

The Distribution and Deposition Algorithm for Multiple Oligo Nucleotide Arrays

Kang Ning

ningkang@comp.nus.edu.sg

Hon Wai Leong

leonghw@comp.nus.edu.sg

Department of Computer Science, National University of Singapore, 3 Science Drive
2 117543, Singapore

Abstract

As the scale of the microarray experiments increases, a single oligo nucleotide array is no longer large enough. Therefore, the use of *multiple oligo arrays for one experiment* becomes more important. The design and synthesis of multiple arrays to minimize the overall synthesis cost is an interesting and important problem. We formulate the *multiple array synthesis problem (MASP)* that deals with the distribution of the probes (or oligos) to different arrays, and then deposition of the probes onto each array. We propose a cost function to capture the synthesis cost and a performance ratio for analysis of the quality of multiple arrays produced by different algorithms. We propose a **Distribution and Deposition Algorithm (DDA)** for the solving the MASP. In this algorithm, the probes are first distributed onto multiple arrays according to their characteristics such as GC contents. Then the probes on each arrays are deposited using a good deposition algorithm. Two other algorithms were also proposed and used for comparison. Experiments show that our algorithm can effectively output short synthesis sequences for multiple arrays, and the algorithm is efficient.

Keywords: multiple oligo nucleotide arrays, distribution, deposition

1 Introduction

DNA microarrays have recently become the method of choice for the monitoring of expression level of a large number of genes, especially for whole genomes [15]. Very briefly, DNA microarrays are composed of a set of distinct nucleic acid samples arranged in a regular lattice of spots on a solid support. In microarray experiments, fluorescent labeled DNA-derived samples will hybridize to the oligos in these DNA microarrays. This hybridization allow the monitoring of gene expressions or polymorphisms in genomic DNA [4]. Currently, there are two widely used formats of DNA microarrays: the *cDNA arrays* [13] and *synthesized oligo nucleotide arrays* [4, 8, 11]. In this paper, we focus on the synthesis of oligonucleotide arrays (or *oligo arrays* in short).

One technique for depositing the oligos onto the oligo array is the *photolithographic method* [3]. In the *photolithographic method* (used by Affymetrix[®]), a special *mask* is fabricated for each *synthesis cycle* of nucleotide addition. The mask permits light to penetrate only at positions where nucleotides are to be added. A *synthesis cycle* consists of shining light through the mask onto the chip surface and the positions where light passes through the mask and reaches the chip are the deprotected position which are activated for synthesis. After photo-deprotection, the chip is washed in a solution containing a *single nucleotide* that binds to oligos at the deprotected positions. The pattern in which light penetrates the masks directs the base-by-base synthesis of the oligos on the solid surface of the array. Photolithographic method has been used to produce arrays with as many as 40,000~65,000 DNA oligos with minimal cross-hybridization or inter-feature variability [4].

In each *synthesis cycle* only one nucleotide is deposited, and the sequence of nucleotides to be deposited in *successive* synthesis cycle is called the *synthesis sequence*. The *length* of the synthesis

sequence is the *number of synthesis cycles needed* for the synthesis process. Thus, it is important to reduce the length of the *synthesis sequences* since each of the synthesis steps is a *costly and error prone* manufacturing process.

We first describe the current design and synthesis of a *single oligo array* for an experiment - this is usually decomposed into two steps: *probe selection* and *oligo deposition*. *Probe selection* deals with the selection of the set of oligos (also called *probes*) representing the set of genes to be studied - the selected set of oligos are to be deposited into the array. The selection of the probes is important since it directly impacts the discriminative power of the microarray experiment. Some of the important factors for good probes include *probe length*, the *extend of complementarity* and the temperature [17]. There are usually 3~7 probes selected to represent each gene in the microarray experiment [17]. The problem of selecting an optimum set of probes for a set of genes is a complex problem. In this paper, we use the heuristic algorithms developed in [17] to generate a probes set that approximate the optimal set.

Once the set of probes (oligos) is selected, the next step is that of *oligo deposition* described earlier that deposits the oligos onto the oligo microarray using, for example, the *photolithographic method*. This problem has been modeled as the *shortest common supersequence (SCS)* problem that has been extensively studied in recent years. The problem is NP-hard and numerous heuristic algorithms have been proposed. In a recent paper, we proposed a new algorithm called the *LAP (Look-Ahead with PostProcessing)* and have carried out a comparative study of many algorithms for the SCS.

The paper deals with the design and synthesis of *multiple oligo arrays*, namely, microarrays that consist of several smaller microarrays. In multiple oligo arrays, the set of probes for an experiment is spread over several smaller microarrays, as opposed to an equivalent single larger microarray. We see the push for multiple oligo arrays emanating from (at least) two factors: (a) *larger scale microarray experiments* involving larger numbers number of genes and consequently, much larger number of probes, and (b) the greater cost-effectiveness and efficiency of manufacturing *several smaller microarrays*, as opposed to an equivalent single larger microarray.

