from experimental variation when using the defective array. The problem of developing a quality control mechanism that detects during the manufacturing process if a step has failed is therefore of clear practical importance.

One approach to the quality control problem, formulated first by Hubbell and Pevzner [2], is to design a small set of special quality control probes. Their ingenious idea was to manufacture the same probe sequence at a number of different spots, each spot using a different schedule of steps of the protocol. A protocol step i therefore has an associated set P_i of quality control spots that use this manufacturing step. These quality control probes are then hybridized with a complementary fluorescent target. The intensities within the set P_i provide a "signature" for the quality of step i. If many of the intensities within P_i are significantly lower than the remaining intensities, this is a good indication of step i being flawed. This is because all the spots have the same sequence and should therefore have similar hybridization behavior (hence similar intensities) if they are correctly manufactured. The focus of the work of Hubbell and Pevzner is to generate sets P_i that are sufficiently large and sufficiently unique that a failed step can be identified even in the presence of some unreliable spot intensities. This method is then used repeatedly for each probe in a supplied set $\mathcal{S}$ of probes. However, there may be steps in a protocol that cannot be used in manufacturing any of the probes in a given set $\mathcal{S}$. Assuming that $\mathcal{S}$ is supplied implies that the failure of such a step cannot be detected. Moreover, since there is no coordination among the solutions generated for distinct probes (the algorithm being used separately on each probe), Hubbell and Pevzner do not exploit the ability of the probes to collectively make the set of spots using a protocol step as large and as unique as possible.

We consider an alternative formulation of this problem that does not assume that the quality control probe sequences are supplied. We take the choice of the probe sequences into our own hands in order to guarantee that every protocol step is well covered by the quality control mechanism. Our design ensures that the number of distinct probes is small and that they hybridize poorly with themselves and with each other. This is a necessary constraint because if probes hybridize well with themselves or each other, then their corresponding complementary targets will too, rendering them unavailable to hybridize to the probes [3]. Our design further ensures that each probe hybridizes well only with the target that is complementary to it, and hybridizes poorly with the targets meant for the other probes. This property allows us to use multiple quality

control targets (up to 4 in our current designs) simultaneously, thereby relaxing the requirement of Hubbell and Pevzner [2] that all probes are complementary to substrings of a single target.

The fact that we want balanced and sufficiently unique signatures for all steps in the protocol suggests a connection to the elegant theory of combinatorial design. For our purposes, a combinatorial design is just a 0-1 matrix with appropriate balance and uniqueness properties. The chief contribution of this work is to solve the quality control problem by developing a framework that builds on techniques from combinatorial design. For a preview, see Figure 2.

Because of space limitations, many details and most proofs are omitted from this version, but can be found in the full paper [4].

2. The Quality Control Problem

A quality control scheme for a protocol with n steps using m spots can be viewed as an $m \times n$ 0-1 matrix $\mathcal{Q}$, with each column representing a protocol step and each row representing a spot. Each column of $\mathcal{Q}$ is labelled with the nucleotide used in that step. The entry $\mathcal{Q}_{ij}$ is 1 if and only if step j was used in manufacturing the probe at spot i. We will refer to such a matrix $\mathcal{Q}$ as a *Quality Control* (QC) matrix. The sequence of the oligonucleotide at spot i can be read out by concatenating the labels of the columns at which row i has a 1.

The probes manufactured at the m quality control spots are not all different. There will in general be c distinct probes, with several spots containing the same probe but manufactured using different schedules of steps of the protocol.

To actually perform quality control of a protocol, the quality control probes defined by $\mathcal{Q}$ are manufactured using the protocol onto m reserved spots on each chip of a wafer [5]. The manufacturer takes one chip from the wafer and tests it as follows: the chip is hybridized with fluorescent targets complementary to the c probes, scanned, and the resulting vector of m intensity values is used to determine which step, if any, failed.

