Efficient Probe Selection in Microarray Design

Leszek Gąsieniec, Cindy Y. Li, Paul Sant, Prudence W.H. Wong

Abstract— **The DNA microarray technology, originally developed to measure the level of gene expression, had become one of the most widely used tools in genomic study. Microarrays have been proved to benefit areas including gene discovery, disease diagnosis, and multi-virus discovery. The crux of microarray design lies in how to select a unique probe that distinguishes a given genomic sequence from other sequences. However, in cases that the existence of a unique probe is unlikely, e.g., in the context of a large family of closely homologous genes, the use of a limited number of non-unique probes is still desirable.qyy**

Due to its significance, probe selection attracts a lot of attention. Various probe selection algorithms have been developed in recent years. Good probe selection algorithms should produce as small number of candidate probes as possible. Efficiency is also crucial because the data involved is usually huge. Most existing algorithms usually select probes by filtering, which is usually not selective enough and quite a large number of probes are returned. We propose a new direction to tackle the problem and give an efficient algorithm to select (randomly) a small set of probes and demonstrate that such a small set of probes is sufficient to distinguish each sequence from all the other sequences. Based on the algorithm, we have developed a probe selection software RANDPS**, which runs efficiently and effectively in practice. A number of experiments have been carried out and the results will be discussed.**

I. INTRODUCTION

DNA microarrays [6] have become a very important research tool. They are used for performing a large number of hybridization experiments simultaneously. Besides their prevalent use to measure the amount of gene expression [23] in a cell, microarrays are an efficient tool for making a qualitative statement about the presence or absence of biological target sequences in a sample. A DNA microarray ("chip") is a plastic or glass slide which consists of thousands of (about 60,000) short DNA sequences known as probes. A probe is a contiguous substring of a gene, which acts as its fingerprint (a.k.a signature). Fingerprinting is the technique of identifying or confirming specific DNA fragments by "cutting" them with special enzymes, observing the unique pattern of the fragment sizes that result, and then comparing this with the pattern of a known DNA fragment. Usually, a probe is 20-70 base pairs (bps) long.

A typical application of microarrays is detection of different members of a virus family in a sample. In this case, we have a database of the DNA sequences (called targets) for a known family of viruses and we wish to identify an unspecified virus whose DNA sequence is present in the database. What we need is a set of hybridization tests based on good selection of probes such that on every known family, the set of answers (red, green, yellow or black signal on the microarray) that we receive is unique with respect to any other virus in the database. Therefore, the probe should bind only to its corresponding sequence, and not to any other sequence available in the database. If this is the case, we say that the probe is unique. The quality of the probe selection process can be expressed by the proportion of DNA sequences in the database possessing unique probes.

Depending upon the application, the hybridization experiments are conducted using either single or multiple probes and very often under the assumption that there is only one target present in the sample. The *probe selection problem* we studied is to find the smallest number of good probes with specified length for every gene in the genome, that satisfies (1) *Homogeneity* - melting temperature for every probe should be within some pre-defined range, to make sure that the probes are able to hybridize to their intended targets at about the same experimental temperature; (2) *Sensitivity* - to detect self-complementarity of probes and (3) *Specificity* - identifies probes that are unique to each gene in the genome on the basis of the Hamming Distance [7] as the similarity measure[1]. The specificity check is computationally expensive and takes the most time in the probe selection process. The brute force approach for specificity checking scans through the whole length-n genome for every length-m probe and determines if the Hamming distances are large enough. Such a process is expensive and requires $O(mn^2)$ time. For example, brute force specificity checking would take about 72 hours for S.pombe genome of length 7.1×10^6 bps and is thus impractical for

Leszek Gąsieniec is with the Department of Computer Science, The University of Liverpool, L69 3BX, UK (email: leszek@csc.liv.ac.uk)

Cindy Y. Li is with the Department of Computer Science, The University of Liverpool, L69 3BX, UK (email: cindy@csc.liv.ac.uk)

Paul Sant is with the Department of Computing and Information Systems, University of Luton, LU1 3JU, UK (email: paul.sant@luton.ac.uk)

Prudence W.H. Wong is with the Department of Computer Science, The University of Liverpool, L69 3BX, UK (email: pwong@csc.liv.ac.uk)

[1]For two strings s and t, the Hamming distance $H(s,t)$ is the number of positions where the characters at corresponding positions of the two strings differ. For example, if s = 00010101, t = 00011010, then $H(s,t)$ = 4.

large genomes.

To further improve the quality of the probe selected, we use additional constraints, including the rules described by [14] and those used in the Affymetrix probe selection criteria: (1) no single base (A, T, C or G) exceeds 50% of the probe size; (2) the length of any contiguous sequence of As and Ts or Cs and Gs region is less than 25% of the probe size; (3) GC-content is between 40% and 60% of the probe sequence (GC-content is the percentage of nucleotides which are G or C in the sequence). We refer to these constraints as *Quantitative criteria*.

