

Fast and Accurate Probe Selection Algorithm for Large Genomes

Wing-Kin Sung

Department of Computer Science
National University of Singapore
Singapore
ksung@comp.nus.edu.sg

Wah-Heng Lee

Genome Institute of Singapore
Singapore

Abstract

The oligo microarray (DNA chip) technology in recent years has a significant impact on genomic study. Many fields such as gene discovery, drug discovery, toxicological research and disease diagnosis, will certainly benefit from its use. A microarray is an orderly arrangement of thousands of DNA fragments where each DNA fragment is a probe (or a fingerprint) of a gene/cDNA. It is important that each probe must uniquely associate with a particular gene/cDNA. Otherwise, the performance of the microarray will be affected. Existing algorithms usually select probes using the criteria of homogeneity, sensitivity, and specificity. Moreover, they improve efficiency employing some heuristics. Such approaches reduce the accuracy. Instead, we make use of some smart filtering techniques to avoid redundant computation while maintaining the accuracy. Based on the new algorithm, optimal short (20 bases) or long (50 or 70 bases) probes can be computed efficiently for large genomes.

1 Introduction

Rapid developments in bio-technology have uncovered the complete sequences of numerous genomes. Many more genomes are to be mapped out in the near future. However, knowing the sequences of the genomes is only the start. In the pro-genome era, we would like to study the activities of genes within a cell. More precisely, we hope to measure the amount of mRNAs transcribed from every gene (which is called the gene expression level of the gene).

Traditional methods in molecular biology generally work on a “one gene in one experiment” basis, which means that the throughput is very limited and the “whole picture” of gene function is hard to obtain. Currently, DNA microarray is used to analyze gene expressions. DNA microarray or DNA chip is a glass or nylon slide containing a set of spots where each spot contains identical short DNA sequences

known as probes. Each probe is a substring of a gene, which acts as its fingerprint. To analyze the genes’ activities within a cell, the mRNAs are extracted and transcribed into cDNAs. Then the cDNAs are fluorescent-labelled and introduced to the chip. Due to Watson-Crick base pairing, the cDNA representing a particular gene will bind (or hybridize) to the corresponding probe. The cDNAs which fail to hybridize are washed off the chip. Then, the cDNAs’ expression levels are measured based on the fluorescent level of each spot [9].

DNA microarray technology is able to monitor the whole genome on a single chip so that researchers can have a better picture of the interactions among thousands of genes simultaneously. Thus there is a dramatic increase in knowledge of regulatory expression patterns and throughput. This makes the DNA microarray technology invaluable to studies in genomics and medicine. Some of its important uses are discussed below.

Gene Identification: Cells of different tissues have specific phenotypes and functionality. Microarray technology using cDNAs can measure the levels and patterns of cell and tissue-specific gene expression important in growth, metabolism, development, behavior and adaptation of living systems [1]. A cluster analysis of microarray data is used to find the genes which are preferentially expressed in specific tissues. These genes are important to research because they serve the functionality of the cell type and control genes that ensure the cell only performs the functions of its specific type.

Genetic Diseases Research: Genes which are not transcribed properly often result in genetic diseases. These genes may have too much or too little expression. These defects occur in cancer where often, regulatory genes are deleted, inactivated, or become constantly active. Cancer can be caused by several independent gene regulatory defects. The causes for cancer differ from patient to patient thus making their identification highly complicated. Since

microarrays technology can identify the change in transcription levels and gene expression patterns in diseased states, it has become invaluable for cancer investigation [3]. Microarrays can also be used to find genes that are linked to diseases or increase the susceptibility to inherited diseases. In a study, scientists at the Rockefeller University used microarray technology to detect Single Nucleotide Polymorphism (SNP) of the human MU opioid receptor gene. This was done by hybridization or single nucleotide extension on an oligonucleotide gelpad microchip. The group states that the microchips have a lot of potential for SNP studies [6]. The identification of specific gene alterations associated with a disease can identify candidate targets for therapeutic intervention, using specifically designed drugs.