Definition 2.1: (QC Problem) Given a protocol $\mathcal{P}$ with n steps up to 1 of which may fail, and a budget of m quality control spots up to d of which may be unreliable, construct an $m \times n$ QC matrix $\mathcal{Q}$ such that an intensity vector $\mathcal{I}$ of the m spots manufactured using $\mathcal{Q}$ allows unique identification of the failed step, if any.

The problem we solve in this work is not quite as general as the one stated in Definition 2.1. We cannot hope to take arbitrary parameter values n, m, and d as input and produce a QC matrix $\mathcal{Q}$ that meets the specifications. We

explain in Section 2.2 why solving this general version would entail solving long-standing open questions in combinatorial design. However we are able to produce QC matrices for a wide range of values of n, m, and d that covers the desired settings in practice. We also do not solve this for arbitrary protocols $\mathcal{P}$, but rather a specific set of 24 periodic protocols, namely, $[\pi(\mathrm{ACGT})]^{n/4}$, where π is any permutation and n is a multiple of 4 in the range $60 \leq n \leq 132$. Again, this covers the typical protocols in practice.

2.1. Assumptions

1. Step failure model: when a step fails, a spot will show a low intensity if and only if the failed step was used in manufacturing the probe at that spot, with up to d exceptions. When no step fails, each spot will show a high intensity, with up to d exceptions.

2. Spots containing different probes in general may have different hybridization behaviors. Hence we will not compare intensity values of two different probe sequences. We will also not make the assumption that, within the set of spots sharing the same probe, we can distinguish between all intensities high and all low.

3. We are allowed multiple quality control targets that are designed so as to hybridize poorly to themselves and to each other. Each probe is designed to hybridize poorly to all but one of these targets.

Definition 2.2: We say that two single-stranded nucleotide sequences *hybridize poorly* if and only if, when they are arranged in antiparallel fashion, shifted an arbitrary offset with respect to each other, at least two out of every four consecutive pairs of aligned bases are not complementary. A set S of such sequences is said to *hybridize poorly* if and only if every sequence $s \in S$ hybridizes poorly to itself, to every other sequence in S, and to the complement of every sequence in S that is not a rotation of s.[a]

2.2. Identifying the Failed Step

In this section we define a property of a QC matrix $\mathcal{Q}$, called "separation," and establish that high separation is sufficient to identify any one failed step when up to d spots may show unreliable intensities.

Definition 2.3: Let $\mathcal{Q}$ be an $m \times n$ QC matrix with c distinct probes $\{q_k \mid 1 \leq k \leq c\}$. Let p_i be the probe at row i, $1 \leq i \leq m$. By convention, define $\mathcal{Q}_{i0} = 0$ for all $1 \leq i \leq m$. For any k with $1 \leq k \leq c$, and any pair

[a] For example, the sequence CACG CACG is a rotation of the sequence ACGC ACGC.

$j \neq j'$ with $0 \leq j, j' \leq n$, let

$$\begin{aligned} D_k(j,j') &= \#\{i \mid p_i = q_k \text{ and } \mathcal{Q}_{ij} \neq \mathcal{Q}_{ij'}\}, \\ L_k(j,j') &= \#\{i \mid p_i = q_k \text{ and } (\mathcal{Q}_{ij} \neq 1 \text{ or } \mathcal{Q}_{ij'} \neq 0)\}, \\ R_k(j,j') &= \#\{i \mid p_i = q_k \text{ and } (\mathcal{Q}_{ij} \neq 0 \text{ or } \mathcal{Q}_{ij'} \neq 1)\}. \end{aligned}$$

The *separation* of $\mathcal{Q}$ is defined to be:

$$sep(\mathcal{Q}) = \min_{\substack{0 \leq j,j' \leq n \\ j \neq j'}} \sum_{k=1}^{c} \min(D_k(j,j'), L_k(j,j'), R_k(j,j')). \tag{1}$$

The D_k portion of Definition 2.3 has an intuitive explanation based on the *Hamming distance* between two vectors, which is the number of corresponding positions at which the two vectors have unequal values. A large Hamming distance between columns j and j' of $\mathcal{Q}$ is necessary in order to be able to detect the difference between step j failing and step j' failing. Similarly, a large Hamming distance between column j of $\mathcal{Q}$ and the conventional column 0 (i.e., a large number of ones in column j) is necessary in order to detect the difference between step j failing and no step failing.