In this work, we focus on efficient selection of a minimal set of good probes which leads to the use of smaller and therefore cheaper microarrays rather than reporting all probes in order to increase the efficiency. The search for probes should be both time and space efficient. All probes should be far (in terms of Hamming distance) from each other. We propose a new approach that takes as input a set of known gene sequences and builds a small cardinality set of probes allowing us to identify the unknown target in the sample. Instead of checking all possible probes, we exploit randomization. We randomly pick probes with some minimal criteria checking. Our experimental results show that almost all genes can be uniquely identified by a single probe; the others need at most a combination of two probes.

A. Previous work

Selection criteria

Lockhart *et al.* [14] were among the first to study the probe selection problem. The quantitative criteria they proposed are widely used [2, 12, 19, 20, 25, 26], with some minor variations. Homogeneity and specificity were also used in their algorithm, though the exact algorithm has not been published. Homogeneity is used in almost all existing algorithms, which is usually measured by the nearest neighbor model (NNM) proposed by SantaLucia [22]. Kaderali and Schliep [9] focus on melting temperature (T_m) and compute the optimal (the best) probe using suffix trees and dynamic programming. However, this is too slow, especially for large genomes, e.g., it takes 2 weeks to design a probe set for the whole yeast genome. A different formula was also used in [2, 28] to calculate T_m. Other work like [15] also only focuses on criteria related to thermodynamic evaluation. It is generally agreed that T_m and free energy can be used as parameters to evaluate probe hybridization behavior and have been shown to be useful [12] [2].

[2]Some researchers [17, 29] argue that thermodynamic criteria may not be adequate for microarray analysis, we leave this decision to biologists while we mainly provide a computational tool to design probe using thermodynamic criteria.

As for specificity, there are two major measurements: Hamming distance [12, 18, 25] and BLAST search [2, 19, 20, 26, 28]. Using BLAST [1] (http://www.ncbi.nih.gov/blast/), the algorithms assume the search is done in advance and the results passed as input. The computation time, thus, depends on the number of sequences in the BLAST database; e.g., the algorithm by Rouillard *et al.* [20] takes from 4 to 12 hours to design up to three 45mers probes per gene for most of the bacterial genome. On the other hand, if Hamming distance is used, the algorithm becomes fully automatic since the distance is calculated purely computationally.

Sensitivity is also a popular consideration to avoid self-binding of probes selected. This may be done by checking the stability of the secondary structure formed (stable means not a good candidate). MFOLD [30], Vienna RNAfold [8] and Smith-Waterman [24] algorithms have been used in [2, 15, 20, 28] for this purpose. Other algorithms [12, 18, 19, 25, 26] directly check sensitivity by eliminating probes that are self-complementary.

Existing software

Based on all three criteria, a number of algorithms have been proposed. Li and Stormo [12] use a fast approximate matching search algorithm myersgrep [16] for uniqueness checking. However, the algorithm is still not fast enough for computing probes of large genome sets. It takes almost four days to design a length-24 probe set for *Saccharomyses cerevisiae* genome (12M bps with about 6000 genes). Rahmann [18] presented a fast algorithm eliminating candidates that have a long common factor with other genes. This algorithm allows selection of probes for large genomes like *N.crassa* with total size 43MB in 4 hours on Compaq ES40 (833 MHz) with 16GB memory. However, the approach only designs short probes and requires a lot of space during computation. Sung and Lee [25] attempted to reduce the time complexity by using several filtering steps and exploiting the Pigeon Hole Principle [3] to avoid redundant comparisons. A length 50mers probe set for *N.crassa* can be generated in 3.5 hours on SunFire Workstations (700MHz) with 4GB memory. Relogio *et al.* [19] proposed a modified version of the Gene Skipper software; the specificity check only considers perfect matches ignoring possible mismatches which may still result in probes that are non-specific and bind to other sequences in addition to the target. Tolonen *et al.* [26] also only considers perfect match; specificity checking requires no region of self-complementarity of five or more bases at either end; the Quantitative criteria is relaxed such that the GC-content is between 25% to 75%. Wright and Church [28] proposed an algorithm which terminates once good probes (not necessary optimal) are found. They also introduced an interesting concept

to define probe sequence complexity based on the Lempel-Ziv (LZ) compression algorithm [11]. Independently, this idea was also employed by Bozdech *et al.* [2].

Recently, Klau *et al.* [10] presented the first approach to select a minimal probe set for the case of non-unique probes in the presence of a small number of multiple targets in the sample. Their approach is based on Integer Programming mixed with a branch-and-cut algorithm. Their preliminary implementation is capable of separating all pairs of targets optimally in a reasonable time and achieves a considerable reduction on the numbers of probes needed compared to previous greedy algorithms.