Cell Cycle Variation: As cells undergo their cycle of life, there will be DNA replication, mitosis and eventually death. These processes all require different gene products, i.e., microtubule spindle proteins. The cell will have genes that control these cell cycle events. Microarray analysis of expression patterns at various time intervals will help elucidate the genes that are responsible for cell cycle regulation [8].

Drug Discovery and Toxicological Research: Microarray technology is exploited by pharmaceutical industries. The presence of alternate forms of a gene or unusual expression of a gene can result in resistance to chemotherapy. Various types of studies may be able to correlate the genetic profile of individual patients and the individual response to various drugs or toxins. DNA microarrays can assist in the identification of certain genes that demonstrate altered expression in a given cell or tissue type in response to drug or toxin exposure [7]. The information obtained from these studies can be used to assist in the selection of custom and individually tailored drug therapy to treat disease.

Forensic Identification: DNA samples can be used by forensic scientists to identify individuals. Scientists use microarrays to find the markers in a DNA sample by designing small pieces of DNA (probes) that will each seek out and bind to a complementary DNA sequence in the sample [4]. A series of probes bound to a DNA sample creates a distinctive pattern for an individual. The data from DNA samples of an individual is then used to create a DNA profile of that individual (sometimes called a DNA fingerprint). These DNA fingerprints can then be used to identify potential suspects whose DNA may match evidence left at crime scenes or to exonerate persons wrongly accused of crimes.

In the past, a probe for a gene is simply its random substring. However, such approach cannot guarantee that every probe does not cross-hybridize, that is, it cannot ensure ev-

ery probe can bind or hybridize with the target gene only. Other factors such as the secondary structure and melting temperature of probes may also cause hybridization errors. The use of such microarrays reduces the experiment accuracy. Accuracy is very important as a probe set once found is used for thousands of experiments. Therefore, it is challenging and important to have some specialized algorithm which can accurately select "good" probe for every gene.

1.1 Previous Works

Research in better algorithms for probe design has been on going for some time due to the need for better quality microarrays. Lockhart et al [15] contributed the first program for designing probes in 1996. This program is used by Affymetrix to design short probes of length 20 to 25. They did not publish their algorithm. Li and Stormo [14] proposed a heuristic algorithm to solve the probe design problem. To improve efficiency, their algorithm uses advance data structure suffix array [16] and fast pattern matching program myersgrep [17]. However, the algorithm is still not fast enough for the computation of large genome sets. It took almost four days to design a length-24 probe set for *Saccharomyces cerevisiae* genome, which is about 9.5×10^6 bps and includes 6343 genes. Rouillard, Herbert and Zuker [20] suggested to use BLAST to avoid cross-hybridization and use Mfold program to avoid secondary structure. Their algorithm is more efficient. Within a day, their algorithm can design a length-50 probe set for the *Saccharomyces cerevisiae* genome. Kaderali and Schliep [10] attempted to design an accurate probe set. By utilizing heuristic dynamic programming, they try to compute the most stable alignment between every probe and every sequence. Although their solution has higher accuracy, the algorithm is very slow and is unsuitable for large genomes. It takes 9 hours to design a probe set for 58 HIV-1 subtypes of total length 600×10^3 . Lipson, Webb and Yakhini [13] presented an algorithm that computes the exact specificity of a probe by using its longest common contiguous substring. To find probes of length l , it states that if two strings s and t have a Hamming distance less or equal to d , then there must be at least one substring of s , length $\geq \lceil \frac{l-d}{d+1} \rceil$, which is a perfect match to the corresponding substring of t . This approach is slow as it took 46 minutes to find length-30 probes for a length-300 transcript of *S. cerevisiae*. Recently, Rahmann [19] presented a fast algorithm that is practical for designing short probes up to 30 nucleotides. He approximates the unspecificity of a probe by computing its longest common contiguous substring, thus, greatly improves the efficiency. Now, it is able to design probes for large genome like *Neurospora crassa* in 4 hours. However, his approach can only design short probes. Furthermore, his approximation is loose and some good probes may miss out.