The L_k and R_k portions of Definition 2.3 capture the part of Assumption 2 from Section 2.1 that one may not be able to differentiate between all probe intensities high and all low, which is why the D_k portion alone is not sufficient. For example, suppose step j were used in *every* spot i. Even if no spot failed, if step j were to fail all spots would show equal (low) intensities. One might well not be able to distinguish this case from no step failing, in which all spots would also show equal (high) intensities. Using a similar explanation to one given above, this portion implies that each column of $\mathcal{Q}$ has a large number of zeros.

The intensity vector $\mathcal{I}$ is a vector of m real numbers, giving an intensity reading for each of the m spots. We wish to interpret these real numbers as high ("0"), low ("1"), or unreadable ("?"). This interpretation is subject to reasonable constraints that two similar intensities of the same probe are not interpreted as one high and one low, and two distant intensities of the same probe are not interpreted as both high or both low.

Let $\Phi(\mathcal{I}) \in \{0, 1, ?\}^m$ be such an interpretation of intensity vector $\mathcal{I} \in \Re^m$, where $\Re$ is the set of real numbers. The reason why high intensity corresponds to "0" and low to "1" is because the object is to use this interpretation vector to identify which column of the QC matrix it resembles most. When step j fails and none of the spots are faulty, the intensity vector interpretation $\Phi(\mathcal{I})$

one expects to see is exactly the 0-1 vector forming the j^{th} column of the QC matrix. In general up to d spots may be unreliable, so if step j fails, $\Phi(\mathcal{I})$ will equal the j^{th} column of the QC matrix with at most d exceptional positions. Note that not all the d unreliable spots need be interpreted as "?": some may be erroneously interpreted as high or low.

Theorem 2.4: Suppose $sep(\mathcal{Q}) \geq 2d+1$ and $\mathcal{I}$ is the intensity vector of the m spots. Then, for $1 \leq j \leq n$, step j fails if and only if there is an interpretation Φ of $\mathcal{I}$ such that $\delta(\mathcal{Q}_{*j}, \Phi(\mathcal{I})) \leq d$, where δ is the Hamming distance and $\mathcal{Q}_{*j}$ is the j^{th} column of $\mathcal{Q}$. No step fails if and only if there is an interpretation Φ of $\mathcal{I}$ such that $\delta(0^m, \Phi(\mathcal{I})) \leq d$.

Given spot failure tolerance d, an $m \times n$ QC matrix $\mathcal{Q}$ with $sep(\mathcal{Q}) \geq 2d+1$, and an intensity vector $\mathcal{I} \in \Re^m$, Theorem 2.4 can be applied to identify which protocol step, if any, has failed. An algorithm solving this problem must check if, for any j, $0 \leq j \leq n$, there exists an interpretation Φ such that $\delta(\mathcal{Q}_{*j}, \Phi(\mathcal{I})) \leq d$. If so, it returns the value j as the step that has failed. (As in Definition 2.3, Q_{*0} by convention is the vector 0^m, and a returned value of $j = 0$ corresponds to no step having failed.) In the full version of this paper [4], we describe an $O(mn + m \log m)$ time algorithm for performing this task.

The following theorem provides one simple way to combine QC matrices, and illustrates a tradeoff between the goals of maximizing separation and minimizing the number of spots.

Theorem 2.5: Suppose that $\mathcal{Q}_1$ is an $m_1 \times n$ QC matrix, and $\mathcal{Q}_2$ is an $m_2 \times n$ QC matrix. Then the union $\mathcal{Q}_1 + \mathcal{Q}_2$ of their rows has n steps, $m_1 + m_2$ spots, and $sep(\mathcal{Q}_1 + \mathcal{Q}_2) \geq sep(\mathcal{Q}_1) + sep(\mathcal{Q}_2)$.