B. Our result

We propose a new approach to probe selection for DNA microarrays. Our algorithm performs efficient probe selection providing unique probes for almost all target sequences in the considered genomes (the datasets are summarized in Table I). The best results are obtained on large genomes. This is due to the size of datasets from which our randomized procedure profits in the context of certain probability laws governing large numbers. More detailed discussion on the selection of our procedure can be found in Section II-C. Our algorithm is quick because exhaustive search is not required, it is also fully automatic as we do not rely on external software.

The experimental results show that our algorithm is much faster than existing algorithms especially for large genomes. Our randomized procedure selects probes efficiently from short (24 bases) through long (64 bases) probes for large genomes. Furthermore, our approach significantly reduces the number of probes needed in microarray design.

The length of the probes designed by existing software ranges from 20 to 70: around 20 [9, 12, 14, 25, 26], around 30 [9, 18] around 50 [12, 20, 25], and around 70 [2, 12, 25, 28]. Our software is able to design probes of various length in this range (see Section III). As for the number of probes returned, some algorithms returned all probes [25] requiring longer computational time while most of the other software return a small number of probes. We follow the approach adopted by most software and report a small number (this is feasible due to the randomization procedure we employed).

II. Probe selection

We start with the criteria of the probe selection problem.

A. Probe selection criteria

Every length-m substring of a gene sequence is called a *candidate*. For every candidate, we check whether it satisfies fundamental probe selection criteria: (1) Quantitative criteria; (2) Homogeneity; (3) Sensitivity. Any candidate that passes all these three criteria is called a *probe*.

Quantitative criteria are described by Lockhart *et al.* [14] and are used in Affymetrix probe selection criteria: (1) the content of any single base (As, Ts, Cs or Gs) does not exceed 50% of the candidate size; (2) the length of any contiguous As and Ts or Cs and Gs region is less than 25% of the candidate size; (3) GC-content is between 40% and 60% of the candidate.

Homogeneity criterion requires that the melting temperature of candidates should be within some pre-defined range, because a good probe set needs to hybridize to their intended targets at about the same temperature in experiments.

Melting temperature [21] of a probe is the temperature at which 50% of the oligonucleotides and its perfect complement are in duplex. Since it is impossible to know the target DNA concentration, the calculation is approximate, but still useful. Melting temperature T_m of each candidate in our approach is calculated as

$$T_m = \frac{\Delta H}{\Delta S + R \times \ln(c/4)} - 273.15$$

where ΔH and ΔS are the enthalpy and entropy for the helix formation respectively, R is the molar gas constant (1.987 cal/(K mol)), and c is the total molar concentration of the annealling oligonucleotides when oligonucleotides are not self-complementary[1].

Sensitivity criterion filters out candidates prone to self-complementarity (see Figure 1). This is to reject all candidates who may fold back on themselves rather than on target sequences. Consider every segment of a candidate of length ℓ. If its reversal forms a consecutive length ℓ complementary segment within itself, the candidate is considered prone to fold back on itself.

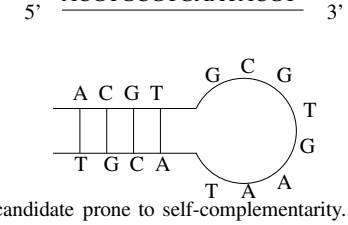

Fig. 1. A candidate prone to self-complementarity.

[1]The nearest neighbor model is well adapted to compute the T_m for short sequences, but may lead to an overestimate of the T_m of probes longer than 50nt. Other methods of T_m is calculated by the formula [27] T_m = 81.5 + $(16.6 \log([Na^+])$ + $41[(G + C)/length]$ - $(500/length)$ where $[Na^+]$ is the sodium ion concentration. However, evidence for size limitation of the nearest neighbor model and parameters is sparse [2]. For 70-mer probes, the difference between the T_m values calculated using these method is negligible [28].

TABLE I

INFORMATION OF THE DATASETS.

	E.coli	S.cerevisiae	S.pombe	N.crassa	A.thaliana	Mouse Chromosome 2
Total length	4,752,411	8,783,280	7,272,320	17,484,362	33,581,216	182,887,278
No. of genes	5,253	5,888	5,471	10,633	26,186	1,302
Avg. length per gene	905	1,492	1,329	1,644	1,282	140,466

Another useful measure for sensitivity is the free energy. The total difference in the free energy of the folded and unfolded states of a DNA duplex is approximated by a nearest-neighbour model:

$$\Delta G_i(total) = \sum_j n_{ij}\Delta G_j + \Delta G_i(init) + \Delta G_i(sym) \quad (1)$$

where each different oligonucleotide duplex is given the subscript i, ΔG_j is the free energy for the 10 possible Watson-Crick nearest-neighbour stacking interactions, n_{ij} is the number of occurrences of each nearest neighbour j, in each sequence i, $\Delta G_i(init)$ is the initiation free energy, and $\Delta G_i(sym)$ equals +0.4 kcal/mol if duplex i is self-complementary and zero if it is non-self-complementary [4]. DNA oligonucleotide nearest-neighbour thermodynamic parameters are available [22] and they allow prediction of oligonucleotide DNA hybridization energies.