As the number of genes in an experiment increases, a single oligo array is not large enough to perform the experiments which require, say, a few million oligos. Even though arrays with one million of oligos are already in use, these are expensive and are not yet for production use. The next generation of large scale experiments may require one hundred million oligos [6]. Therefore, it will be inevitable that multiple oligo arrays are necessary for these large scale experiments [4, 15].

The use of multiple arrays is not only the result of array size limitation. Using multiple arrays, each array will be comparatively smaller. The small arrays also have the advantage that the manufacture process is simpler, cheaper and faster, and the quality controls are also easier for the researchers carrying out the experiment. In reality, the multiple array systems are feasible, as manufacturing cost of microarrays continue to decrease [15]. We illustrate this with a small example shown in Figure 1. In this example, we assume that there are 12 genes and each gene is represented by one oligo (4-mers). We assume that the oligos have similar melting temperature and little complementarity. On the right, we use a single array for all 12 oligos and the shortest deposition sequence for *all* the 12 oligos is ACGACTACTGATG (of length 13). On the left, we use 4 arrays each with of 3 different oligos. A *different* deposition order can be applied on *each* of the sub-arrays. Thus, the four arrays have optimal deposition sequences of AGCAGTA, AGACGTAC, AATCGCATC and CGTCATG (lengths 7, 8, 9, 7), which are much shorter. For this example, the relative cost (defined later) using multiple array is $93 = (3 * 7 + 3 * 9 + 3 * 8 + 3 * 7)$, while that for a single large array is $156 = 12 * 13$.

There is an increasing number of researchers using multiple arrays in their experiments and we therefore believe that the use of multiple arrays will become increasingly widespread. Therefore, it is also important to design *efficient algorithms for the design and synthesis of multiple arrays to minimize the overall synthesis cost*. However, to the best of our knowledge, there has been no research devoted specifically to this problem.

In this paper, we propose a *cost function* that most accurately measures the relative synthesis cost

of each array and gives a better estimate the overall synthesis cost for multiple arrays. We also propose several algorithms for solving the *multiple array synthesis problem (MASP)*. Two algorithms, Greedy-MA and SH-MA, are based on generalizing algorithms for the single array problem, namely, the Greedy and Sum-Height, respectively. We also propose a new **Distribution and Deposition Algorithm (DDA)** for solving the MASP. This algorithm is an extension of our previous research on synthesis of probes on a single array [9]. The DDA algorithm first *distributes* the probes to the different arrays according to their characteristics such as *GC contents*. Then, we use SH [7] or LAP deposition algorithm [9] to generate the synthesis sequences for each of the individual arrays.

A-C-GT-	ACG--T-----
-GCA-T-	--G-C-A-T----
A-CA-A	AC-A--A-----
AGCAGTA	
AA--G--T-	A--A-----G-T-
--TCG-A--	-----T-C-GA--
A--C-C-C-	AC--C--C-----
AATCGCATC	
A-A-GT--	A--A-----G-T-
-GAC-T--	--GACT-----
A--C-AC	AC-AC-----
AGACGTAC	
C--CAT-	-C--C-A-T----
-GTC-T-	--G--T-CT----
--TCA-G	-----T-C-A-G
CGTCATG	ACGACTACTGATG

Figure 1: An example to illustrate the use of multiple arrays. There are 4 arrays containing 12 oligos. Their alignments in different arrays, as well as the synthesis sequences (in bold) are given.

2 Problem Definition and Algorithm

We shall assume that N is the number of probes in each array, M is the number of arrays, K is the length of each probe, and $q = |\Sigma|$ is the size of the alphabet ($q = 4$ for DNA sequences). There are MN probes in the probe set.

2.1 Problem Definition

We are given a set $S = \{s_1, s_2, \dots, s_{NM}\}$ of probes that has been selected from the probe selection step. Our aim is to produce a multiple array $MA = \{A_1, A_2, \dots, A_M\}$ that consists of M arrays each containing N probes and that together contains all the NM probes in the probe set S .

For each array A_i , we define $cost(A_i) = (L_i * N_i)$ to be the product of the number of deposition steps L_i and the number of probes N_i in the array A_i . This cost function reflects the relative cost of a mask of size N_i for array A_i for L_i synthesis steps. Then, the total cost for the multiple array $MA = \{A_1, A_2, \dots, A_M\}$ is given by $cost(MA) = \sum_{i=1}^M cost(A_i) = \sum_{i=1}^M L_i * N_i$.