We are now in a position to state the precise design problem we solve. The array manufacturer specifies as inputs the number n of steps, the protocol, and the length k of each probe. The QC design problem is to construct an $m \times n$ QC matrix $\mathcal{Q}$ with k ones per row such that the number m of spots is small and $sep(\mathcal{Q})$ is large. Furthermore, the set of c distinct probes hybridizes poorly, according to Definition 2.2. In our designs, we never use more than $c = 8$ distinct probes.

One cannot expect to optimize both the objective functions m and $sep(\mathcal{Q})$ in a single QC matrix. For instance, Theorem 2.5 says that duplicating the spots of $\mathcal{Q}$ simultaneously doubles m and $sep(\mathcal{Q})$. Instead, in Section 4 we will construct a variety of QC matrices $\mathcal{Q}$ that offer the manufacturer a spectrum of choices for m and $sep(\mathcal{Q})$.

One should also not expect to find an algorithm that, given arbitrary values n and m, computes an $m \times n$ QC matrix $\mathcal{Q}$ that maximizes $sep(\mathcal{Q})$. This is likely to be infeasible at the present time, because even the existence of certain combinatorial designs (such as a Hadamard matrix of order $4t$, which

is equivalent to a $(4t-1) \times (4t-1)$ QC matrix $\mathcal{Q}$ with $sep(\mathcal{Q}) = 2t-1$) is a long-standing open problem [6].

3. A Combinatorial Design Approach

We will assume that the protocol is $(\mathrm{ACGT})^{n/4}$, generalizing to other protocols in the full paper [4].

3.1. Balanced Codes

A good QC matrix $\mathcal{Q}$ has many of the properties of a good error-correcting code, which is a type of combinatorial design: if one thinks of the columns of $\mathcal{Q}$ as binary codewords, then one part of Definition 2.3 (the constraint on D_k) guarantees that the Hamming distance between any pair of codewords is at least $sep(\mathcal{Q})$. However, good QC matrices have many more constraints that make their design more complicated than that of error-correcting codes. We introduce a specialized type of code to satisfy these constraints.

Definition 3.1: A *balanced binary code* with parameters $(v, b, r_{\min}, r_{\max}, k, d_{\min})$ is a $b \times v$ 0-1 matrix with the following properties:

1. Every row contains exactly k ones.
2. The minimum number of ones in any column is $r_{\min}$, and the maximum is $r_{\max}$.
3. The minimum Hamming distance between any pair of columns is $d_{\min}$.

A subset of the codewords from certain types of error-correcting codes, such as Hadamard codes and quadratic residue codes [7], form balanced codes. However, our major source of balanced code constructions comes from 2-designs:

Definition 3.2 (Colbourn and Dinitz [8]): A *2-design* with parameters (v, b, r, k, λ) is a $b \times v$ 0-1 matrix D with the following properties:

1. Every row contains exactly k ones.
2. Every column contains exactly r ones.
3. For every pair j, j' of distinct columns, there are exactly λ rows i such that $D_{i,j} = D_{i,j'} = 1$.

Proposition 3.3: Any 2-design with parameters (v, b, r, k, λ) is a balanced code with parameters $(v, b, r, r, k, 2(r-\lambda))$.

Another source of balanced codes comes from the following product construction.

Theorem 3.4: Let C' be a balanced code with parameters $(v', b', r'_{\min}, r'_{\max}, k', d'_{\min})$ and C be a balanced code with parameters $(v, b, r_{\min}, r_{\max}, k, d_{\min})$. Then there is a balanced code $C' \times C$ with parameters

$$(v'v, b'b, r'_{\min}r_{\min}, r'_{\max}r_{\max}, k'k, \min(d'_{\min}r_{\min}, d_{\min}r'_{\min})).$$

Proof: Replace every one in C' by a copy of C, and every zero in C' by a $b \times v$ matrix of zeros. □

Balanced codes do not capture the notion of poor hybridization. A "QC block" is just a balanced code with an additional hybridization constraint:

Definition 3.5: A *QC block* for a protocol $\mathcal{P}$ is a $b \times v$ balanced code in which the b probes $p_1, p_2, \ldots, p_b$ are all distinct and, for every integer s, the set $\{p_1^s, p_2^s, \ldots, p_b^s\}$ hybridizes poorly (see Definition 2.2).