The thermodynamic parameters used in our melting temperature and free energy calculation were estimated from experimental measurements on short probes. Therefore, although we used both to model long probe binding stability, the free energy values should be viewed as a function of binding stability on a relative scale, rather than be interpreted as the absolute free energy generated during DNA duplex formation.

In this work, we are mainly interested in efficient selection of *unique probes*, playing a role of gene signatures. We say that probe p is a *unique probe* for gene g in a genome, if and only if it occurs in g and there is no close occurrence (in terms of Hamming distance, see Specificity criterion) of p in any other gene of the genome.

Specificity identifies probes that are unique to each gene in the genome. This condition minimizes cross-hybridization of the probes with other gene sequences. For two strings s and t, the Hamming distance $H(s,t)$ is the number of positions where the characters at corresponding positions of the two strings differ. For example, if s = 00010101, t = 00011010, then $H(s,t)$ = 4. Hamming distance has been used as the basis for coding theoretic approaches [5, 13] to the DNA word design problem. In particular, Hamming distance becomes a powerful tool for determining closeness/similarity and recently has been adopted as the specificity measure [12, 18, 25]. Thus, if the Hamming distance between a probe and every candidate (excluding those candidates from the gene where the probe belongs to) is greater than some constant, the probe is said to be specific enough.

B. The problem

To summarize, given a set S of gene sequences also called *targets* or *target sequences*, the objective is to find for each gene sequence g in S a probe p (a contiguous substring of a gene) which hybridizes only to g. The probe p is said to be a unique probe of gene g. If such a probe p does not exist, i.e. p cross-hybridizes to other sequences in S, then find a small collection of probes that uniquely identifies g.

As shown in Table II, we are interested in finding a unique or a small group of probe(s) for each gene sequence in S. In this example, probe p_4 is a unique probe of gene sequence g_1, while p_2 and p_5 together identify g_2.

TABLE II

AN EXAMPLE OF TARGETS AND THEIR PROBES.

	p_1	p_2	p_3	p_4	p_5
g_1				•	
g_2		•			•
g_3		•	•		
g_4	•				•

C. Randomized probe selection algorithm

In this section, we present a new algorithm to select probes for DNA microarrays. Initially, our algorithm exploits several filters (based on probe selection criteria) to reduce the search space for probes. However, the main idea used here is to explore randomization to reduce the time complexity of the search. And indeed, randomly generated sequences are expected to possess properties of unique probes. E.g. probe selection criteria enforce balanced distribution of base pairs in probes which is naturally satisfied by random sequences. Moreover, the Hamming distance between two randomly chosen sequences of length m over a 4 letter alphabet is about $3m/4$, which is also highly desired property of a system of probes.

Our probe selection algorithm starts with the filtering stage applied on the whole genome. For each candidate, we test whether it passes the probe selection criteria (1), (2) and (3) and we eliminate all candidates who fail the test. For (2) Homogeneity, we require that the melting temperature lies between 78 and 90; for (3) Sensitivity, we reject all candidates with a self-complementary segment of length more than or

Algorithm 1 Probe selection (m: length of probe, S: genome, d: Hamming distance threshold, default as 5).

```
i ⇐ 0 and not_found ⇐ true;
for every gene g ∈ S: do
  while i < 5 and not_found is true do
    generate a random sequence r_i of length m;
    find the closest probe p_i in gene g;
    if H(p_i, q) ≥ d for all candidates q in other genes in S-{g} then
      p_i is chosen as the unique probe for g, report p_i, not_found ⇐
      false;
    end if
    i ⇐ i + 1;
  end while
end for
```

equal to 4. When this is completed, we iterate a probe selection procedure which acts on all genes in the genome consecutively. The probe selection procedure, see Algorithm 1, runs with gene $g \in S$, generates a unique (if it is able to find it) probe p for gene g. This is done as follows: (a) generate a random sequence r of length m; (b) find the closest match p of r among probes in the target; (c) check whether p satisfies specificity criterion. This process is iterated at most five times which allows us to obtain a good trade-off between the accuracy of the search procedure and its running time. The code of the procedure could be easily modified to incorporate the case when a unique probe is not found, in this case, we check whether a combination of any two (and very rarely three) already selected probes identifies uniquely the considered gene g.

It should be pointed out that our algorithm terminates once probes have been found to satisfy the probe selection criteria, rather than searching for optimal probes. In this end, we are in line with [2, 18–20, 25, 26, 28]. Using this strategy, our algorithm can select probes for large genomes for which algorithms demanding optimality are unsuccessful [9, 12].