We can now define the *multiple array synthesis problem (MASP)* as that of computing the multiple array $MA = \{A_1, A_2, \dots, A_M\}$ that minimizes $cost(MA)$. It is obvious that the MASP is NP-hard since it is a generalization of the SCS problem which is NP-hard [5]. In fact, the MASP with $M=1$ (a single array) is precisely the SCS problem.

2.2 Algorithms for Solving the MASP

We now present three algorithms for solving the multiple array synthesis problem - two algorithms based on different generalizations of existing algorithms for single array synthesis problem and one that is based on distribution, followed by deposition.

The Algorithm Greedy-A: The first algorithm, which we call *Greedy-A*, is based on a generalization of the Greedy algorithm for single array synthesis. The Greedy algorithm is based on probe-pair alignment. In Greedy, we first find the pair of probe (s_i, s_j) in S that gives the best alignment, namely, gives the shortest $SCS(s_i, s_j)$. We then replace the probes s_i and s_j by the merged-probe $SCS(s_i, s_j)$, and recursively apply the algorithm. The recursive scheme is described as follows: $\text{Greedy}(s_1, s_2, \dots, s_{NM}) = \text{Greedy}(SCS(s_1, s_2), s_3, \dots, s_{NM})$.

To generalize this to Greedy-A for multiple array synthesis, we compute $SCS(s_i, s_j)$ for every pair of probes. Then, the pairs with shortest SCS are grouped together. We then apply the procedure progressively. An array is found whenever there are N probes in a group G , and array is removed from the algorithm. The algorithm proceeds until there M arrays are found. Algorithm Greedy-A has time complexity $O(K^2 N^2 M^2)$ and space complexity $O(KNM + N^2 M^2)$. Unfortunately, it was shown [11] that Greedy-A does not necessarily output synthesis sequences shorter than periodical sequences.

The Algorithm Greedy-D: Another greedy algorithm, which we call *Greedy-D*, is based on the generalization of the Sum-Height (SH) [7] deposition algorithm. The Greedy-D algorithm deposits all probes in S simultaneously using SH deposition algorithm. When N probes are completed, it outputs these N probes as an array. Then the SH deposition algorithm is re-started on the remaining probes *de novo*. This process continues until M arrays have been computed and all probes are distributed to the M arrays. Algorithm Greedy-D has time complexity $O(KNM^2)$ and space complexity $O(KNM)$. The SH algorithm generally produces synthesis sequences that are better than those of the periodical sequences and so we expect Greedy-D to do so too.

The Distribution and Deposition Algorithm (DDA): Finally, we present an algorithm for the MASP based on a distribution and deposition paradigm. Since the deposition problem has been well studied, the idea is to first discover “*desirable properties*” of the array that contribute to shorter SCS during the deposition step. Then, we should distribute the probes in S to the M arrays so that the resulting arrays have these desirable properties.

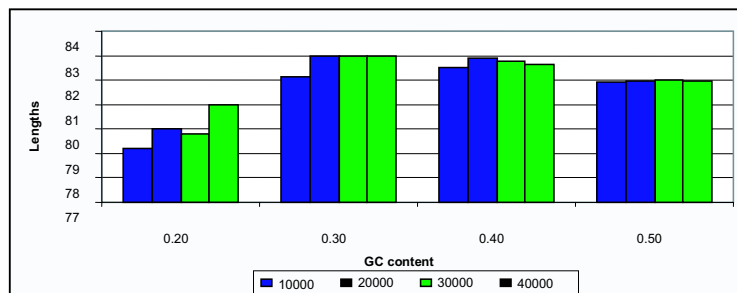


Figure 2: The effect of different number of oligos and GC contents on the lengths of the synthesis sequences.

Factors Affecting the Length of the Synthesis Sequences: Our first step is to analyze different factors that may influence the lengths of the synthesis sequences. The *size of the array* is one such factor, and the *alphabet content* is another important factor. In this analysis, we set the number of probes in each array to the range 10,000, 20,000, 30,000 and 40,000, and we have used $\Sigma = \{A, C, G, T\}$. It is obvious that other alphabet sets have similar properties. The GC content is defined as the sum of alphabet contents of “G” and “C”. Because of the symmetric property of GC content, the setting of

20%, 30%, 40% and 50% is appropriate. Also, we have used probes of length 25 in this analysis. The synthesis sequences are generated by LAP algorithm [9].

The results are illustrated in Figure 2. From Figure 2, we see that the synthesis sequences are shorter for arrays where the average GC content is 20% and 50%, compared to arrays with average GC content 30% and 40%. We also observed that smaller arrays (fewer probes in the array) gives shorter synthesis sequences in general, which is to be expected. The results also show that the arrays with 10,000 probes have shorter synthesis sequences, while there is not much difference in the lengths of the synthesis sequences between arrays with 20,000, 30,000 and 40,000 probes. For arrays with GC content of 50%, the synthesis sequences for the different array sizes are about the same and they are all relatively short.

