Recovering sparse signals using sparse measurement matrices in compressed DNA microarrays [*]

HARIS VIKALO[a] FARZAD PARVARESH[b] SIDHANT MISRA[c] BABAK HASSIBI[b]

[a]ECE Department, The University of Texas, Austin, TX 78701
[b]EE Department, California Institute of Technology, Pasadena, CA 91125
[c]Indian Institute of Technology, Kanpur, India
e-mail: hvikalo@ece.utexas.edu, farzad, hassibi@caltech.edu

Abstract

Microarrays (DNA, protein, etc.) are massively parallel affinity-based biosensors capable of detecting and quantifying a large number of different genomic particles simultaneously. Among them, DNA microarrays comprising tens of thousands of probe spots are currently being employed to test multitude of targets in a single experiment. In conventional microarrays, each spot contains a large number of copies of a single probe designed to capture a single target, and hence collects only a single data point. This is a wasteful use of the sensing resources in comparative DNA microarray experiments, where a test sample is measured relative to a reference sample. Typically, only a fraction of the total number of genes represented by the two samples is differentially expressed, and thus a vast number of probe spots may not provide any useful information. To this end we propose an alternative design, the so-called *compressed microarrays*, wherein each spot contains copies of several different probes and the total number of spots is potentially much smaller than the number of targets being tested. Fewer spots directly translates to significantly lower costs due to cheaper array manufacturing, simpler image acquisition and processing, and smaller amount of genomic material needed for experiments. To recover signals from compressed microarray measurements, we leverage ideas from compressive sampling. For sparse measurement matrices, we propose an algorithm that has significantly lower computational complexity than the widely-used linear-programming-based methods, and can also recover signals with less sparsity.

Index Terms:

DNA microarrays, compressive sampling, sparse measurements

[*] This work was supported in part by a Grubstake Award from California Institute of Technology, a grant from the David and Lucille Packard Foundation, and by the Millard and Muriel Jacobs Genetics and Genomics Laboratory at Caltech.

1 Introduction

Over the past decade, high-throughput assay technologies have received a lot of attention in the genomic research community and biotech industry. Among these technologies, DNA microarrays have attracted much interest due to their capability to test as many as tens of thousands of different nucleotide sequences simultaneously. This stands in contrast to traditional techniques that are able to analyze only a small number of nucleotide sequences at a time.

Sensing in DNA microarrays [1]-[8] is based on the process of hybridization in which complementary DNA strands bind to each other creating structures in lower energy states. Typically, the surface of a DNA microarray comprises an array of spots, each spot containing a large number of identical single-stranded DNA sequences (*probes*) designed to capture copies of a single DNA molecule (*target*) of interest. DNA microarrays are often used to measure gene expression levels, i.e., to quantify the process of transcription of DNA information into messenger RNA molecules (mRNA). The information transcribed into mRNA is further translated to proteins, the molecules that perform most of the functions in cells. Therefore, by measuring gene expression levels, we may be able to infer critical information about the functionality of cells or whole organisms [9]-[11], study diseases and the effects of drugs on them [12]-[18], etc. DNA microarrays are often used to compare the gene expression levels of a test sample with that of a reference sample. In a typical scenario, only a fraction of the total number of genes (e.g., $30,000$ in an entire genome) is found to be differentially expressed. Therefore, a large fraction of a microarray does not contribute any information about the subset of the genes that are differentially expressed. To remedy this, in [19] a microarray architecture comprising spots that contain mixtures of several different probes was proposed, so that a signal measured at each probe spot is potentially a combination of as many targets. This allows acquisition of multiple data points for each of the targets being tested, including those that are indeed differentially expressed.

However, the signal recovery in the *composite microarrays* of [19] does not exploit inherent sparseness of the signal.

In this paper, we leverage ideas from *compressive sampling* to enable more economic usage of the sensing resources in composite microarrays. The essential idea of compressive sampling is that we may be able to recover an inherently sparse signal by using far fewer measurements than what is typically needed for a signal which is not sparse [21]-[27]. Compressive sampling is closely related to the problem of solving an underdetermined system of linear equation with a sparseness constraint – which is precisely the problem of signal recovery in composite microarrays with fewer probe spots than probes. By judiciously choosing probes comprising each spot, we may be able to recover sparse signal from a microarray wherein the number of probe spots is significantly reduced. We refer to such platforms as *compressed microarrays*. Having fewer probe spots translates to lower costs due to cheaper array manufacturing, simpler image acquisition and processing, and smaller amount of genomic material needed for experiments. Moreover, decreasing sample volume size is critically important in order to further the applications of microarray technology in diagnostics, and enable environmental monitoring applications.

Typically, DNA microarrays are manufactured by either spotting (i.e., printing) probe molecules in their allotted spots, or by a direct probe synthesis on the array. In this paper we focus on the former, i.e., we propose and study compressed microarrays manufactured by probe spotting, and l_1-optimization techniques for a sparse signal recovery therein. On the other hand, recent work [38] proposes the design of probes, each of which can potentially capture several different targets, and employs the belief propagation approach to facilitate sparse signal recovery. The design of probes in [38] can be quite challenging; in particular, balancing probes selectivity, specificity, as well as performing array calibration (i.e., determining the strength of binding of each target analyte to its corresponding probe) can be problematic. Our approach, however, employs already-designed probe sets and simply requires mixing a number of different probes prior to spotting them on an array – a procedure which is readily feasible. We should also note that, for simplicity of array manufacturing, we insist that the number of probes constituting spots in compressed microarrays is not very large (which in turn poses constraint that the so-called sampling matrix be sparse itself). As we will show in the paper, this additional constraint on the compressed microarray design is actually beneficial from the signal processing perspective since it enables the development of novel efficient signal processing algorithms for sparse signal recovery.

The paper is organized as follows. In Chapter 2, we give a brief background on DNA microarrays and compressive sampling. Chapter 3 introduces the compressed DNA microarray architecture and presents a simulation study of their performance. In Chapter 4, a novel, computationally efficient algorithm for the recovery of sparse signals specifically tailored for applications to compressed DNA microarrays is presented. Experimental results are described in Chapter 5, while the summary and conclusions are given in Chapter 6.

2 Background

2.1 DNA microarrays

To evaluate the abundance of target molecules in a biological sample, DNA microarrays rely on hybridization, a process in which single-stranded nucleotide sequences bind to each other creating structures in lower energy states. In fluorescent-based systems, the target molecules are labeled with fluorescent tags prior to the actual experiment. When applied to the microarray and under appropriate experimental conditions (e.g., temperature and salt concentration), labeled target molecules begin hybridizing to the complementary probes. The process of hybridization may take hours before it reaches the steady-state. Then, the array is washed, at which point unbound target molecules are removed. Finally, the fluorescent molecules attached to targets bound to probe spots are excited and their emission is measured to obtain an image. The image intensities are correlated to the hybridization process, and thus provide the information about the amount of targets under evaluation.

DNA microarrays are often used to evaluate gene expression levels, i.e., quantify the process of transcription of DNA data into mRNA. In *gene expression profiling* applications, DNA microarrays compare the gene expression levels of a test sample with the gene expression levels of a reference sample. For instance, one may be interested in comparing gene expression between normal and diseased (e.g., cancerous) cells. This is typically done using two-color microarrays, in which the two samples are labeled with two different types of fluorescent tags; in particular, the two types of tags are capable of emitting light at different wavelengths. The difference of the two signals is used as an indication of the relative amounts of the mRNA in the test and the reference sample. If the amount of mRNA in the test sample is much smaller or larger than the amount of mRNA in the reference sample, the corresponding gene is said to be under-expressed or over-expressed, respectively.