1.2 Our Result

Our algorithm selects good probes based on the criteria of homogeneity, sensitivity and specificity as proposed by Lockhart [15]. The homogeneity filter eliminates probes that hybridize at a temperature that is out of the experiment temperature range while the sensitivity filter eliminates probes with secondary structures. This ensures that the probes are able to hybridize with their intended targets at the experiment temperature.

Specificity filter eliminates probes that cross-hybridize. This step is very computational intensive and takes up the most time in probe design programs. However, by the use of the Pigeon Hole Principle, we sped up the specificity filter greatly. Our algorithm only finds and checks exact regions in the genome that potentially cause cross-hybridization. Since these regions are small compared to the entire genome, we avoid redundant checks. Most importantly, our approach is not a heuristical approach and thus is able to filter all “bad” probes.

Oligonucleotide expression arrays consist of short oligo arrays of 20 bases (Affymetrix geneChip) and long oligo arrays of 50 [11] or 70 bases [5]. Our algorithm can select probes efficiently for both short and particularly long oligos for very large genomes. In one experiment, 173 genes with a total length of 3.5×10^6 bps were compared with the 1.4×10^9 bps human genome. The length-60 probe set for the 173 genes was generated in 21 hours. For more commonly tested genomes, Table 1 summarizes the relative performance of our algorithm with the algorithms mentioned in Section 1.1.

The experimental results show that our algorithm is much faster than existing algorithms. For E.coli, even though 50-mers take a longer time to compute than 23-mers, our algorithm is 830 times faster than Li and Stormo’s algorithm. For *S. cerevisiae*, our algorithm finds 50-mers 43 times faster than the algorithm by Rouillard, Herbert and Zuker. In large genomes such as *Neurospora crassa*, our algorithm performs about 1.7 times faster than Rahmann’s algorithm. Most importantly, unlike other algorithms which sacrifice accuracy for speed by the use of heuristics, our non-heuristical algorithm is able to achieve both speed and accuracy. Thus, we can get a more reliable probe set in a much shorter time. Currently, the production of a microarray is a long and tedious process of lab testing to find the probes unique to each gene. Our algorithm can greatly reduce the time scientists spend on lab-testing each gene for probes and increase microarray throughput. This will eventually lead to a faster production of more accurate microarrays which will no doubt be invaluable in the fight against diseases and genomics research.

We have developed a program *FindProbe* which makes use of our algorithm to select probes for genomes. The pro-

gram software has been accepted and is currently used by the *Genome Institute of Singapore*.

2 Probe Design Problem

This section first formally defines the probe design problem. Then, the framework of our algorithm *FindProbe* is presented.

2.1 Definition

Given a set of genes $G = \{g_1, g_2, \dots, g_n\}$ and a parameter m which specifies the length of the probes, the probe design problem finds, for every gene g_i , a length- m probe (that is a substring of g_i) which satisfies (1) Homogeneity, (2) Sensitivity and (3) Specificity.

Homogeneity: Temperature is one of the important experiment conditions to ensure a probe can hybridize. We select probes whose melting temperature are close to the experiment temperature. CG rich sequences are susceptible to non-specific interactions that may reduce reaction efficiency. Thus, the CG content of good probes should not be too high or too low [18].

Sensitivity: Sensitivity, the ability to detect low-abundance mRNAs, is a key performance feature of microarrays. Probes that form significant secondary structures jeopardize sensitivity. Thus it is important to reject probes with high self-complementariness and select probes with minimal secondary structure. To do this, the free energy for each probe is computed based on the nearest-neighbor model [2]. The free energy for each probe should be as high as possible.