The Distribution Algorithm: Based on these observations, we see that for distribution process, the GC contents of the probes should be chosen so that (a) they are similar on an array, and (b) on each array, they are near 20% (or 80%, given the symmetric property of the nucleic acids) or 50%. Also, arrays with smaller sizes have shorter synthesis sequences. Therefore, we have designed the distribution process in which probes are iteratively selected so that each array has such property. We have given the general pseudo codes of the distribution process in Figure 3.

Distribution (MA)

```

Input: A set of probes  $S=\{s[1],s[2],\dots,s[N*M]\}$ ;
          ( $M = \text{number of arrays, and } N = \text{size of each array}$ )
Output: Multiple arrays  $MA=\{A[1], A[2] \dots A[M]\}$ 
begin
  for (each probe  $s[i]$  in  $S$ ) do
    Compute the frequency count for each alphabet  $\alpha[k]$ 
    in the of probe  $s[i]$ ; (for all  $k=1,2,\dots,q$ );
  endfor
  for (each alphabet  $\alpha[k]$ ) do
    Sort the probes in  $S$  by the frequency of  $\alpha[k]$ 
    to generate sorted array  $SA[k]$ ;
  endfor
  while(there exist non-empty array  $SA[k]$ ) do
    if (there exist an empty array  $A[j]$ )
      then  $A^*=S[j]$ 
      else  $A^*$  is the array with  $\alpha[k]$  content most similar to  $SA[k]$ 
    endif
    for (probe  $s[i]$  in  $SA[k]$  in decreasing frequency count)
      Add  $s[i]$  into  $S^*$ ;
      Delete  $s[i]$  from every sorted array  $SA[*]$ ;
      If ( $|A^*|=N$ ) then Break; endif
    endfor
  endwhile
end;

```

Figure 3: The pseudo codes for the distribution of sequences for different sets.

The Deposition Step: Once the probes have been distributed, the deposition of the probes for each of the M arrays is done by (a) the Sum Height (SH) algorithm [7] or (b) the LAP algorithm [9]. The LAP algorithm uses a look-ahead version of the SH algorithm ((3,1)-*LA-SH*, to be precise) for finding a synthesis sequence, and a post-processing algorithm to further reduce the length of this sequence while preserve the common supersequence property. We refer the reader to [9] for details of the LAP algorithm.

Overall, our DDA algorithm has time complexity of $O(N^3 K^2 + NMq \log(NM))$.

Performance Ratio: To compare the performance of different algorithms, we also propose the following performance ratio. For each array, the simple periodic supersequence $S_{PS} = (\alpha_1 \alpha_2 \dots \alpha_q)^K$ has a length of qK . Thus, for multiple arrays $MA = \{A_1, A_2, \dots, A_M\}$, with N probes per array, each of length K , the periodic supersequence gives an obvious upper bound of $qKNM$ for $cost(MA)$. Thus, it is natural to define the *performance ratio* $R_X(MA)$, of an algorithm X as $R_X(MA) = cost_X(MA) / qKNM$, where $cost_X(MA)$ is the cost for the multiple array MA produced by algorithm X .

We have computed and compared the *performance ratios* of different algorithms, as an indicator of how well the algorithm performs. Smaller performance ratios indicate better solution. It is easy to see that for DDA, $R_{DDA}(MA) \leq 1$ since the synthesis sequences produced by LAP is also bounded by qK . In our experiments, $R_{DDA}(MA)$ is much smaller than 1.

3 Experimental Evaluation

3.1 Datasets and Experiment Settings

We have analyzed the performance of Distribution and Deposition algorithm (DDA), and compare it with Greedy algorithm in different settings. For the deposition process, both SH and LAP algorithms are used for analysis. We have implemented our program in C++ and Perl. The experiments are performed on a PC with 3.0GHz CPU and 1.0GB RAM, running a Linux system.