2.2 Compressive sampling

In compressive sampling, we are interested in estimating an n-dimensional signal $\mathbf{x}$ which has no more than k non-zero entries. (Note that we do not know a priori the locations of the non-zero entries.) So, $k < n$; in fact, we frequently focus on applications where $k << n$.

The vector $\mathbf{x}$ is not directly observable. Instead, we observe m linear combinations of the entries of $\mathbf{x}$,

$$y_i = \sum_{j=1}^{n} A_{ij} x_j, \quad i = 1, 2, \ldots, m, \tag{1}$$

where $k < m < n$. In other words, the number of measurements that we collect is smaller than the size of the vector $\mathbf{x}$, yet larger than the number of its non-zero entries. Collecting the coefficients A_{ij} into an $m \times n$

matrix A, we can write (1) in a matrix form

$$\mathbf{y} = A\mathbf{x}. \tag{2}$$

The underdetermined system of equations (2) may, in principle, be solved by using the fact that the vector $\mathbf{x}$ is sparse. In particular, we could consider all possible combinations of k columns of A, and attempt to solve the corresponding system of equations which is overdetermined (since each one has m equations with k unknowns). Assuming that each of these combinations of columns forms a matrix with a full rank, at least one of the overdetermined systems will have a solution. This solution determines the positions and values of the non-zero entries in $\mathbf{x}$. However, the outlined approach requires solving up to $\left(\begin{array}{c} n \\ k \end{array}\right)$ systems of equation, which, for $k << n$, is approximately $O(n^k)$, and is clearly practically infeasible.

On the other hand, for a long time it has been known that constrained l_1 minimization,

$$\min_{\mathbf{x},\, A\mathbf{x}=\mathbf{y}} \|\mathbf{x}\|_1, \tag{3}$$

as well as the related constrained quadratic programming (so-called Lasso [20]),

$$\min \|y - Ax\|_2 \ \textit{ subject to } \ \|x\|_1 \leq \beta, \tag{4}$$

where $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$ denotes the l_1-norm of the vector $\mathbf{x}$, and β is an appropriately chosen constant, perform well when employed for finding sparse solutions (see, e.g., [21], [22], [35]). Only recently there have been theoretical results justifying the performance of the constrained l_1 minimization. These results show that, for measurement matrices A which satisfy certain conditions, the constrained l_1 minimization recovers the solution if the unknown vector $\mathbf{x}$ is sparse enough, i.e., if the ratio k/n is sufficiently small

[25], [28]. The aforementioned conditions require that all subsets of the columns of A be sufficiently non-singular. Verifying such conditions is quite difficult and, thus, finding k which guarantees perfect reconstruction is rather hard. Recently, there has been a lot of effort on finding k which asymptotically (for large n) guarantees that the solution will be found with high probability for a large class of random matrices A [28]. In particular, there exist such results for matrices A with random Gaussian entries, as well as those with distributions that are symmetric around zero. We should also mention an ongoing search for recovery methods in the presence of noise [33]-[37]. These are of particular importance in practical scenarios, including compressed microarrays presented in this paper.

A note about complexity: A straightforward solution to l_1 minimization problem is obtained by linear programming, which can be solved in polynomial time (often $O(n^3)$, where n denotes the number of unknowns). For applications with very large n, cubic complexity may present practical difficulties. Therefore, there is a need for more efficient algorithms and an ongoing effort in signal processing community to find them [29]-[32]. As a general comment, it appears reasonable to expect that measurement matrices with special structure (e.g., sparse A) may lead to faster algorithms (more on this in Section 4).

3 Compressed microarrays

When quantifying a sparse signal, compressive sampling provides cost-efficient utilization of the sensing resources. In particular, we recall from Section 2.2 that a sparse signal may be recovered from a small number of linear combinations of its components. The compressive sampling ideas are relevant to the applications of DNA microarrays in gene expression profiling, where the gene expression levels of a test sample are compared with the gene expression levels of a reference sample. Since in practical scenarios only a small fraction of the total number of genes is differentially expressed, the difference of the signals produced by the two samples is sparse. Moreover, linear combinations of the signal components may be acquired by the composite probe spots comprising a mixture of several probe sequences as in [19]. The sparseness constraint, on the other hand, suggests possible recovery of the signal from potentially far fewer probe spots than the total number of probe sequences composing the spots of the microarray.

In [39], the authors developed a statistical model for microarrays, which is directly applicable to the compressed microarrays. In particular, for a compressed microarray with n spots containing probes designed to quantify m different targets, we can write

$$\mathbf{y} = A\mathbf{x} + \mathbf{w} + \mathbf{v}, \tag{5}$$

where $\mathbf{y}$ denotes the n-dimensional measurement, $\mathbf{x}$ denotes the m-dimensional data vector (the number of copies of each target), $\mathbf{v}$ is the n-dimensional zero-mean iid Gaussian additive noise due to instrumentation and other biochemistry-independent noise sources, $\mathbf{w}$ denotes the shot-noise (i.e., zero-mean iid Gaussian noise with covariance proportional to the signal – see, e.g., [39]-[40]), and where A is an $n \times m$ binary matrix containing information about probe mixing. In other words, the (i, j) element of A is non-zero if and only if the jth target can bind to some of the probes in the ith spot. We limit the entries in A to binary $1/0$ for the sake of manufacturing simplicity, e.g., to impose the constraint that each microarray spot contains an equal amount of different probes comprising it. Each row of the matrix A corresponds to a probe spot. The composition of the ith probe spot, $1 \leq i \leq m$, is determined by the positions of ones in the ith row of A. Moreover, the number of different probes in the ith spot is equal to the number of ones in the ith row of the matrix A. An illustration of the compressed microarray system model is shown in Figure 1.

In a two-color microarray experiment, we are comparing two samples characterized by data vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, and are interested in finding differentially expressed genes, i.e., finding non-zero entries of the vector

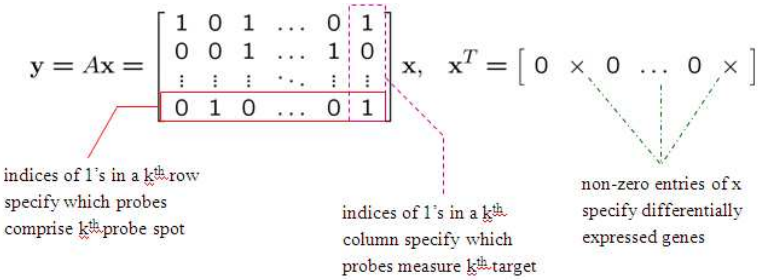
y = Ax =
1 0 1 ... 0 1
0 0 1 ... 1 0
0 1 0 ... 0 1
x, x^T = [0 × 0 ... 0 ×]
indices of 1's in a kth row specify which probes comprise kth probe spot
indices of 1's in a kth column specify which probes measure kth target
non-zero entries of x specify differentially expressed genes

Figure 1: *An illustration of the compressed microarray system model.*

$x = \mathbf{x}_1 - \mathbf{x}_2$. Defining $y = \mathbf{y}_1 - \mathbf{y}_2$, $w = \mathbf{w}_1 - \mathbf{w}_2$, and $v = \mathbf{v}_1 - \mathbf{v}_2$, we can write

$$y = Ax + w + v. \tag{6}$$

The vector x in (6) is sparse, i.e., it has a small number of entries that are non-zero (or significantly larger than zero). Recalling the discussion of compressive sampling, it should appear clear that since x is sparse, one may be able to recover it using (3) or (4).