Specificity: Specificity identifies probes that are unique to each gene in the genome. This condition minimizes cross-hybridization of the probes with other DNA sequences. A length- m substring s_a of a gene g_a is defined to be specific to the gene if the distance between the segment, s_a , and any length- m segment s_b from gene, g_b , where ($g_a \neq g_b$), does not exceed some pre-specified limit under some predefined distance measurement. Our program uses the Hamming distance as the similarity measurement. 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. Thus, if the Hamming distance between a probe and every subsequence in the genome is greater than some constant, the probe is said to be specific enough.

2.2 Algorithm FindProbe

This paper proposes a new algorithm *FindProbe* to solve the probe design problem.

The algorithm is based on probe elimination. Initially, we assume for every gene g_i , every length- m substring of

| | Li and Stormo BIBE, 2000 | Rouillard, Herbert and Zuker Bioinformatics, 2002 | Rahmann WABI, 2002 | Our algorithm |
|--------------------------|-----------------------------|--|-----------------------|-----------------------|
| <i>E. coli</i> | 23-mers 1.5 days | | | 50-mer 3.1 minutes |
| <i>S. cerevisiae</i> | 24-mers 4 days | 50-mers 1 day | | 50-mers 49 minutes |
| <i>Neurospora crassa</i> | | | 25-mers 4 hours | 50-mers 3.5 hours |

Table 1. Comparison between our algorithm and other algorithms

g_i is a feasible probe. “Bad” probes are filtered out using the following 3 steps:

1. Filter oligo probes in the genome which fail to satisfy homogeneity criterion.
2. Filter oligo probes in the genome which fail to satisfy sensitivity criterion.
3. Filter oligo probes based on the specificity criterion using the Pigeon-Hole Principle. This is to remove probes that can cross-hybridize with any of the sub-sequences in the whole genome.

The rest of the paper details Steps 1 to 3 in Sections 3 to 5 respectively.

3 Homogeneity Filtering

Homogeneity criterion requires the melting temperature for every probe should be within some pre-defined range. This is important because probes in a good probe set need to hybridize with their intended target at about the same temperature. In our algorithm, the melting temperature T_m for a length- m probe p is computed based on the nearest-neighbor thermodynamic parameters [21] and given by

$$T_m(p) = \frac{\Delta H(p)}{\Delta S(p) + R \times \log(C_T)} - 273.15 + 16.6 \log(Na^+)$$

where R is the molar gas constant ($1.987 \text{ cal}/^\circ\text{C} \times \text{mol}$) and C_T is the total molar concentration of the annealing oligonucleotides when oligonucleotides are self-complementary. For non-self-complementary oligonucleotides, C_T is replaced by $\frac{C_T}{4}$. Na^+ is the salt concentration of the solution in which the oligomers are dissolved. $\Delta H(p)$ and $\Delta S(p)$ are the enthalpy and entropy for helix formation of p respectively.

Subsequently, the optimal hybridization temperature T_h of p is determined from $T_m(p)$ by

$$T_h(p) = T_m(p) - 25 - 0.62(C_F)$$

where C_F is the formamide concentration of the solution [12].

In addition, the content of any single base in a probe should not exceed 50% [18]. Formally, we should keep $r_1\% \leq C + G \text{ content} \leq r_2\%$ where r_1 and r_2 are user defined percentage ranges. Typically, we set $r_1 = 40$ and $r_2 = 60$.

Since the computation of melting temperature, hybridization temperature and CG content requires only a single pass of the probe sequence, the homogeneity of a probe can be determined in linear time.

4 Sensitivity Filtering

Sensitivity filter eliminates probes that form secondary structures. We use a simplified secondary structure prediction algorithm to determine whether a probe can form secondary structures. Taking a segment of length x from the 3' end of each probe, if it can form a consecutive length y complementary segment with itself, it is said to have a secondary structure. We eliminate such probes. This is shown in Figure 1.