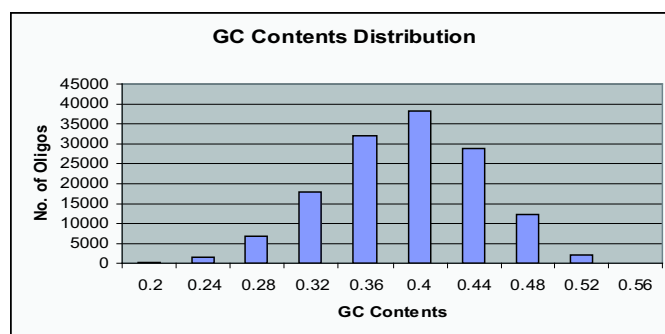
The datasets include randomly generated DNA sequences, as well as oligos randomly selected from real gene sequences. We have randomly generated some simulated DNA sequences with specific length and GC content similar to those of the real DNA sequences (details shown in following tables) for analysis of the effectiveness of distribution method and deposition method. The presence of alphabets ‘‘A’’ and ‘‘T’’ are uniformly distributed in the oligo sequences, so dose alphabets ‘‘C’’ and ‘‘G’’ (details refer to random DNA oligos generator at <http://www.comp.nus.edu.sg/~ningkang/random.html>). The analysis on the real gene sequences are more important, so we have randomly selected some gene sequences from rodent sequences set, plant sequences set and EST sequences set (details shown in following tables), which are obtained from the well-known GenBank [2] (<ftp://ftp.ncbi.nih.gov/genbank/>). To facilitate our research, we have chosen the Primer3 software [12] to select the good probes (oligos) from genes. The probe length (details shown in following tables), extend of complimentarity and temperature (default values) are thus optimized. After suitable probes are selected from each of the genes, we check all of the probes, and remove those which will cause cross-hybridization.

For the multiple arrays, we have first analyzed the performance of distribution method and deposition method separately. After these analyses, we have performed analyses and comparisons with different settings on the simulated DNA sequence and real gene sequences. One of the comparisons is to compare DDA on multiple arrays with deposition algorithm (SH and LAP) on a single array (which contains all of the oligos in MS). The purpose of this comparison is to show that even there can be very large array so that all of the oligos can fit in, the cost of DDA on multiple arrays is still better.

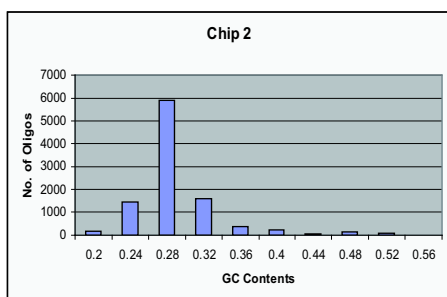
3.2 The Experimental Results

Assessment of the Distribution Process: We have first assessed the distribution method and deposition method separately. The first analysis on distribution method is about the complementarities of the array, and their melting temperature. Results (details not shown) shows that the oligos on each of the arrays have very little cross-hybridization, and the melting temperatures are also close.

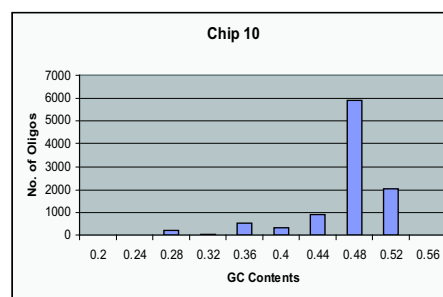
We have also observed that for oligos on a single array, the GC contents are close, and they are different from 30% ~ 40%. We have illustrated one example to show that the oligos with different GC contents are well distributed onto multiple arrays. We have selected oligos (by Primer 3) from the real EST gene sequences for this example, and distribute them on 16 arrays. The number of genes is 15,348, and there are 10 oligos selected for each gene. The array size is 10,000.



(a)



(b)



(c)

Figure 4: The effect of distribution process on the GC contents of the oligos on multiple arrays.

The distribution of GC contents for all of the oligos is shown in Figure 4(a). From Figure 4(a) we see that the GC contents of the oligos are ranging from about 20% to 56%. After distribution process, the 16 arrays are numbered with increasing average GC contents. We have randomly selected two typical arrays that are not extreme cases (extreme GC contents). From the array number 2 and number 10 (Figure 4(b) and Figure 4(c)), we see that the GC contents of the oligos on each array are very similar, while most of these arrays have average GC contents different from 30% ~ 40%. Therefore, the distribution process is effective.

The effectiveness of the LAP deposition algorithm has been empirically proven by our previous research [10]. The LAP deposition algorithm can generate shorter synthesis sequences than Alphabet [1], Majority Merge [5], Tournament [16], Greedy [16], and Reduce-Expand [1] algorithms.

Results on Simulated DNA Sequences: It is trivial that the performance ratios are better than using “periodic supersequence”. In this section, we have compared DDA with Greedy algorithm, and we have also compared DDA with deposition algorithms applied on single array (with all of the oligos on an array).

Based on previous analysis, for DDA, we have set the size of the arrays to be 10,000 and 20,000 (other array sizes have similar comparative performance to 20,000, as shown in Figure 2). The lengths of the oligos being tested are 25 and 35. For each of the settings (array size (N), number of genes (G), gene length (L), number of oligos per gene (OPG), oligo length (K)), we have generated 10 datasets. The results are the average length of synthesis sequences for each specific setting.