We should briefly mention the important issue of probe design. Two among the most important properties of microarray probes are their sensitivity and specificity. Sensitivity is a measure of how strongly a probe reacts with the target which it is supposed to capture. Specificity, on the other hand, is the ability of a probe to discriminate between targets, i.e., its ability to ignore (do not bind or cross-hybridize to) other targets. In (6), we have implicitly assumed that all probes are equally sensitive and that there is no probe-target binding due to cross-hybridization. The scenario wherein these assumptions do not hold and techniques which take that into account are considered in [39]. Imbalanced sensitivity, for instance, may be incorporated in the compressed microarray model by appropriately scaling selected non-zero entries of A. Imperfect specificity, on the other hand, would require increasing the fraction of non-zero entries in A. In general, cross-hybridization is detrimental to the complexity of the signal recovery in compressed microarrays and thus special attention should be payed to specificity of probes in compressed microarrays.

As an illustration, in Figure 2 we demonstrate the performance of l_1-constrained minimization employed for the detection of sparse signals in a compressed microarray simulated according to the model (6). The microarray comprises $n = 24$ probe spots, and each spot contains a mixture of 24 different probes chosen from the set of $m = 96$ available probe sequences, each designed to capture one target of interest. So, the dimension of the matrix A is 24×96. Moreover, the number of non-zero entries in x is $k = 8$. Parameters of the microarray model (6) are chosen so as to mimic a realistic experiment. As implied by Figure 2, the algorithm successfully recovers sparse data from noisy observations.

For the same system, we compare the relative mean-square-error of the signal vector estimate obtained via l_1-optimization in the compressed microarray with the mean-square-error of the direct readout in the conventional (i.e., not compressed microarray). The obtained relative mean-square-error is shown as a function of signal-to-noise ratio (defined as the ratio of the expected signal power and the expected additive noise power) in Figure 3. The mean-square-error is computed over 1000 Monte Carlo runs, where we randomly generated signal and noise. As expected, the reduction (by a factor of 4) of the number of probe spots in a compressed microarray is penalized by a slightly higher estimation mean-square error.

When designing compressed microarrays, we have the freedom in choosing the coefficient matrix A, i.e., choosing the mixtures which compose the probe spots. A major design challenge is the construction of matrices A which facilitate the exact recovery of the sparse signal in the absence of noise, and robust performance of the recovery algorithms in a noisy case. A large fraction of the compressive sampling literature considers a random choice of A. For example, there are results showing that a large class of random matrices satisfy the reconstruction requirements mentioned in Section 2.2 with high probability. This essentially means that, for a random choice of A, as the dimension of the problem n increases, the probability of finding the correct solution approaches one. We should note that these results require that the signal be ”sparse enough”.

Probabilistic results have somewhat limited relevance to the design of compressed microarrays. Costly synthesis of probes and manufacturing of arrays call for non-random constructions of A and efficient algorithms which would *deterministically* guarantee exact signal recovery in a noiseless scenario, and be robust with respect to noise. Moreover, to maintain complexity and cost of array manufacturing low, the number of different probes composing the spots should not be too large. This translates to a constraint on the number of ones in each row of the coefficient matrix A. Therefore, signal recovery methods that are inspired by low-density parity check (LDPC) codes and expander graphs [44]-[51] appear promising in addressing the above issues. For instance, an LDPC code generator matrix is a well suited choice for the coefficient matrix A in compressed microarray applications: it has fixed and low number of ones in each row and column, and is generally non-singular which is one of the requirements for signal recovery capability.

Finally, since we have the freedom of choosing the coefficient matrix A, and since A is sparse itself, it is reasonable to expect that we can find faster and more robust algorithms, e.g., faster than linear programming. Fast algorithms are of particular importance in the compressed microarray applications, where the size of the unknown vector x may be rather large (up to tens of thousands, i.e., the number of genes).

3.1 Compressed microarrays for aCGH technologies

Array comparative genomic hybridization (aCGH) technology, wherein total genomic DNA from a test and a reference samples are compared on a DNA microarray, has recently emerged as a platform for detecting and mapping alterations in genomic structures. The arrays used in aCGH applications may have as many as million of distinct probes and are thus capable of providing high-resolution information about the DNA sequence in focus. The aforementioned alterations (i.e., changes in DNA copy number) include small amplifications and deletions, as well as large chromosomal gains and losses [42]. Such alterations are commonly encountered in various types of human cancers, including breast and ovarian (see, e.g., [42] and the references therein). Therefore, detecting DNA alterations can provide valuable information about the genomic mechanisms of cancer as well as be used in diagnostics and search for drug treatments.

The alterations typically affect continuous segments of a genome, and their total length is just a fraction of the length of entire genome. By mapping DNA segments to their chromosomal locations and reordering x accordingly, the unknown signal is both sparse and piecewise constant (see, e.g., [43] and the references therein). Sparse signal with piecewise constant segments can be efficiently recovered using l_1 minimization. For instance, the Lasso optimization problem (4) can be modified to

$$\min \|y - Ax\|_2 \ \textit{subject to} \ \|x\|_1 \leq \beta, \ \|Dx\|_1 \leq \gamma, \tag{7}$$

where β and γ are appropriately chosen constants, and where the matrix D denotes the differentiation operator, i.e., its general structure is given by

$$D = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ & & \ddots & \ddots & \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

[We should note that other constraints on the signal, such as piecewise linear, can be taken into account in a way similar to (7).]

In Figure 4, we illustrate the performance of the algorithm (7) when applied for detection of sparse, piecewise-constant signals in compressed microarrays simulated according to the model (6). The microarray comprises $n = 24$ probe spots, and each spot contains a mixture of 24 different probes chosen from the set of $m = 96$ available probe sequences, each designed to capture one target of interest. Hence, the dimension of the matrix A is 24×96. Moreover, the number of the non-zero entries in x is $k = 28$. The simulation parameters of the model (6) are chosen so as to mimic realistic experiment. As implied by Figure 4, (7) successfully recovers sparse data from noisy observations. Note that the fraction of non-zero components of x is fairly large ($k = 28$ out of $m = 96$). In fact, without the additional piecewise-constant constraint, l_1 minimization is not able to recover the signal since it is not sufficiently sparse. However, since the derivative of a piecewise-constant signal is very sparse, the signal can be efficiently recovered using (7).

4 Speeding up the sparse signal recovery in applications with sparse coefficient matrices

When the coefficient matrix A is sparse, as in the compressed microarray applications, the sparse signal recovery may be performed more efficiently than in the cases where A has a general structure. To this end, we start off by making several observations below.

Let us consider the noiseless case and y_i, the ith component of the observation vector y. It is obtained as an inner product of the ith row of A with the vector x,

$$y_i = \sum_{k=1}^{n} a_{ik} x_k, \tag{8}$$

where a_{ik} denotes the (i, k) entry of A. The sparseness of both A and x implies that y_i may be zero for some i; clearly, the chance of this happening increases with the sparseness of A and x since, as their sparseness increases, it becomes more likely that, for a given i, we cannot find k such that both $a_{ik} \neq 0$ and $x_k \neq 0$.

On the other hand, in the compressed microarray applications A comprises zeros and ones while the non-zero entries of x are real numbers. Therefore, if $a_{ik} x_k \neq 0$ for any k, it is highly unlikely that y_i in (8) is zero. Let $\mathcal{K}_i$ denote the set of indices k, $1 \leq k \leq n$, such that $a_{ik} \neq 0$. If $y_i = 0$, we may conclude that, with high probability, $x_k = 0$ for all $k \in \mathcal{K}_i$.