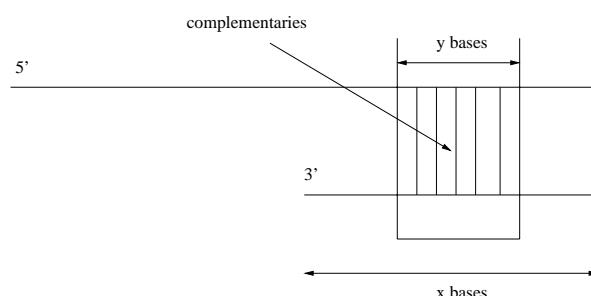


Figure 1. Diagram showing how a gene sub-sequence form a secondary structure

In this filter, we want to eliminate probes which are able to fold back on itself, thus forming a U-shaped structure as shown in Figure 1. x specifies the length from 3' end of each probe to check for complementaries. A higher x value results in a more stringent sensitivity filter as we are checking a larger portion from the 3' end of each probe for complementaries. y specifies the number of consecutive complementaries allowed in the length- x 3' end of each probe.

Thus, the lower the value of y , the more stringent is the filter.

For every probe, from within the length- x 3' end, checking that there are not more than y consecutive complementaries with the 5' end of the probe requires only a single pass of the probe sequence. Since this secondary structure prediction algorithm scan through the probe sequence only once, it runs in linear time.

5 Specificity Filtering

Specificity minimizes cross-hybridization of probes with other DNA sequences. For each length- m probe p in gene g , we aim to find out whether there exists a length- m substring q in $G - \{g\}$ such that $H(p, q) < \text{some threshold } w$. If such a q is found, then the probe p is said to be able to cross-hybridize with other genes and thus, it is not a "good" probe. The aim of the specificity filter is to filter out all "bad" probes.

The brute force approach scans through the whole length- n genome for every length- m probe and determine if the Hamming distances are big enough. Such process is slow and it takes $O(mn^2)$ time. For example, specificity filtering would take 72 hours for *S. pombe* genome of length 7.1×10^6 bps and thus impractical for large genomes.

In this section, an approach is described to do specificity filtering. The approach makes use of the Pigeon Hole Principle to speed up the searching process. Note that unlike the filters described in [14] or [19], our specificity filter can filter out all "bad" probes since we do not take advantage of any heuristics.

5.1 Basic Algorithm based on Consecutive Hashing

This section presents an algorithm that makes use of the Pigeon Hole Principle to determine whether a probe will cross-hybridize. Below is the key lemma.

Lemma 5.1 *If there exist length- m probes p and q such that $H(p, q) < w$, then we can find length- k substrings $p' = p[i..i+k-1]$ and $q' = q[i..i+k-1]$ such that $H(p', q') \leq v$ where $v = \lfloor (w-1) \frac{k}{m} \rfloor$.*

Proof. Given p and q where $H(p, q) < w$, there are at most $(w-1)$ mismatches between p and q . These mismatches are distributed across $\frac{m}{k}$ pairs of length- k substrings ($p[ik..(i+1)k-1]$, $q[ik..(i+1)k-1]$) in p and q , for $i = 0, 1, \dots, \frac{m}{k}$. By pigeon hole principle, at least one pair of the length- k substrings has at most $v = \lfloor \frac{w-1}{m/k} \rfloor$ mismatches. The lemma follows. \square

For any two length- k substrings s_1 and s_2 in the genome, we say they form a hit if $H(s_1, s_2) \leq v$ where $v = \lfloor (w -$

$1) \frac{k}{m} \rfloor$. For an arbitrary chosen length- k substring s in a genome, the set of hits for s can be found by enumerating all substrings in the genome that has Hamming distance $\leq v$ with s . We do the same procedure for each and every length- k substring in the genome to obtain all hits in the genome. By Lemma 5.1, if there is no hit between a length- m probe p of gene g and any length- m substring q of gene $g' \in G - \{g\}$, $H(p, q) \geq w$. Then, p is a "good" probe since it cannot cross-hybridize. Otherwise, p is a "bad" probe. Based on this idea, "bad" probes can be filtered out using the basic algorithm in Figure 2.