To analyze the effectiveness of multiple arrays, the results of SH (R_{SH}) and LAP (R_{LAP}) on *all of the oligos* in *MS* (that is, on single array) are computed, and compared with DDA on multiple arrays using SH ($R_{DDA-SH}(MS)$) and LAP deposition method ($R_{DDA-LAP}(MS)$). The results of Greedy algorithm ($R_{Greedy-A}(MS)$ and $R_{Greedy-D}(MS)$) is also computed. Results are shown in Table 1.

Table 1: The results of DDA on simulated datasets. The performance ratios are listed for different datasets.

Datasets	N	G	L	OPG	K	R_{SH}	R_{LAP}	$R_{Greedy-A}$	$R_{Greedy-D}$	R_{DDA-SH}	$R_{DDA-LAP}$
Case1	10,000	5,000	500	5	25	0.82	0.81	0.95	0.81	0.81	0.79
	10,000	5,000	500	10	25	0.82	0.82	0.91	0.81	0.80	0.79
	10,000	5,000	500	5	35	0.70	0.69	0.71	0.71	0.70	0.69
	20,000	5,000	500	5	25	0.82	0.81	0.85	0.81	0.81	0.81
Case2	10,000	5,000	1,000	5	25	0.84	0.82	1.01	0.83	0.81	0.78
	10,000	5,000	1,000	10	25	0.83	0.82	0.95	0.82	0.81	0.80
	10,000	5,000	1,000	10	35	0.71	0.71	1.23	0.73	0.71	0.71
	20,000	5,000	1,000	5	25	0.84	0.82	0.96	0.83	0.82	0.78
Case3	10,000	10,000	500	5	25	0.84	0.81	0.99	0.83	0.82	0.79
	10,000	10,000	500	10	25	0.84	0.81	1.10	0.83	0.82	0.79
	10,000	10,000	500	5	35	0.72	0.70	1.02	0.71	0.72	0.70
	20,000	10,000	500	5	25	0.84	0.81	1.05	0.84	0.83	0.80
Case4	10,000	10,000	1,000	5	25	0.82	0.81	0.98	0.85	0.82	0.80
	10,000	10,000	1,000	10	25	0.84	0.81	1.09	0.83	0.82	0.78
	10,000	10,000	1,000	5	35	0.72	0.72	1.00	0.75	0.71	0.68
	20,000	10,000	1,000	5	25	0.82	0.81	1.10	0.83	0.83	0.80
Case5	10,000	20,000	500	5	25	0.84	0.83	0.92	0.80	0.81	0.78
	10,000	20,000	500	10	25	0.84	0.83	1.03	0.81	0.81	0.78
	10,000	20,000	500	5	35	0.72	0.72	1.09	0.70	0.69	0.69
	20,000	20,000	500	5	25	0.84	0.83	1.08	0.82	0.83	0.80

From the result in Table 1, we can see that except for Greedy-A, all of the performance ratios are smaller than 0.85, which is much better than R_{ps} . $R_{Greedy-A}$ is bigger than one for many cases, which are worse than many other algorithms. Since the time needed by Greedy-A is also magnitudes larger than other algorithms, we think this greedy algorithm has bad performance, and will not compare it with other algorithms in later parts. $R_{Greedy-D}$ is smaller than 1, but still a little bit larger than R_{DDA-SH} and $R_{DDA-LAP}$. Compared between multiple arrays and single array, we observe that the performance ratios $R_{DDA-SH}(MS)$ and $R_{DDA-LAP}(MS)$ are always equal or smaller than 0.8 for oligos of length 25, and equal or smaller than 0.7 for oligos of length 35. While if all of these oligos are in one array, then R_{SH} and R_{LAP} are more than 0.8 for oligos of length 25, while more than 0.7 for oligos of length 35. This is more obvious when the number of genes is larger ($\geq 10,000$) and the number of oligos per gene is larger (10). Since the array sizes do not affect the lengths of the synthesis sequences much (refer to Figure 2), we believe that the GC contents play an important role for the small performance ratios. The results of LAP are better than SH, showing that LAP is a better deposition algorithm. It is also observed that performance ratios on longer (35) oligos are smaller than those on shorter (25) oligos. Previous researches [6] show that the optimal performance ratio is about 0.60 to 0.75, compare $R_{DDA-LAP}$ to this, we see that the performance ratios of the DDA (using LAP) is very good.

To see the actual length of synthesis sequences for multiple arrays, we have given these lengths for some datasets, as shown in Table 2. Note that for datasets with few DNA sequences ($N \leq 5,000$),

there are sometimes not enough appropriate long oligos to deposit onto multiple arrays (even with array size of 10,000).