Similarly, if two or more entries in the observation vector y are equal and non-zero, with high probability it is so because they measure the same non-zero components of x. For instance, if $y_i = y_j \neq 0$, they are equal because not all x_k, $k \in \mathcal{K}_i \cap \mathcal{K}_j$, are zero. More importantly, $y_i = y_j \neq 0$ also means that all x_k, $k \in (\mathcal{K}_i \cup \mathcal{K}_j) \setminus (\mathcal{K}_i \cap \mathcal{K}_j)$, *are* zero. In other words, if $y_i = y_j \neq 0$, then $x_k = 0$ for every k such that $a_{ik} \neq a_{jk}$. Similar statements can be made if more than two components of the observation vector $\mathbf{y}$ are non-zero and equal.

Using the observations above, we can recover many of the components of x and often all of them. If all of the components are not found, one can attempt to find the rest via the constrained l_1 optimization problem (3). The advantage now is that, due to the removal of many unknowns and equations, the computational complexity of this step is significantly reduced – see comments below.

The procedure described above can be formalized with the following algorithm.

Input: y, A.

1. Form subsets collecting indices of the components of y of equal value. Denote by $\mathcal{S}$ a subset among them with the largest cardinality (if there exist more than one such subset, choose any). Denote by y_s the value of the components of y with an index from $\mathcal{S}$.

2. If $|\mathcal{S}| = 1$, terminate the algorithm.

3. For each $s_i \in \mathcal{S}$, $1 \leq i \leq |\mathcal{S}|$, find $\mathcal{K}_i = \{k : a_{s_i k} \neq 0\}$.

4. Find subsets $\mathcal{P}$ and $\mathcal{Q}$ defined as $\mathcal{P} = \mathcal{K}_1 \cap \mathcal{K}_2 \cap \cdots \cap \mathcal{K}_{s_i}$, $\mathcal{Q} = (\mathcal{K}_1 \cup \mathcal{K}_2 \cup \cdots \cup \mathcal{K}_{s_i}) \setminus \mathcal{P}$.

5. If $|\mathcal{P}| = 1$, then

5.1. Set $\hat{x}_p = y_s$, where p is the sole component of $\mathcal{P}$.

5.2. $y = y - y_s A_p$, where A_p denotes the p^{th} column of A.

5.3. Remove the components of y having indices in $\mathcal{S}$, and the corresponding rows of A.

5.4. Remove the columns of A having indices in $\mathcal{P} \cup \mathcal{Q}$.

6. If $|\mathcal{P}| > 1$, then

6.1. Remove all but one component of y with indices in $\mathcal{S}$, and the corresponding rows of A.

6.2. Remove all columns of A with indices in $\mathcal{Q}$.

7. Go to step 1.

We will refer to the above as the *sparse matrix pre-processing* (SMPP) algorithm. Based on the previously made observations, the SMPP algorithm exploits sparseness of the coefficient matrix A to compute as many components of x as possible. In this process, some rows and columns of A become redundant. Hence, in step 5 and step 6, the SMPP algorithm updates A by removing the redundant columns and rows. In doing so, the algorithm may fully recover the sparse signal x. If not, it will terminate after recovering a number of components of x and removing the corresponding rows and columns of A. The remaining components of x can be recovered with a linear program, solving a reduced size l_1 minimization problem whose constraints involve the reduced coefficient matrix A. We will refer to the combination of the SMPP and linear program as the SMPP-LP algorithm. Whether the SMPP algorithm will be sufficient to recover x, or SMPP-LP need to be used, depends upon the sparseness and particular structure of A.

The computational complexity of linear programming, often $O(n^3)$ where n is the size of the problem, may be prohibitive for high-dimensional problems. On the other hand, the complexity of the pre-processing described in this section is only linear in n. Therefore, the pre-processing algorithm, which may significantly reduce the size of the problem that needs to be solved with linear program, extends practical feasibility of sparse recovery to large problems such as those encountered in microarray applications.

Figure 5 shows a comparison of the CPU time of SMPP-LP and linear program. The graph shows the CPU time, averaged over $N = 1000$ Monte Carlo runs, required for solving a sparse recovery problem $y = Ax$ where the $n \times m$ matrix A and the k-sparse, $n \times 1$ vector x are generated randomly. The matrix A is sparse with six 1's per column, and the ratios $m/n = 0.25$, $k/n = 0.1$.

On another note, the SMPP (or SMPP-LP) algorithm may relax the requirement on the sparseness of the signal. For instance, when the coefficient matrix is generated randomly from a symmetric distribution, constrained l_1 minimization may recover signals with k/n at most 0.17. Empirical studies of the SMPP-LP algorithm for sparse signal recovery in applications where A is sparse imply that the ratio k/n may be as large as 0.25. In Figure 6, we compare the performance of the SMPP-LP algorithm with the linear program. The size of x is $n = 400$, and the largest k that can be recovered with either algorithm is plotted as a function of the number of measurements m. Clearly, linear program requires more strict sparseness constraints than the SMPP-LP algorithm. The benefit of the latter is particularly significant for small m/n.

As mentioned earlier, low-density parity check (LDPC) code generator matrices are well suited for the compressed microarray applications: they are very sparse, with constant number of ones in each row and column. When they are chosen as coefficient matrices, and the SMPP-LP algorithm is used for recovery, the signal need not be very sparse. We have tested several rate-$1/2$ LDPC codes, publicly available from [52]. In particular, we used the LDPC codes as coefficient matrices A of dimensions 408×816, 504×1008, 2000×4000, and 4000×8000. For all of the above, the SMPP-LP algorithm was able to recover randomly generated signal vectors with sparseness $k \approx m/2 = n/4$, i.e., it could recover up to $k/n = 0.25$.

We recall that the discussion in this section concerns the sparse signal recovery in the noiseless case. In the noisy case, we use steps $1 - 6$ of the SMPP algorithm.

5 Experimental verification

In this section, we present a series of proof-of concept experiments designed and conducted to demonstrate data acquisition and signal recovery in compressed microarrays. The experiments were conducted in the Millard and Muriel Jacobs Genetics and Genomics Laboratory at California Institute of Technology. The goal was detection and quantification of $k \leq 8$ targets on an array otherwise capable of testing $n = 96$ different targets. Moreover, the desired probe spot compression ratio, m/n was chosen to be 4. Therefore, the compressed microarray has only $m = 24$ probe spots, each comprising a combination of a number of different probe sequences. Mixtures of the probes, synthesized oligonucleotide sequences, were deposited to their respective spots; the targets are cDNA molecules extracted from *Escherichia Coli*. Details are given below.

5.1 Description of the experiments

Targets for the compressed microarray experiment were generated using The RNA Spikes$^{\text{TM}}$, a commercially available set of 8 purified RNA transcripts purchased from Ambion Inc. Lengths of the RNA sequences are $(750, 752, 1000, 1000, 1034, 1250, 1475, 2000)$ nucleotides, respectively. Typically, these spikes are used in microarrays for calibration purposes and have been chosen so that the eight sequences have little mutual correlation. The RNA sequences were reverse transcribed to obtain cDNA targets, which were then labeled with Cy5 dyes. We denote the set of these 8 targets by $\mathcal{T}_8$.

Eight oligo probes designed for capturing the targets in $\mathcal{T}_8$ were also purchased from Ambion Inc. Moreover, we acquired 88 probes designed to test the mouse genome. We denote the set of Ambion probes as $\mathcal{P}_8$, and the set of mouse genome probes as $\mathcal{P}_{88}$. The full set of 96 oligonucleotide probes, all of them 25 nucleotides long, is denoted as $\mathcal{P}_{96}$. The targets from $\mathcal{T}_8$ do not cross-hybridize with (i.e., bind to) the probes from $\mathcal{P}_{88}$.

We designed $m = 24$ different mixtures, each comprising 24 probes selected from $\mathcal{P}_{96}$. Each of the mixtures is deposited in one of the spots of the compressed microarray. Content of the mixtures determine composition of the coefficient matrix A; hence, each row in A has 24 ones and 72 zeros. The structure of A we used in the experiments is illustrated in Figure 7 (we omit the full matrix for brevity).