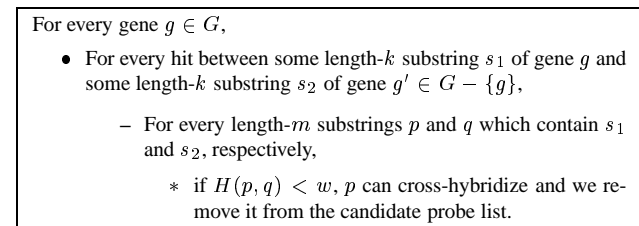


Figure 2. Basic algorithm for specificity filter

In the worst case, the time complexity of the basic algorithm is as bad as the brute force approach. Moreover, since the genome sequence looks random, its performance is quite good in practise. Below, with the assumption that the genome sequence looks random, we analysis the average complexity of the basic algorithm.

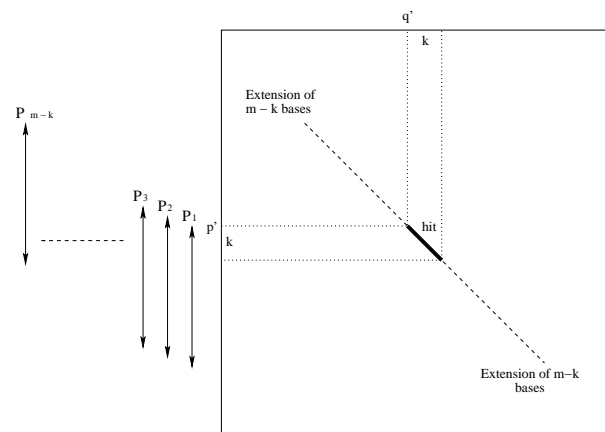


Figure 3. Diagram showing a consecutive hit and how extension is done. The bold line represents a length- k hit. We extend the line $m - k$ bases downwards to get the Hamming distance of P_1 . The Hamming distance of P_2 can be obtained by shifting the line upwards by 1 base and so on.

Observe that the probability of getting a length- k hit is very small. It equals $\frac{1}{4^k} \sum_{i=0}^v \binom{k}{i} 3^i$. Thus, the expected

number of hits in the genome is $\frac{1}{4^k} [\sum_{i=0}^v \binom{k}{i} 3^i] n^2$. Each

hit can appear in $m - k$ different adjacent probe pairs. The Hamming distance of the initial probe pair (P_1, Q_1) of each hit (p', q') is computed by doing $m - k$ extensions downwards. Formally, $H(P_1, Q_1) = H(p', q') + H(P_1[k], Q_1[k]) + \dots + H(P_1[m - 1], Q_1[m - 1])$. Subsequently, we need to do at most $m - k$ extensions upwards from the initial probe pair of each hit, in order to get the Hamming distance of the adjacent $m - k$ probes with the substrings in the region of the hit. That is, $H(P_n, Q_n) = H(P_{n-1}, Q_{n-1}) - H(P_{n-1}[m - 1], Q_{n-1}[m - 1]) + H(P_n[0], Q_n[0])$. This is shown in Figure 3. Based on the Hamming distance obtained by comparing substrings in the region of hits, the specificity of the probe can be determined. Thus the complexity of the algorithm is

$$O\left(\frac{1}{4^k} \left[\sum_{i=0}^v \binom{k}{i} 3^i\right] [n^2(2m - 2k)]\right).$$

5.2 Improved Algorithm based on Gapped Hashing

An improvement to the algorithm can be made if we change the consecutive hit described in Section 5.1 to a gapped hit. Lemma 5.1 can be extended as follows:

Lemma 5.2 *If there exist length- m probes p and q such that $H(p, q) < w$, then we can find length- k substrings $p' = p[i]p[i + \frac{m}{k}]p[i + 2\frac{m}{k}] \dots p[i + (k-1)\frac{m}{k}]$ and $q' = q[i]q[i + \frac{m}{k}]q[i + 2\frac{m}{k}] \dots q[i + (k-1)\frac{m}{k}]$ for $i = 0, 1, \dots, \frac{m}{k} - 1$ such that $H(p', q') \leq v$ where $v = \lfloor (w - 1)\frac{k}{m} \rfloor$.*