An example of an 8×8 QC block with parameters $(8, 8, 4, 4, 4, 4)$ is given in Figure 1. Its eight poorly hybridizing probes are $(\text{ACGC})^s$, $(\text{TAGT})^s$, $(\text{CACG})^s$, $(\text{AGTT})^s$, $(\text{ACAT})^s$, $(\text{GTCG})^s$, $(\text{ATAC})^s$, and $(\text{CGGT})^s$. Its four complementary targets are GCGT ... GCGT G, AACT ... AACT A, ATGT ... ATGT AT, and CGAC ... CGAC CG.

3.2. Product Construction of QC Matrices

The method we will use to construct good QC matrices is to apply the product construction of Theorem 3.4, with C' a balanced code and C a QC block. Figure 2 shows an example, where C' consists of ten codewords from the 8-Hadamard code[7], and C is the QC block of Figure 1.

If the parameters of C' are $(v', b', r'_{\min}, r'_{\max}, k', d'_{\min})$ and the parameters of C are $(v, b, r_{\min}, r_{\max}, k, d_{\min})$, then the QC matrix $C' \times C$ will have $v'v$ steps, $b'b$ spots, and b distinct probes, each of length $k'k$ and each occurring at b' distinct spots. More specifically, if $p_1, p_2, \ldots, p_b$ are the distinct probes of C, then $p_1^{k'}, p_2^{k'}, \ldots, p_b^{k'}$ are the distinct probes of $C' \times C$. By Definition 3.5, this set of distinct probes hybridizes poorly.

What remains is to determine $sep(C' \times C)$, in order to be able to apply Theorem 2.4.

Theorem 3.6: If C' is a balanced code with parameters $(v', b', r'_{\min}, r'_{\max}, k', d'_{\min})$ and C is a QC block with parameters $(v, b, r_{\min}, r_{\max}, k, d_{\min})$, then

$$\begin{aligned} sep(C' \times C) = \min(\ & d'_{\min}r_{\min}, \\ & r'_{\min}\min(r_{\min}, d_{\min}), \\ & (b' - r'_{\max})\min(r_{\min}, d_{\min})\ ). \end{aligned}$$

A	C	G			C		
			T	A		G	T
	C			A	C	G	
A		G	T				T
A	C			A			T
		G	T		C	G	
A			T	A	C		
	C	G				G	T

Figure 1: An 8×8 QC block. For ease of visualization, the figure shows blanks instead of zeros, and the appropriate nucleotide from the protocol instead of ones.