Table 2: The results of DDA on simulated datasets. The lengths of the synthesis sequences for each array are listed for different datasets.

Datasets	N	G	K	Length _{SH}	Length _{LAP}	Length _{Greedy-D}	Length _{DDA-SH}	Length _{DDA-LAP}
Case1	10,000	25,000	25	82	81	81,81,80	82,80,80	80,79,78
	10,000	50,000	25	82	82	82,82,82,81,80	83, 82,80,79,80	81,81,80,74,77
	10,000	25,000	35	98	97	100	98	97
	20,000	25,000	25	82	81	81,80	82,80	81,78

It is obvious that the lengths of synthesis sequences by DDA on multiple array (DDA-SH and DDA-LAP) are shorter than results of Greedy-D, and are shorter than those on single array (SH and LAP).

Results on Real Gene Sequences: Again, for the real genes, we have compared DDA on multiple arrays with SH and LAP applied on single array. The same settings are compared. Results are shown in Table 3.

Table 3: The results of DDA on real datasets. The performance ratios are listed for different datasets.

Datasets	N	G	L	OPG	K	R _{SH}	R _{LAP}	R _{Greedy-D}	R _{DDA-SH}	R _{DDA-LAP}
gbest1	10,000	15,348	50~100	5	25	0.84	0.83	0.83	0.83	0.81
	10,000	15,348	50~100	10	25	0.85	0.85	0.83	0.82	0.80
	10,000	5,074	50~100	5	35	0.76	0.76	0.76	0.75	0.74
	20,000	15,348	50~100	5	25	0.84	0.83	0.83	0.83	0.82
gbpln1	10,000	28,916	500~50,000	5	25	0.86	0.86	0.84	0.84	0.82
	10,000	28,916	500~50,000	10	25	0.86	0.86	0.84	0.84	0.82
	10,000	13,450	500~50,000	5	35	0.80	0.77	0.77	0.77	0.75
	20,000	28,916	500~50,000	5	25	0.86	0.86	0.86	0.85	0.82
gbrod1	10,000	8,958	1,000~100,000	5	25	0.84	0.83	0.83	0.82	0.80
	10,000	8,958	1,000~100,000	10	25	0.86	0.84	0.83	0.83	0.81
	10,000	4,026	1,000~100,000	5	35	0.76	0.76	0.77	0.76	0.75
	20,000	8,958	1,000~100,000	5	25	0.84	0.83	0.84	0.84	0.81

From the result in Table 3, the effectiveness of DDA on real gene sequences is obvious. The results of R_{Greedy-D} are always equal or bigger than R_{DDA-SH} and R_{DDA-LAP}. R_{DDA-SH}(MS) and R_{DDA-LAP}(MS) are about 0.05 less in performance ratios than R_{SH} and R_{LAP}, respectively. These are more obvious for more oligos (more than 1,000,000). Again, the results of LAP are better than SH. Also, the results are better on smaller size arrays, which is consistent with our previous analysis. The GC content still plays an important role.

Comparing Table 1 with Table 3, we observe that performance ratios on real gene sequences are about 0.1 more than those on simulated DNA sequences. This is probably due to the large variances in the real gene sequences (and thus oligos).

Computational Efficiency: The time complexity of DDA is already analyzed in previous section. In experiments, the time needed for generation of the synthesis strategy for multiple arrays depends on the probe selection time and the time for the generation of distribution and deposition strategy. The time for the probe selection is relatively long, costing about 30 minutes for 10,000 genes with average length of 10,000. The time for DDA is quite reasonable, with around 10 minute for 50,000 25-mer oligos (10,000 genes) when the LAP deposition method is used (around 30 seconds when SH deposition method is used). Greedy-D is faster with about 1 minute for 50,000 length-25 sequences. We also note that the time is very long if we try to generate synthesis strategy for all of the oligos

onto single array using LAP (about 5 to 30 minutes, depending on size of dataset). The memory used by DDA is also quite reasonable, with about 100MB for 100,000 25-mer oligos (20,000 genes).

4 Discussion and Conclusion

The use of multiple arrays (for one experiment) is of growing importance with increasingly larger scale microarray experiments such as those for whole genome expression analysis. Therefore, it is important to have efficient algorithms for the design and synthesis of multiple arrays that will minimize the overall synthesis cost. However, there is, as yet, little research on this problem.

In this paper, we formulate the multiple array synthesis problem (MASP) and we propose a cost function to more accurately reflect the total synthesis cost for multiple arrays, and defined a performance ratio measurement. We propose an algorithm, called the Distribution and Deposition Algorithm (DDA), for solving the MASP. The focus is on the minimization of *overall cost* of probe synthesis.