5.2 Experimental results

For all of the experiments described in this subsection, the sparse signal vector x was constructed such that $x_k \neq 0$ if and only if $k \in \mathcal{K} = \{1, 9, 17, 25, 33, 41, 49, 57\}$. In particular, x_1 contains information about the amount of the first target from the set $\mathcal{T}_8$, x_9 contains information about the amount of the second target from $\mathcal{T}_8$, etc.

In the first experiment, the targets from $\mathcal{T}_8$ were applied to a microarray, where the individual amounts of targets were (5ng, 5ng, 2ng, 1ng, 10ng, 2ng, 1ng, 1ng), respectively. The experiment was left overnight and the array, after washing away the sample, was scanned. Figure 8 shows (a) the measured light intensities of the compressed microarray spots, and (b) the recovered signal. Clearly, the strongest 8 components of the recovered signal correspond to the targets in $\mathcal{T}_8$.

In the second experiment, the targets from $\mathcal{T}_8$ were applied to a microarray, where the individual amounts of targets were (5ng, 5ng, 2ng, 1ng, 10ng, 2ng, 1ng, 1ng), respectively. Similar to the above, the experiment was left overnight and the array, after washing away the sample, was scanned. Figure 9 shows (a) the measured light intensities of the compressed microarray spots, and (b) the recovered signal. Once again, the strongest 8 components of the recovered signal correspond to the targets in $\mathcal{T}_8$.

These early experimental results indicate practical feasibility of the techniques proposed in the paper. The reduction in the number of probe spots is rather significant: the compressed microarrays we designed and tested have 4 times fewer spots than what their conventional counterparts would need to perform the same task. The recovered signal was noisy yet its strongest components correctly identified targets from the test sample.

We conducted several more compressed microarray experiments testing the targets from $\mathcal{T}_8$, including the experiments where we added complex biological background (i.e., total mice DNA) to the sample; in these experiments, the strongest components of the recovered signal vector correctly identified targets from $\mathcal{T}_8$ and thus the compressed microarray appears capable of detecting them in the presence of biological background presence. As a part of the future work, we intend to calibrate the array (i.e., determine the strengths of the binding of the targets from $\mathcal{T}_8$ to their corresponding probes) in order to enable precise quantification of their amounts.

6 Summary and conclusions

In this paper, we discussed a novel DNA microarray architecture which we refer to as compressed DNA microarrays. In compressed microarrays, each probe spot contains a mixture of a number of different probes. By exploiting inherent sparseness of the signals in gene expression studies, target detection and quantification can be performed with an array that has significantly smaller number of spots than the number of probes consisting them. To this end, we used ideas from compressive sampling, and employed linear programming to solve an appropriate l_1-minimization problem. As we have demonstrated in simulations as well as with experiments, if the signal vector is sufficiently sparse, l_1-minimization can recover it.

In addition, we proposed the use of compressed microarray platforms in array comparative genomic hybridization (aCGH) technologies. There, we demonstrated that the signal can be recovered by performing l_1-minimization with an additional constraint.

Practical limitations impose certain requirements on the design of compressed microarrays. This is reflected by the so-called measurement matrix being sparse and comprising $1/0$ entries. For such a measurement matrix, efficiency of l_1-minimization can be significantly improved. To this end, we proposed an algorithm for pre-processing the coefficient matrix and, in the process, determining a fraction of (if not the full) signal vector. The algorithm reduces the size of (or completely eliminates need for) linear program, and can recover signals with higher signal content than linear programming which requires more sparse signal. We demonstrated the benefits of the proposed algorithm for both random as well as LDPC code inspired coefficient matrices. The algorithm was developed with the noiseless scenario in mind; it is of interest to search for its robust variants.

There are many directions where the work presented in the current paper can be extended. Random matrices, often considered in the compressive sampling literature, have limited relevance to the compressed microarray systems. Therefore, we need to find deterministic coefficient matrices that are sparse and have the properties required for signal recovery. To this end, we have taken some steps in this direction by studying the use of LDPC codes. It is of interest to extend this study, and expand it by looking into, e.g., expander graphs [51], etc.

7 Acknowledgements

The authors would like to thank Dr. Jose Luis Reichmann and Vijaya Rao of the Millard and Muriel Jacobs Genetics and Genomics Laboratory at California Institute of Technology for their help with the experiments presented in the paper.

References

[1] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, ”Quantitative monitoring of gene expression patterns with a complementary DNA microarray,” *Science*, 270(5235), October 1995, pp. 467-70.

[2] M. Schena, D. Shalon, R. Heller, A. Chai, P. O. Brown, and R. W. Davis, ”Parallel human genome analysis: microarray-based expression monitoring of 1000 genes,” *Proceedings of the National Academy of Sciences (PNAS)*, 93(20), October 1996, pp. 10614-9.

[3] D. Shalon, S. J. Smith, and P. O. Brown, ”A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization,” *Gen. Research*, 6(7), July 1996, pp. 639-45.

[4] J. DeRisi et. al., ”Use of a cDNA microarray to analyse gene expression patterns in human cancer,” *Nature Genetics*, 14(4), December 1996, pp. 457-60.

[5] A. P. Blanchard, R. J. Kaiser, and L. E. Hood, ”High-Density Oligonucleotide Arrays,” *Biosensors & Bioelectronics*, 1996, 11:687-690.

[6] A. P. Blanchard, and L. E. Hood, ”Sequence to array: probing the genome’s secrets,” *Nature Biotechnology*, 1996, 14:1649.

[7] M. Schena, *Microarray Analysis*, John Wiley & Sons, 2003.

[8] U. R. Mueller and D.V. Nicolau (Eds.), *Microarray Technology and Its Applications*, Springer, 2005.

[9] M. Schena et. al., ”Microarrays: biotechnology’s discovery platform for functional genomics,” *Trends in Biotechnology* 1998, 16, 301-306.

[10] D. D. Shoemaker et. al., ”Experimental annotation of the human genome using microarray technology,” *Nature*, 409(6822), 2001, pp. 922-927.

[11] W. Zhang and I. Shmulevich (Eds.), *Computational and Statistical Approaches to Genomics*, Kluwer Academic Publishers, 2002.

[12] J. Kononen et. al., ”Tissue microarrays for high-throughput molecular profiling of tumor specimens,” *Nature Medicine*, 4(7), July 1998, pp. 844-847.

[13] M. J. Marton et. al., ”Drug target validation and identification of secondary drug target effects using DNA microarrays,” *Nature Medicine*, 4(11), Novemebr 1998, pp. 1293-301.

[14] J. Khan et. al., ”Expression profiling in cancer using cDNA microarrays,” *Electrophoresis*, 20(2), February 1999, pp. 223-9.

[15] C. A. Afshari, E. F. Nuwaysir, and J. C. Barrett, ”Application of complementary DNA microarray technology to carcinogen identification, toxicology, and drug safety evaluation,” *Cancer Research*, 59(19), October 1999, pp. 4759-60.

[16] U. Scherf et. al., ”A gene expression database for the molecular pharmacology of cancer,” *Nature Genetics*, 24(3), March 2000, pp. 236-44.

[17] J. Marx, ”DNA arrays reveal cancer in its many forms,” *Science*, September 2000, 289: 1670-1672.

[18] D. T. Ross et. al., ”Systematic variation in gene expression patterns in human cancer cell lines,” *Nature Genetics*, 24(3), March 2000, pp. 227-35.

[19] I. Shmulevich, J. Astola, D. Cogdell, S. R. Hamilton, and W. Zhang, ”Data extraction from composite oligonucleotide microarrays,” *Nucleic Acids Research*, vol. 31, no. 7, 2003.