D. Time complexity

The brute force approach for specificity checking scans through the whole length-n genome for every length-m probe and determines if the Hamming distances are large enough. Such a process is computationally expensive, requiring $O(mn^2)$ time. In comparison, we pick up a probe of length m by using randomization for every gene in the genome, then scan through the whole genome for specificity checking. By doing this, we do not need to check every probe in each gene which greatly reduce the time complexity. Thus, the time complexity of our algorithm is $O(kmn)$ where k (usually much smaller than n) is the number of genes in the whole genome, m is the length of probe and n is length of the whole genome.

E. Speeding up methods

To speed up our probe selection procedure, we exploit an "encoding" method to test self-complementarity and specificity. Consider every segment of a candidate of length 4, if its reversal forms complementary segment within itself, the candidate is prone to form a secondary structure. In particular, every segment of a candidate of length ℓ is encoded as follows:

$$\sum_{i=0}^{\ell-1} c_i \times 4^{(\ell-i-1)} \tag{2}$$

where c_i is either 0, 1, 2 or 3 (standing for A, C, G, T, respectively) representing the ith base of the segment. For example, a sequence ATCG is encoded as $0 \times 4^3 + 3 \times 4^2 + 1 \times 4^1 + 2 \times 4^0 = 54$. Furthermore, we exploit the tabling method to speed up the specificity checking process. We precompute a matrix $D = [D_{ij}]$ in which the rows and columns are indexed by numerical values obtained (by Formula 2) from all possible DNA sequences of length 4. Each entry D_{ij} is the Hamming distance between two DNA sequences with numerical value i and j. For example, if $i = 0$, representing AAAA, and $j = 255$, representing TTTT, then $D_{0,255}$ = 4. By looking up the appropriate entry in the table, Hamming distance between two probes of length-m can be quickly determined.

III. Analysis of experimental results

Our software RandPS is written in C and is developed and tested on Athlon XP2000+ Cluster with 2GB memory. The software is available on our website (http://www.csc.liv.ac.uk/∼cindy/RandPS/RandPS.htm). The size of RandPS code is 25KB which is simple and clean while being efficient and effective. Inputs of RandPS are FASTA formatted gene sequences, downloaded from the NCBI website (http://www.ncbi.nlm.nih.gov/). RandPS uses a size-n array, where n is the concatenated length of gene sequences of a genome, to store the inputs, together with another two size-n arrays to store the corresponding numerical value of each base in the genome and the status ($candidate$ or $probe$) of each position in the concatenated sequence. We report our results using several genomes that have been widely used for the probe selection problem. The genomes involved in the experiments are listed in Table I. These datasets have been used in experiments in [9, 12, 18, 20, 25, 26].

The experiments were undertaken in order to evaluate the performance of our software on various types of genomes. In terms of time consumption, it takes about 20 minutes to process the E.coli genome, 40 minutes to process the

S.cerevisiae genome, 60 minutes for S.pombe, 310 minutes for N.crassa, 470 minutes for Mouse chromosome 2 and 1520 minutes for A.thaliana. In terms of accuracy of probe selection, we are able to find unique probes for up to 99% of genes in the whole genome. The full details of the experimental results are shown in Tables III-VIII [3]. We have run our programme 30 times on each dataset for each probe length. In these four tables, the first three rows are basic information about the datasets, which are name of the genome, length of the genome and number of genes in the genome. The column "Probe length" lists the different lengths we used to test performance of our software. The column "1 probe" shows the number of genes which can be identified by a unique probe, while "2 probes" column shows the number of genes which require a combination of two probes for unique identification. The percentages in brackets are calculated on the basis of the number of genes with probes (i.e., total number of genes minus number of genes without probes). The "no probe" column shows the number of genes with no feasible probes.

TABLE III

RESULTS OF RANDPS FOR E.COLI.

Genome	E.coli		
Length	4,752,411		
# of genes	5,253		
Probe length	Number of genes requiring		
	1 probe	2 probes	no probe
24	4759 (90.7%)	490 (9.3%)	4
32	4791 (91.3%)	457 (8.7%)	5
40	4805 (91.6%)	442 (8.4%)	6
48	4808 (91.7%)	436 (8.3%)	9
56	4827 (92.1%)	413 (7.9%)	13
64	4832 (92.3%)	405 (7.7%)	16

TABLE IV

RESULTS OF RANDPS FOR S.CEREVISIAE.

Genome	S.cerevisiae		
Length	8,783,280		
# of genes	5,888		
Probe length	Number of genes requiring		
	1 probe	2 probes	no probe
24	5481 (93.2%)	401 (6.8%)	6
32	5516 (93.9%)	361 (6.1%)	11
40	5525 (94.2%)	341 (5.8%)	22
48	5549 (94.7%)	313 (5.3%)	26
56	5560 (95.0%)	292 (5.0%)	36
64	5560 (95.1%)	288 (4.9%)	40

As a further illustration of our software in terms of accuracy of the probe set, we compare the free energy of a group of our probes with the optimal probes with minimum free energy, which is found by using a brute force approach. This is shown in Figures 2-7 on examples of one hundred arbitrarily chosen genes for each genome. In comparison, probes that we found are very close to the optimal one. Thus our software is able to find high quality probes.