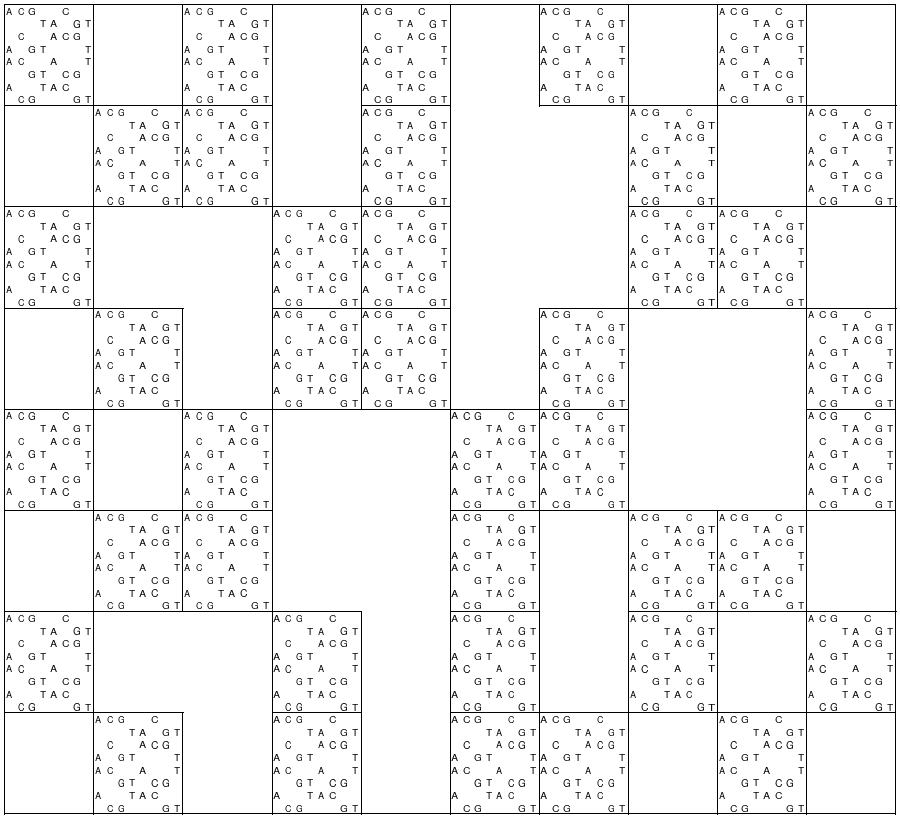

Figure 2: The product of 10 codewords from the 8-Hadamard code and the 8×8 QC block of Figure 1, resulting in a 64×80 QC matrix $\mathcal{Q}$ with minimum separation $sep(\mathcal{Q}) = 16$.

As an example, if C is the 8×8 QC block of Figure 1, then

$$sep(C' \times C) = 4 \min(d'_{\min}, r'_{\min}, b' - r'_{\max}).$$

4. Results: Achieved QC Matrices

Table 1 shows some of the QC matrices achievable by using the product construction of Section 3.2. Each row of the table describes a QC matrix that is the product of the balanced code specified in the last column and the QC block specified in the penultimate column. For example, the QC matrix shown in Figure 2 corresponds to the row of the table with 80 steps and 64 spots.

The separations in column 4 of the table are calculated using Theorem 3.6. For each fixed number of steps (column 1), the table offers a small spectrum of designs to suit the manufacturer's spot budget and spot failure tolerance (columns 3–4). Arbitrary linear combinations of these designs can be formed according to Theorem 2.5, to provide a broader spectrum of choices.

The manufacturer uses Table 1 to look up the QC matrix $\mathcal{Q}$ for the appropriate choice of parameters in the first four columns of the table, where the "sep" parameter is chosen to be greater than twice the number of faulty spots the manufacturer is willing to tolerate. The QC matrix $\mathcal{Q}$ is used to manufacture the quality control probes onto reserved spots, which are hybridized with complementary fluorescent targets. The resulting intensity vector $\mathcal{I}$ is then used along with $\mathcal{Q}$ to identify the failed step, if any, using the algorithm following Theorem 2.4.

The 8×8 QC block has already been presented in Figure 1. The 6×12, 6×8, and 4×4 QC blocks are given in the full paper [4].

5. Open Problems

1. Handle more than one step failure. Binary superimposed codes [11] appear to be a promising way to extend our hierarchical design approach to handle multiple step failures.
2. Relax the step fault model. When a step fails, not every spot using that step will have the same low intensity. The change in intensity more realistically will be a function of how far from the center of the probe the failed step is (Lipschutz *et al.* [1]).
3. Develop a general technique for designing balanced codes. These designs appear not to have been studied prior to this, even in the combinatorial design literature [12]. Alon and Tompa [13] have developed one such technique, resulting in many new balanced codes and QC matrices.