Proof. Given p and q where $H(p, q) < w$, there are at most $(w - 1)$ mismatches between p and q . These mismatches are distributed across $\frac{m}{k}$ pairs of length- k substrings $p' = p[i]p[i + \frac{m}{k}]p[i + 2\frac{m}{k}] \dots p[i + (k-1)\frac{m}{k}]$, $q' = q[i]q[i + \frac{m}{k}]q[i + 2\frac{m}{k}] \dots q[i + (k-1)\frac{m}{k}]$ in p and q , for $i = 0, 1, \dots, \frac{m}{k} - 1$. By pigeon hole principle, at least one pair of the length- k substrings has at most $v = \lfloor \frac{w-1}{m/k} \rfloor$ mismatches. The lemma follows. \square

By using gapped hashing, each hit only appears in $\frac{m}{k}$ different probe pairs, instead of $m - k$ different probe pairs. Thus, we need to do at most $\frac{m}{k}$ extensions for each hit. This is shown in Figure 4. The complexity of the algorithm improves by a constant factor of at most $\frac{1}{2}$.

A further improvement to the algorithm can be done by using preprocessing. We pre-compute the hamming distance between any two length- α substrings and store them

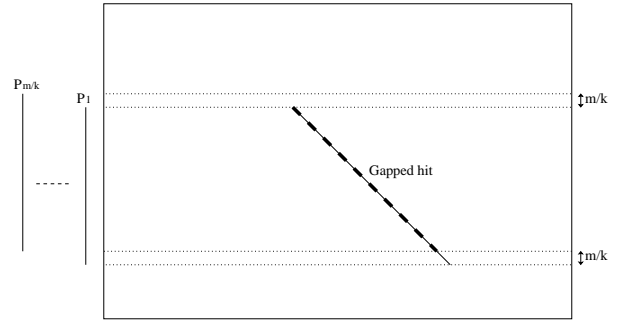


Figure 4. Diagram showing a gapped hit and how extension is done.

in a table. Since we need to pre-compute hamming distances for a table of 4^α mappings, the choice of α is dependent on the memory size and thus set to a value between 10 to 12. Pre-computation is done by first mapping each gene base into binary numbers as follows: $A \rightarrow 00$, $C \rightarrow 01$, $G \rightarrow 10$, $T \rightarrow 11$. This mapping is then used to map two length- m probes p and q into their binary representation p' and q' respectively. Given two gene bases b_1 and b_2 , observe that $b_1 \neq b_2$ iff $\oplus(b_1, b_2) \neq 00$. Using this property, we can find the number of mismatches between p and q by counting the number of 00 in $\oplus(p', q')$. This takes $O(\frac{m}{\alpha})$ time. Then, based on table lookup, the hamming distance between two length- m probes can be determined in $O(\frac{m}{\alpha})$ time instead of $O(m)$ time. Thus, the complexity of the algorithm improves by a factor of $\frac{1}{\alpha}$.

In summary, this section described improvements to the original algorithm based on consecutive hashing. Through gapped hashing, we reduced the time complexity from $O(\frac{1}{4^k} [\sum_{i=0}^v \binom{k}{i} 3^i] [n^2(2m - 2k)])$ to $O(\frac{1}{4^k} [\sum_{i=0}^v \binom{k}{i} 3^i] [n^2(m - k + \frac{m}{k})])$. Subsequently though preprocessing, we reduced the time complexity again to $O(\frac{1}{4^k} [\sum_{i=0}^v \binom{k}{i} 3^i] [n^2(\frac{m}{\alpha} + \frac{m}{k})])$.

6 Experimental Results

Our program is written in C and was developed and tested on SunFire Workstations (700 MHz) with 4GB memory. Inputs of the programs are FASTA formatted nucleotide sequences.