[3]The melting temperature range has been slightly modified for longer probe lengths 48, 56, and 64.

TABLE V

RESULTS OF RANDPS FOR S.POMBE.

Genome	S.pombe		
Length	7,272,320		
# of genes	5,471		
Probe length	Number of genes requiring		
	1 probe	2 probes	no probe
24	5061 (92.6%)	407 (7.4%)	3
32	5064 (92.6%)	404 (7.4%)	3
40	5131 (94.1%)	321 (5.9%)	19
48	5141 (94.3%)	308 (5.7%)	22
56	5154 (94.6%)	294 (5.4%)	23
64	5152 (94.7%)	287 (5.3%)	32

TABLE VI

RESULTS OF RANDPS FOR N.CRASSA.

Genome	N.crassa		
Length	17,484,362		
# of genes	10,633		
Probe length	Number of genes requiring		
	1 probe	2 probes	no probe
24	10530 (99.2%)	90 (0.8%)	13
32	10551 (99.5%)	57 (0.5%)	25
40	10557 (99.5%)	50 (0.5%)	26
48	10558 (99.6%)	45 (0.4%)	30
56	10559 (99.6%)	42 (0.4%)	32
64	10544 (99.6%)	40 (0.4%)	49

TABLE VII

RESULTS OF RANDPS FOR A.THALIANA.

Genome	A.thaliana		
Length	33,581,216		
# of genes	26,186		
Probe length	Number of genes requiring		
	1 probe	2 probes	no probe
24	22407 (85.6%)	3773 (14.4%)	6
32	24400 (93.2%)	1777 (6.8%)	9
40	24813 (94.8%)	1358 (5.2%)	15
48	25094 (95.9%)	1063 (4.1%)	29
56	25238 (96.5%)	910 (3.5%)	38
64	25327 (96.9%)	807 (3.1%)	52

In our experiments, we have noticed that there are some genes with no probes. An investigation of these genes revealed that some of these genes are duplicated or very similar to some other genes in the genome. Another reason is that the lengths of some of these genes are too short. Apart from these cases, our algorithm is able to select probes for all genes.

IV. CONCLUSION

We have proposed a new approach to select (randomly) a small set of probes and demonstrated that such a small set of probes is sufficient to distinguish each gene from all the

TABLE VIII

RESULTS OF RANDPS FOR MOUSE CHROMOSOME 2.

Genome	Mouse chromosome 2		
Length	182,887,278		
# of genes	1,302		
Probe length	Number of genes requiring		
	1 probe	2 probes	no probe
24	1194 (91.7%)	108 (8.3%)	0
32	1229 (94.4%)	73 (5.6%)	0
40	1231 (94.5%)	71 (5.5%)	0
48	1235 (94.9%)	67 (5.1%)	0
56	1239 (95.2%)	63 (4.8%)	0
64	1240 (95.2%)	62 (4.8%)	0

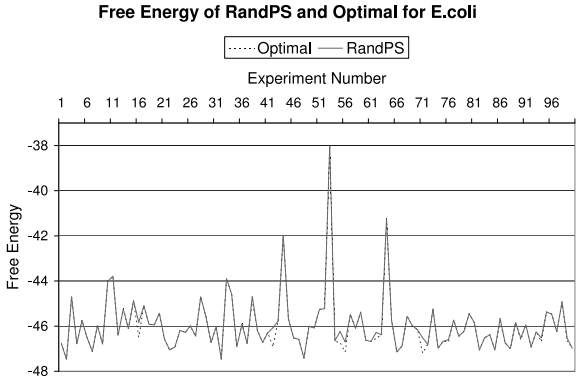

Fig. 2. Comparison of free energy between the optimal probe and the probe chosen by RANDPS on E.coli.

other genes in the genome. Almost all genes can be identified by a unique probe, the others need at most two probes. As we have shown, our method is suitable for large scale datasets and it relies on relatively large length of probes. Moreover, our approach should prove to be useful in the design of a fault-tolerant system of multiple probes, to accommodate common lack of accuracy characterising wetlab experiments. Based on our algorithm, we have also implemented a probe selection procedure RANDPS, which runs efficiently and effectively. The software will be available on request. In further research, we will focus on identification and classification of multiple genes by a system of probes.

V. ACKNOWLEDGMENTS

We are very grateful to Knut Reinert who brought the probe selection problem to our attention. We thank David Peleg for very useful comments on probability aspects of this work as well as David Mount and Christine Heitsch for providing very useful comments on melting temperatures and other very useful information on the subject. We would also like to thank Mia Persson for helpful discussions in the initial stages of this work.