Table 1: Some basic QC matrices achievable by the product construction of Section 3.2. The second column shows the probe length. The last two columns show the QC block and balanced code whose product yields the QC matrix. In the last column, a list of 5 parameters indicates a 2-design (Definition 3.2), "$\times$" indicates a product code (Theorem 3.4), "$+i$" indicates the addition of i extra columns that maintain the balanced code properties [4], and GF(q) refers to balanced codes derived from polynomials over finite fields [9]. The 2-designs referenced in the last column can be found in the compendium of Mathon and Rosa [10], and the error-correcting codes in the survey of Tonchev [7].

steps	leng	spots	sep	block	balanced code
60	16	60	14	4x4	(15,15,8,8,4)
60	18	140	28	4x4	(15,35,21,9,12)
60	20	168	28	4x4	(15,42,28,10,18)
64	16	42	6	6x8	7-Hadamard code
64	16	44	10	4x4	11-Hadamard code
64	16	48	12	4x4	12-Hadamard code
64	16	64	16	8x8	8-Hadamard code
64	20	64	12	4x4	(16,16,10,10,6)
64	16	120	30	4x4	(16,30,15,8,7)
64	18	320	70	4x4	(16,80,45,9,24)
68	16	136	32	4x4	(17,34,16,8,7)
72	18	44	10	4x4	11-Hadamard code
72	18	48	12	4x4	12-Hadamard code
72	16	54	8	6x8	(3,3,2,2,1) x (3,3,2,2,1)
72	16	72	16	8x8	(3,3,2,2,1) x (3,3,2,2,1)
72	20	84	12	6x8	(7,14,8,4,4)+2
72	20	108	16	6x8	(9,18,10,5,5)
72	20	112	24	8x8	(7,14,8,4,4)+2
72	18	136	34	4x4	(18,34,17,9,8)
72	20	144	32	8x8	(9,18,10,5,5)
76	18	76	18	4x4	(19,19,9,9,4)
76	20	76	18	4x4	(19,19,10,10,5)
80	20	44	10	4x4	11-Hadamard code
80	20	48	12	4x4	12-Hadamard code
80	20	64	16	8x8	8-Hadamard code
80	16	90	12	6x8	(10,15,6,4,2)
80	16	120	24	8x8	(10,15,6,4,2)
80	20	152	38	4x4	(20,38,19,10,9)
80	18	160	24	4x4	(4,4,3,3,2) x (5,10,6,3,3)
80	16	380	76	4x4	(20,95,38,8,14)
84	16	42	6	6x12	(7,7,4,4,2)
84	20	126	12	6x12	(7,21,15,5,10)
84	18	140	30	4x4	(21,35,15,9,6)
84	20	168	40	4x4	(21,42,20,10,9)
88	20	66	10	6x8	(11,11,5,5,2)
88	20	84	12	6x8	(7,14,6,3,2)+4
88	20	88	20	8x8	(11,11,5,5,2)
88	20	112	24	8x8	(7,14,6,3,2)+4
88	20	144	32	8x8	(9,18,8,4,3)+2
88	16	264	48	4x4	(22,66,24,8,8)
88	20	308	70	4x4	(22,77,35,10,15)
96	16	42	6	6x12	7-Hadamard code
96	16	48	8	6x12	8-Hadamard code
96	18	48	4	4x4	(4,4,3,3,2) x 3-Hadamard
96	18	64	8	4x4	(4,4,3,3,2) x 4-Hadamard
96	16	84	14	6x12	(8,14,7,4,3)
96	20	90	12	6x8	(10,15,6,4,2)+2
96	20	120	24	8x8	(10,15,6,4,2)+2
96	18	160	20	4x4	(4,4,3,3,2) x (6,10,5,3,2)
96	16	276	46	4x4	(24,69,23,8,7)
100	18	100	18	4x4	(25,25,9,9,3)
100	20	160	32	4x4	(25,40,16,10,6)
100	16	300	48	4x4	(25,75,24,8,7)
104	16	78	8	6x8	(13,13,4,4,1)
104	16	104	16	8x8	(13,13,4,4,1)
104	20	234	30	6x8	(13,39,15,5,5)
104	20	260	50	4x4	(26,65,25,10,9)
104	20	312	60	8x8	(13,39,15,5,5)
108	18	36	4	4x4	degree 2 over GF(3)
108	16	54	8	6x12	(3,3,2,2,1) x (3,3,2,2,1)
108	20	84	12	6x12	(7,14,8,4,4)+2
108	20	108	16	6x12	(9,18,10,5,5)
108	18	156	26	4x4	(27,39,13,9,4)
108	20	216	40	4x4	(3,3,2,2,1) x (9,18,10,5,5)
112	20	108	12	6x8	(3,3,2,2,1) x (4,6,3,2,1) + 2
112	18	112	12	4x4	(4,4,3,3,2) x (7,7,3,3,1)
112	20	144	24	8x8	(3,3,2,2,1) x (4,6,3,2,1) + 2
112	20	168	30	4x4	(28,42,15,10,5)
112	18	336	54	4x4	(28,84,27,9,8)
116	16	232	32	4x4	(29,58,16,8,4)
120	20	42	6	6x12	7-Hadamard code
120	20	48	8	6x12	8-Hadamard code
120	20	66	10	6x12	11-quadratic residue code
120	16	90	12	6x12	(10,15,6,4,2)
120	20	108	18	6x12	(10,18,9,5,4)
120	20	168	28	6x12	(8,28,14,4,6)+2
120	16	240	32	8x8	(3,3,2,2,1) x (5,10,4,2,1)
120	20	348	58	4x4	(30,87,29,10,9)
124	20	124	20	4x4	(31,31,10,10,3)
128	16	96	8	6x8	degree 1 over GF(4)
128	16	120	10	6x8	(16,20,5,4,1)
128	16	128	16	8x8	degree 1 over GF(4)
128	16	160	20	8x8	(16,20,5,4,1)
128	16	192	24	8x8	(4,6,3,2,1) x 4-Hadamard
128	20	288	30	6x8	(16,48,15,5,4)
128	20	384	60	8x8	(16,48,15,5,4)
132	20	66	10	6x12	(11,11,5,5,2)
132	20	84	12	6x12	(7,14,6,3,2)+4
132	20	108	16	6x12	(9,18,8,4,3)+2
132	18	176	24	4x4	(33,44,12,9,3)
132	16	330	40	6x12	(11,55,20,4,6)