[20] R. Tibshirani, ”Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society, Series B*, 58(1):267-288, 1996.

[21] S. S. Chen, *Basis Pursuit*, PhD Thesis, Stanford University, 1995.

[22] S. S. Chen, D. L. Donoho, and M. A. Saunders, ”Atomic decomposition by basis pursuit,” *SIAM Journal on Scientific Computing*, vol. 20, no. 1, 1999, pp. 33-61.

[23] E. Candes, J. Romberg, and T. Tao, ”Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. on Info. Theory*, 52(2), Feb. 2006, pp. 489-509.

[24] E. Candes and T. Tao, ”Near optimal signal recovery from random projections: Universal encoding strategies,” *IEEE Transactions on Information Theory*, 52(12), December 2006, pp. 5406-5425.

[25] D. Donoho, ”Compressed sensing,” *IEEE Trans. on Info. Theory*, 52(4), April 2006, pp. 1289-1306.

[26] D. Donoho and Y. Tsaig, ”Extensions of compressed sensing,” *Sig. Proc.*, 86(3), 2006, pp. 533-548.

[27] E. J. Candes, ”Compressive sampling,” *Proc. of the Intern. Congress of Mathem.*, Madrid, Spain, 2006.

[28] E. Candes and T. Tao, ”Decoding by linear programming,” *IEEE Transactions on Information Theory*, 51(12), December 2005, pp. 4203-4215.

[29] J. Tropp and A. Gilbert, ”Signal recovery from partial information via orthogonal matching pursuit,” to appear in *IEEE Transactions on Information Theory* (preprint available from http://www.acm.caltech.edu/ jtropp/papers/TG07-Signal-Recovery-preprint.pdf).

[30] G. Cormode and S. Muthukrishnan, ”Towards an algorithmic theory of compressed sensing,” *Technical report DIMACS TR 2005-25*, 2005.

[31] G. Cormode and S. Muthukrishnan, ”Combinatorial algorithms for compressed sensing,” *Technical report DIMACS TR 2005-40*, 2005.

[32] A. Gilbert, M. Strauss, J. Tropp, and R. Vershynin, ”Sublinear, small-space approximation of compressible signals and uniform algorithmic embeddings,” in *Proc. SPIE Intelligent Integrated Microsystems*, pp. 623206.01-09, Orlando, Apr. 2006.

[33] E. Candes, J. Romberg, and T. Tao, ”Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, 59(8), August 2006, pp. 1207-1223.

[34] J. Haupt and R. Nowak, ”Signal reconstruction from noisy random projections,” *IEEE Transactions on Information Theory*, 52(9), September 2006, pp. 4036-4048.

[35] M. J. Wainwright, ”Sharp thresholds for high-dimensional and noisy recovery of sparsity,” *Technical report 709, Department of Statistics, UC Berkeley*, May 2006 (available from http://www.stat.berkeley.edu/tech-reports/709.pdf).

[36] E. Candes and J. Romberg, ”Quantitative robust uncertainty principles and optimally sparse decompositions,” *Foundations of Computational Mathematics*, 6(2), April 2006, pp. 227-254.

[37] E. J. Candes and J. Romberg, ”Practical signal recovery from random projections,” *Wavelet Applications in Signal and Image Processing XI, Proc. SPIE Conf. 5914*, 2004 (available from http://www.acm.caltech.edu/ emmanuel/papers/PracticalRecovery.pdf).

[38] M. Sheikh, S. Sarvotham, O. Milenkovic, and R. Baraniuk, ”DNA Array Decoding from Nonlinear Measurements by Belief Propagation,” in *Proc. of 14th Workshop on Statistical Signal Processing*, August 2007, pp. 215-219.

[39] H. Vikalo, A. Hassibi, and B. Hassibi, ”A statistical model for microarrays, optimal estimation algorithms, and limits of performance,” *IEEE Transactions on Signal Processing, Special Issue on Genomic Signal Processing*, vol. 54, no. 6, June 2006, pp. 2444 - 2455.

[40] Y. Tu, G. Stolovitzky, and U. Klein, “Quantitative noise analysis for gene expression microarray experiments,” *Proceedings of the National Academy of Sciences (PNAS)*, October 29, 2002, 14031-14036.

[41] H. Vikalo, B. Hassibi, and A. Hassibi, ”Limits of performance of DNA microarrays,” *Proceedings of the IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006.

[42] D. Lipson, *Computational Aspects of DNA Copy Number Measurement*, PhD Tesis, Technion, 2007.

[43] R. Pique-Regi, E.-S. Tsau, A. Ortega, R. Seeger, and S. Asgharzadeh, ”Wavelet footprints and sparse Bayesian learning from DNA copy number change analysis,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007, pp. 353-356.

[44] S. Sarvotham, D. Baron, and R. Baraniuk, ”Sudocodes - fast measurement and reconstruction of sparse signals,” *Proceedings of the IEEE Symposium on Information Theory (ISIT)*, Seattle, WA, July 2006.

[45] G. A. Margulis, ”Explicit group-theoretic construction of combinatorial schemes and their applications in the construction of expanders and concentrators,” *Problems in Info. Transm.*, 24(1), 1988, pp. 39-46.

[46] M. Sipser and D. Spielman, ”Expander codes,” *IEEE Trans. on Info. Theory*, 42(6), 1996, pp. 1710-22.

[47] G. A. Margulis, ”Explicit construction of expanders,” *Problems in Info. Transm.*, 9(1), 1973, pp. 84-87.

[48] D. Peleg and E. Upfal, ”Constructing disjoint paths on expander graphs,” *Combinatorica*, 9(3), 1989, pp. 289-313.

[49] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, ”Randomness conductors and constant degree expansion beyond the degree / 2 barrier,” *Proceedings of the* 34^{th} *STOC*, 2002, pp. 659-668.

[50] S. Hoory, N. Linial, and A. Wigderson, ”Expander graphs and their applications,” *Bulletin of American Mathematical Society*, 43(4), 2006, pp. 439-561.

[51] W. Xu and B. Hassibi, ”Efficient compressive sensing with deterministic guarantees using expander graphs”, *Proceedings of the IEEE Information Theory Workshop*, Lake Tahoe, September 2007.