The experiments show that DDA is effective in the synthesis of many oligos onto multiple arrays, and its cost is smaller than greedy algorithms. DDA can always have small performance ratios. And the small performance ratios are valuable for the later chemical synthesis process (less time, less error etc.). The time needed by DDA is reasonable, and it is short compared to the probe selection process.

As the scale of microarray experiments is increasing, it is foreseeable that there will be huge scale microarray experiments in the future. We foresee that the different synthesis strategies can be applied on different arrays in parallel, and such parallel process will greatly reduce the time needed and facilitate the broad scale of microarray experiments.

Currently, we have used a publicly available probe selection software (Primer3, [12]) to select oligo probes for multiple arrays. The whole process of selection, distribution and deposition of oligos for multiple arrays can be integrated tighter, and we are under way of developing such effective and efficient process. There are different situations that multiple oligo arrays can be used, and specific experiments may have specific requirements [15]. DDA for the oligos with specific requirements is also important to the experimenters, and this is another direction of our further research.

Acknowledgments

We thank Pavel Pevzner for insightful discussion on this work, and the anonymous reviewers for valuable comments on the paper. This work was partially supported by the National University of Singapore under grant R252-000-199-112.

References

- [1] Barone, P., Bonizzoni, P., Vedova, G. D., and Mauri, G., An approximation algorithm for the shortest common supersequence problem: an experimental analysis, *ACM Symposium on Applied Computing*, 56–60, 2001.
- [2] Benson, D. A., Boguski, M., Lipman, D. J., and Ostell, J., GenBank, *Nucleic Acids Res.*, 22:3441–3444, 1994.
- [3] Fodor, S., Read, J., Pirrung, M., Stryer, L., Lu, A., and Solas, D., Light-directed, spatially addressable parallel chemical synthesis, *Science*, 251:767–773, 1991.
- [4] Gerhold, D., Rushmore, T., and Caskey, C., DNA chips: promising toys have become powerful tools, *Trends Biochem Sci.*, 24:168–173, 1999.

- [5] Jiang, T. and Li, M., On the approximation of shortest common supersequences and longest common subsequences, *SIAM Journal of Computing*, 24:1122–1139, 1995.
- [6] Kahng, A. B., Mandoiu, I. I., Reda, S., Xu, X., and Zelikovsky, A. Z., Computer-Aided Optimization of DNA Array Design and Manufacturing, *Computer-Aided Design of Integrated Circuits and Systems, IEEE Trans.*, 25:305–320, 2006.
- [7] Kasif, S., Weng, Z., Derti, A., Beigel, R., and DeLisi, C., A computational framework for optimal masking in the synthesis of oligonucleotide microarrays, *Nucleic Acids Res.*, 30:e106, 2002.
- [8] Lockhart, D. J., Expression Monitoring by Hybridization to High-Density Oligonucleotide Arrays, *Nat. Biotechnol.*, 14:1675–1680, 1996.
- [9] Ning, K., Choi, K. P., Leong, H. W., and Zhang, L., A Post Processing Method for Optimizing Synthesis Strategy for Oligonucleotide Microarrays, *Nucleic Acids Res.*, 33:e144, 2005.
- [10] Ning, K. and Leong, H. W., Towards a Better Solution to the Shortest Common Supersequence Problem: A Post Processing Approach, Symposium of Computations in Bioinformatics and Biosciences, 84–90, 2006.
- [11] Pease, A., Solas, D., Sullivan, E., Cronin, M., Holmes, C., and Fodor, P., Light-Generated Oligonucleotide Arrays for Rapid DNA Sequence Analysis, *Proceedings of the National Academy of Sciences*, 91:5022–5026, 1994.
- [12] Rozen, S. and Skaletsky, H. J., Primer3 on the WWW for general users and for biologist programmers, *Methods Mol. Biol.*, 132:365–386, 2000.
- [13] Schena, M., Shalon, D., Davis, R., and Brown, P., Quantitative monitoring of gene expression patterns with a complementary DNA microarray, *Science*, 270:467–470, 1995.
- [14] Singh-Gasson, S., Green, R., Yue, Y., Nelson, C., Blattner, F., Sussman, M., and Cerrina, F., Maskless fabrication of light-directed oligonucleotide microarrays using a digital micromirror array, *Nat. Biotechnol.*, 17(10):974–978, 1999.
- [15] Stoughton, R. B., Applications of DNA microarrays in biology, *Annu. Rev. Biochem.*, 74:53–82, 2005.
- [16] Timkovsky, V. G., On the approximation of shortest common non-subsequences and supersequences, *Technical Report*, 1993.
- [17] Tomiuk, S. and Hofmann, K., Microarray Probe Selection Strategies, *Brief. Bioinform.*, 2:329–340, 2001.