Acknowledgments

We thank Noga Alon, Charlie Colbourn, Earl Hubbell, Yuan Ma, and David Smith for sharing their expertise with us. This material is based upon work supported in part by a Sloan/DOE Fellowship in Computational Molecular Biology, by the National Science Foundation and DARPA under grant DBI-9601046, and by the National Science Foundation under grant DBI-9974498.

References

1. Robert J. Lipshutz, Stephen P. A. Fodor, Thomas R. Gingeras, and David J. Lockhart. High density synthetic oligonucleotide arrays. *Nature Genetics Supplement*, 21:20–24, 1999.
2. Earl Hubbell and Pavel A. Pevzner. Fidelity probes for DNA arrays. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 113–117, Heidelberg, Germany, August 1999. AAAI Press.
3. David Smith. Affymetrix, 1999. Personal communication.
4. Rimli Sengupta and Martin Tompa. Quality control in manufacturing oligo arrays: a combinatorial design approach. Technical Report 2000-08-03, Department of Computer Science and Engineering, University of Washington, September 2000. `ftp://ftp.cs.washington.edu/tr/2000/08/UW-CSE-00-08-03.PS.Z`.
5. Earl Hubbell, 1999. Personal communication.
6. R. Craigen. Hadamard matrices and designs. In Colbourn and Dinitz[8], pages 370–377.
7. Vladimir D. Tonchev. Codes. In Colbourn and Dinitz[8], pages 517–542.
8. Charles J. Colbourn and Jeffrey H. Dinitz, editors. *The CRC Handbook of Combinatorial Designs*. CRC Press, 1996.
9. Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.
10. Rudolf Mathon and Alexander Rosa. $2-(v,k,\lambda)$ designs of small order. In Colbourn and Dinitz[8], pages 3–40.
11. W. H. Kautz and R. C. Singleton. Non-random binary superimposed codes. *IEEE Transactions on Information Theory*, 10:363–377, 1964.
12. Charles J. Colbourn, 2000. Personal communication.
13. Noga Alon and Martin Tompa. Balanced codes from near difference sets. In preparation, 2000.