[52] http://www.inference.phy.cam.ac.uk/mackay/CodesFiles.html

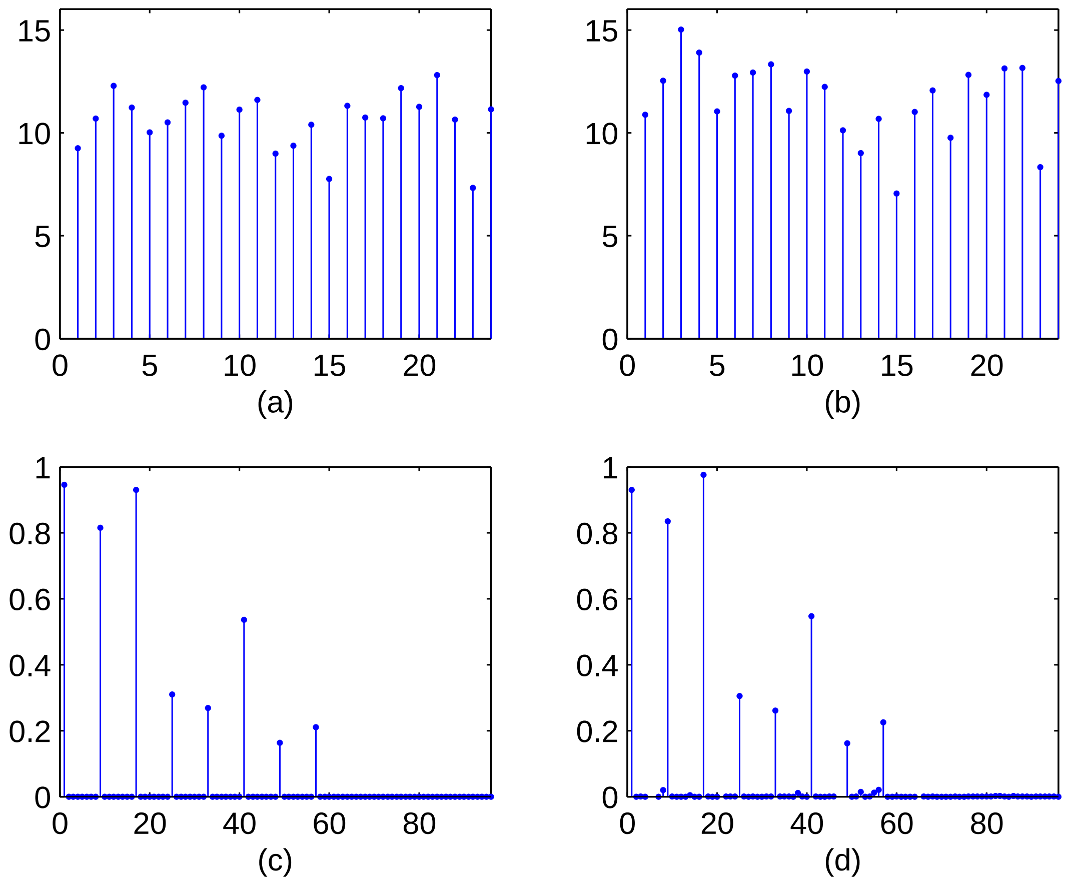

Figure 2: *Demonstration of the sparse signal recovery in a compressed microarray with* $n = 24$ *probe spots, where each spot comprises mixtures selected from the set of* $m = 96$ *probes. The number of non-zero entries in the* 96*-dimensional signal is* $k = 8$*. Subfigures (a) and (b) show the test and the reference signals, respectively, versus probe spot index. Subfigure (c) shows the sparse signal, and subfigure (d) its estimate obtained by solving an appropriate* l_1 *minimization problem.*

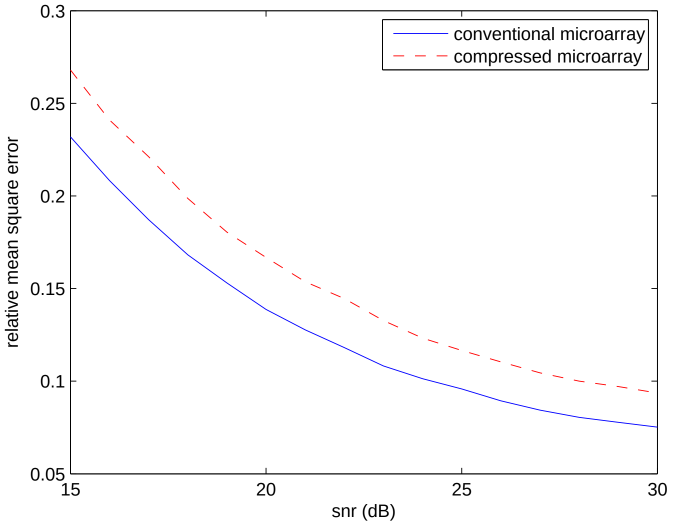

Figure 3: *Comparison of the mean-square error of the estimate of x via l_1-optimization employed for sparse reconstruction in a compressed microarray with the mean-square error of the direct readout in a corresponding conventional microarray. The parameters used in the simulations are $n = 96$, $m = 24$, and $k = 8$. The mean-square error is computed over 1000 Monte Carlo runs.*

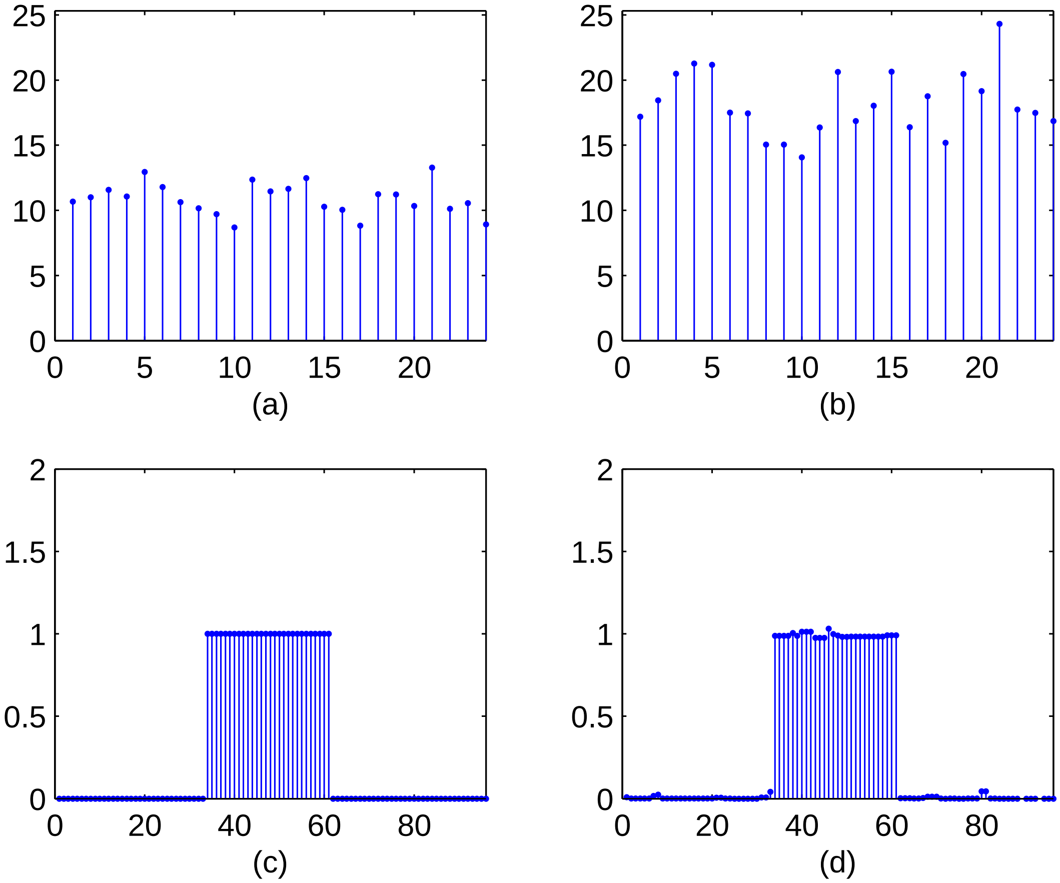

Figure 4: *Illustration of the sparse signal recovery in a compressed microarray for aCGH applications. The toy example considers an array with* $n = 24$ *probe spots, where each spot comprises mixtures selected from the set of* $m = 96$ *probes. The number of non-zero entries in the* 96*-dimensional piecewise-constant signal is* $k = 28$*. Subfigures (a) and (b) show the test and the reference signals, respectively, versus probe spot index. Subfigure (c) shows the sparse signal, and subfigure (d) its estimate obtained by solving (7).*