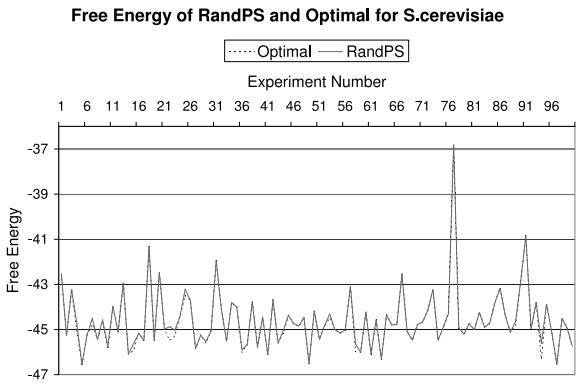

Fig. 3. Comparison of free energy between the optimal probe and the probe chosen by RANDPS on S.cerevisiae.

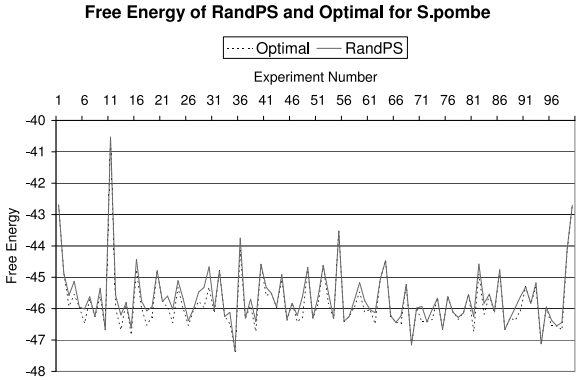

Fig. 4. Comparison of free energy between the optimal probe and the probe chosen by RANDPS on S.pombe.

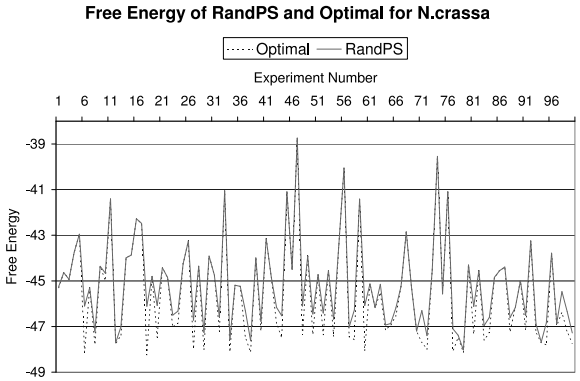

Fig. 5. Comparison of free energy between the optimal probe and the probe chosen by RANDPS on N.crassa.

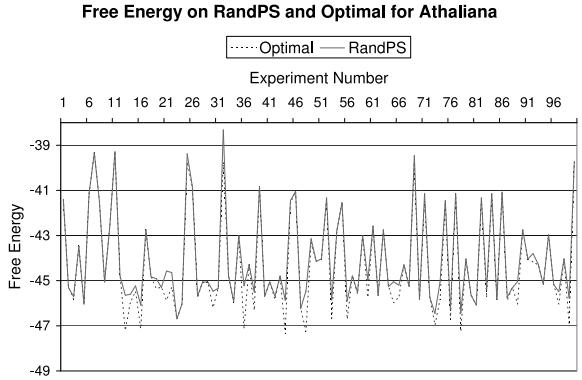

Fig. 6. Comparison of free energy between the optimal probe and the probe chosen by RANDPS on A.thaliana.

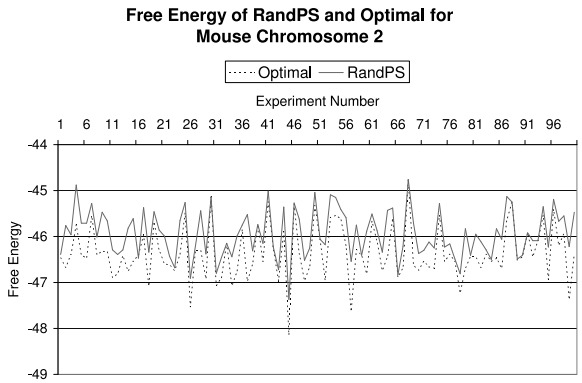

Fig. 7. Comparison of free energy between the optimal probe and the probe chosen by RANDPS on Mouse Chromosome 2.

REFERENCES

[1] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids. Research*, 25(17):3389–402, 1997.

[2] Z. Bozdech, J. Zhu, M. P. Joachimiak, F. E. Cohen, B. Pulliam, and J. L. DeRisi. Expression profiling of the schizont and trophozoite stages of *plasmodium falciparum* with a long-oligonucleotide microarray. *Genome Biology*, 4:R9, 2003.

[3] Peter J. Cameron. *Combinatorics : Topics, Techniques, Algorithms.* Cambridge University Press, 1994.