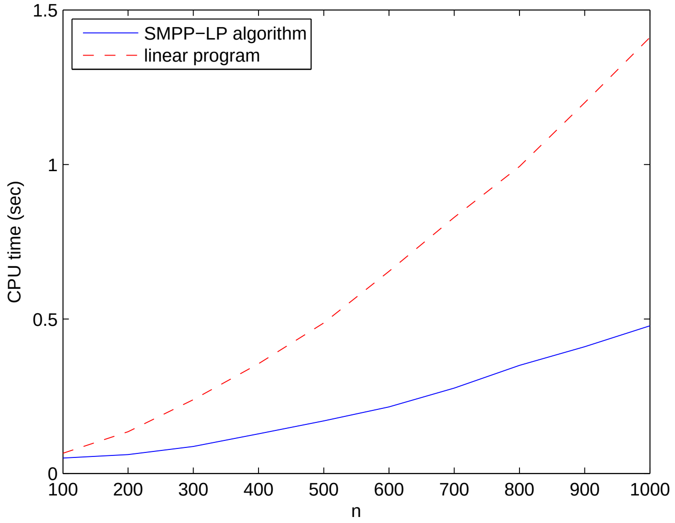

Figure 5: *Comparison of the CPU time of SMPP-LP and linear program. The graph shows the CPU time, averaged over* $N = 1000$ *Monte Carlo runs, required for solving a sparse recovery problem* $y = Ax$ *where the* $n \times m$ *matrix* A *and the* k*-sparse,* $n \times 1$ *vector* x *are generated randomly. The matrix* A *is sparse with six* 1*'s per column,* $m/n = 0.25$*,* $k/n = 0.1$.

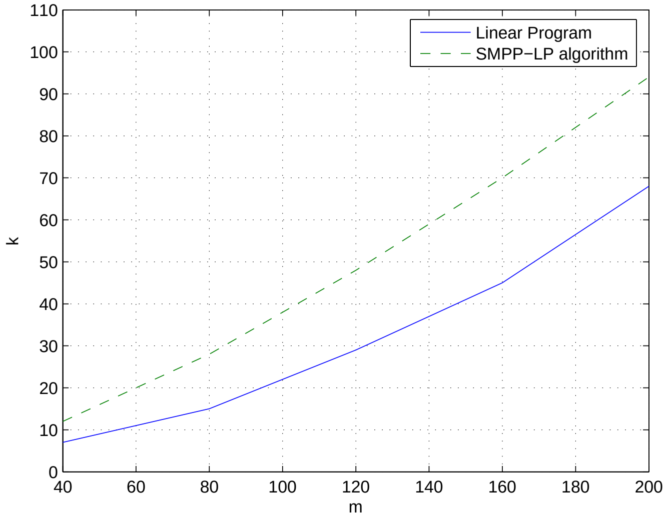

Figure 6: *Comparison of the limits of sparse signal recovery, the SMPP-LP algorithm vs. linear program. The graph shows the maximum number of non-zero elements of the signal vector,* k*, as a function of the number of measurements,* m*, whereas the size of the signal vector is* $n = 400$.

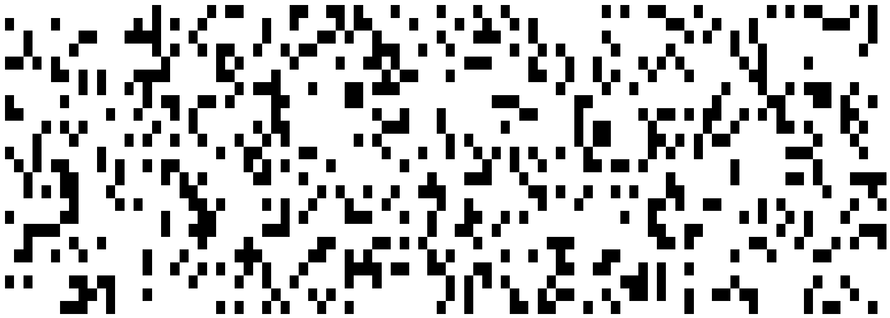

Figure 7: *Structure of the coefficient matrix A. Dark squares denote locations of ones, white denote locations of zeros.*

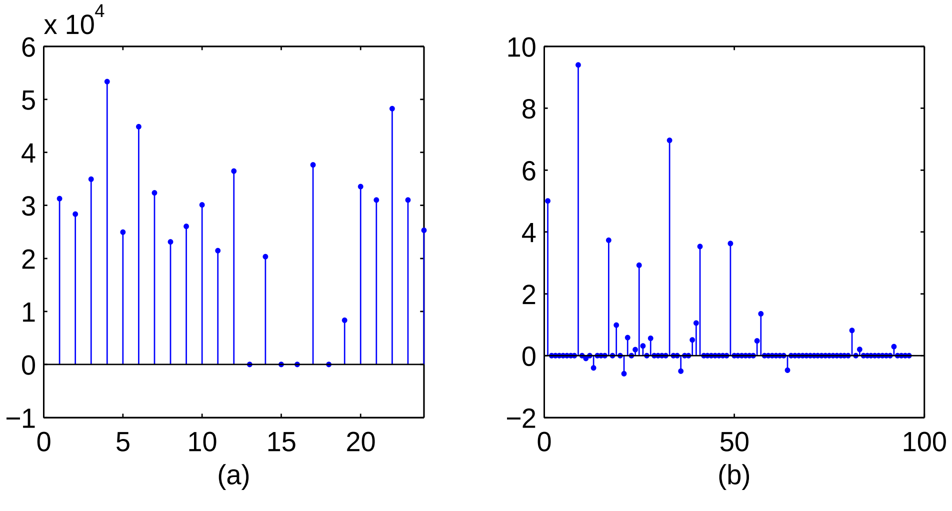

Figure 8: *Measured (a) and recovered (b) signal in an experiment where the targets from $\mathcal{T}_8$ were applied to an array. The amounts of the targets used were (5ng, 5ng, 2ng, 1ng, 10ng, 2ng, 1ng, 1ng), respectively. The strongest 8 components of the recovered signal in plot (b) correspond to the targets from $\mathcal{T}_8$.*

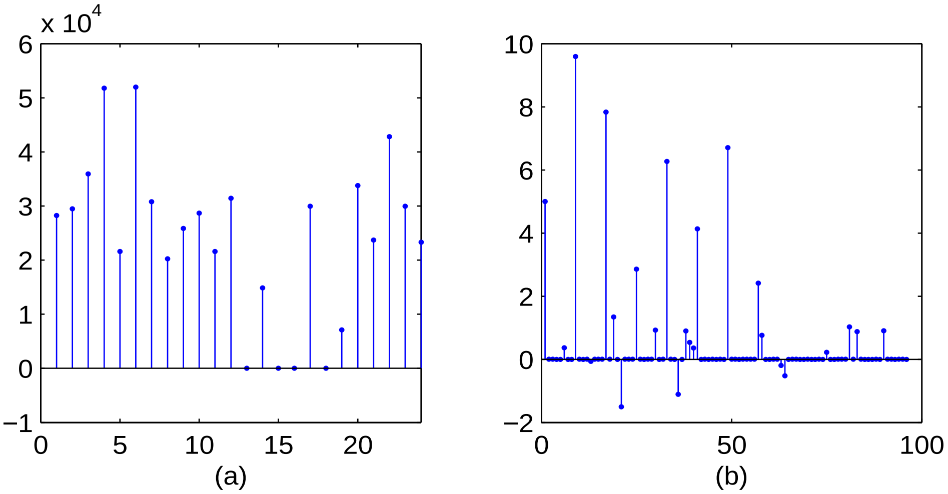

Figure 9: *Measured (a) and recovered (b) signal in an experiment where the targets from $\mathcal{T}_8$ were applied to an array. The amounts of the targets used were (5ng, 10ng, 5ng, 2ng, 5ng, 2ng, 2ng, 1ng), respectively. The strongest* 8 *components of the recovered signal in plot (b) correspond to the targets from $\mathcal{T}_8$.*