[4] C. R. Cantor and P. R. Schimmel. *Biophysical Chemistry Part III: The Behavior of Biological Macromolecules.* W. H. Freeman, San Francisco, CA, 1980.

[5] A. G. Frutos, Q. Liu, A. J. Thiel, A. M. W. Sanner, A. E. Condon, L. M. Smith, and R. M. Corn. Demonstration of a word design strategy for DNA computing on surfaces. *Nucleic Acids. Research*, 25(23):4748–4757, 1997.

[6] D. Gerhold, T. Rushmore, and C. T. Caskey. DNA chips: promising toys have become powerful tools. *Trends Biochem. Sci*, 24(5):168–173, 1999.

[7] R. W. Hamming. Error-detecting and error-correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[8] I. L. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids. Research*, 31(13):3429–3431, 2003.

[9] L. Kaderali and A. Schliep. Selecting signature oligonucleotides to identify organisms using DNA arrays. *Bioinformatics*, 18:1340–1349, 2002.

[10] G. W. Klau, S. Rahmann, A. Schliep, M. Vingron, and K. Reinert. Optimal robust non-unique probe selection using Integer Linear Programming. *Bioinformatics*, 20:i186–i193, 2004.

[11] Z. J. Lempel. A universal algorithm for sequential data compression. *IEEE Trans Inf Theory*, 23:337–343, 1977.

[12] F. Li and G. Stormo. Selection of optimal DNA oligos for gene expression arrays. *Bioinformatics*, 17(11):1067–1076, 2001.

[13] M. Li, H. J. Lee, A. E. Condon, and R. M. Corn. DNA word design strategy for creating sets of non-interacting sets of oligonucleotides for DNA microarrays. *Langmuir*, 18(3):805–812, 2002.

[14] D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. L. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14:1675–1680, 1996.

[15] O. V. Matveeva, S. A. Shabalina, V. A. Nemtsov, A. D. Tsodikov, R. F. Gesteland, and J. F. Atkins. Thermodynamic calculation and statistical correlations for oligo-probes design. *Nucleic Acids. Research*, 31(14):4211–4217, 2003.

[16] E. W. Myers. A fast bit-vector algorithm for approximate string matching based on dynamic programming. In *Proceedings of the Ninth Annual Symposim on Combinatorial Pattern Matching*, pages 1–13, 1998.

[17] F. Naef and M. O. Magnasco. Solving the riddle of the bright mismatches: Labeling and effective binding in oligonucleotide arrays. *Physical Review E*, 68:011906, 2003.

[18] S. Rahmann. Rapid large-scale oligonucleotide selection for microarrays. In *Proceedings of the First Computational Systems Bioinformatics*, pages 54–63, 2002.

[19] A. Relogio, C. Schwager, A. Richter, W. Ansorge, and J. Valcarcel. Optimization of oligonucleotide-based DNA microarrays. *Nucleic Acids. Research*, 30(11):e51, 2002.

[20] J-M. Rouillard, M. Zuker, and E. Gulari. OligoArray2.0: design of oligonucleotide probes for DNA microarrays using a thermodynamic approach. *Nucleic Acids. Research*, 31(12):3057–3062, 2003.

[21] W. Rychlik, W. J. Spencer, and R. E. Rhoads. Optimization of the annealing temperature for DNA amplification in vitro. *Nucleic Acids. Research*, 18(21):6409–6412, 1990.

[22] J. J. SantaLucia, H. T. Allawi, and P. A. Seneviratne. Improved nearest-neighbor parameters for predicting DNA duplex stability. *Biochemistry*, 35(11):3555–3562, 1996.

[23] D. K. Slonim, P. Tamayo, J. P. Mesirov, T. R. Golub, and E. S. Lander. Class prediction and discovery using gene expression data. In *Proceedings of the Fourth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 263–272, 2000.

[24] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[25] W. K. Sung and W. H. Lee. Fast and accurate probe selection algorithm for large genomes. In *Proceedings of the Second Computational Systems Bioinformatics*, pages 65–74, 2003.

[26] A. C. Tolonen, D. F. Albeanu, J. F. Corbett, and H. Handley. Optimized *in situ* construction of oligomers on an array surface. *Nucleic Acids. Research*, 30(20):e107, 2002.

[27] J. G. Wetmur. DNA probes: applications of the principles of nucleic acid hybridization. *Crit Rev Biochem Mol Biol*, 26(3-4):227–59, 1991.

[28] M. A. Wright and G. M. Church. An open-source oligomicroarray standard for human and mouse. *Nature Biotechnology*, 20:1082–1083, 2002.

[29] Z. Wu and R. A. Irizarry. Stochastic models inspired by hybridization theory for short oligonucleotide microarrays. In *Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 98–105, 2004.

[30] M. Zuker, D. H. Mathews, and D. H. Turner. *Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide.* NATO ASI Series, Kluwer Academic publishers, Dordrecht, NL, 1999.