Vigorous testing has been performed on FindProbe using numerous genomes obtained from reliable genome databases. Here, we present a set of experimental results based on a few genomes that have been widely used for test-

ing purposes by other probe design algorithms. In this way, we can effectively compare our algorithm with other algorithms described in Section 1.1. The genomes involved in these two sets of experimental results presented are:

- *Escherichia coli* of length 4.6×10^6 bps
- *Schizosaccharomyces pombe* of length 7.1×10^6 bps
- *Saccharomyces cerevisiae* of length 8.9×10^6 bps
- *Neurospora crassa* of length 3.8×10^7 bps

The set of experiment results give the length-50 probe set for the set of genomes listed above. This experiment was done to evaluate the capability of our program at finding long oligo probes. It took about 3 minutes to finish the *E.coli* genome. It took about 29 minutes to finish the *S. pombe* genome which includes 4987 genes. The *S. cerevisiae* genome which has 6343 genes took 49 minutes to complete. The *Neurospora crassa* genome took about 214 minutes to complete. The size of the probe is 50 bases. The full details of the experiments are shown in Table 2.

FindProbe is a flexible and modular program. It is able to select probes for very large genomes by using a divide and conquer approach. A large genome is partitioned into multiple smaller sized partial genomes and filtering is performed on each of these partial genomes in sequential order. Using this approach, FindProbe was able to generate probes for genes with respect to the 1.4×10^9 bps human genome.

We present two such experiments to select probes for genes requested by the Genome Institute of Singapore. The first experiment involves the 2119 bps *HoxA9*, a novel breast cancer progression gene that resides in Chromosome 7 of the human genome. The probe selection process for *HoxA9* gene took about 5 hours. The second experiment involves tumour suppressor genes that are thought to be involved in Nasopharyngeal Carcinoma that resides in Chromosome 11 in the regions 11q13 and 11q22 of the human genome. There are 173 genes and the length of the genes for probe selection is 3.5×10^6 bps. The probe selection for genes involved in Nasopharyngeal Carcinoma took about 14 hours. Table 3 shows the full results of the two experiments.

As shown in Table 3, the probe selection process for *HoxA9* gene took about 5 hours. 17 probes unique to *HoxA9* gene was selected by our program. In the second experiment, the probe selection for genes involved in Nasopharyngeal Carcinoma took about 14 hours. A total of 1162 probes were selected for the 173 genes.

7 Probes Generated by FindProbe

In this section, we highlight the characteristics of the probes selected for some of the genes by our program. Here is the legend for the tables:

BP: Position of the probe from the gene it is extracted from.

CG: C + G content of the probe.

MT: Melting temperature of the probe in $^{\circ}C$.

HT: Hybridization temperature of the probe in $^{\circ}C$.

FE: Free energy of the probe in kcal/mol.

HD: Global minimum Hamming distance of the probe.

DE: Distance from 3' end.

We present the 50-mer probes selected for the $> pB10D8_100324| SPBP35G2.13c$ gene in the *S. pombe* genome by the program. Table 4 shows the details of these probes.

Our algorithm is able to find probes that are relatively unique to each gene with respect to the Hamming distance. The minimum acceptable Hamming distance for non-cross-hybridization is 15 mismatches for 50-mer probes. The results showed that the 50-mer probes, the probes selected have a global minimum mismatch significantly higher than 15 mismatches. This further reduces the risk of cross-hybridization with unintended targets. The CG-content of the probes selected are very near to the optimal of 50%. In addition, their hybridization temperatures are very close to the required experiment temperature of $42^{\circ}C$. They also have high free energy compared to the average free energy of all probes which is -70 kcal/mol. Thus, these probes are less likely to cause reaction inefficiency and hybridization errors. They are also close to the 3' end of their resident gene. These probes may be preferred because probes from the 3' ends of genes are more heavily labelled in oligo-dT primed labelling reactions.

Next, we present the probes selected by our program for the *HoxA9* gene. These probes are selected by the request of the Genome Institute of Singapore for their probe database. The details of the probes are shown in Table 5.

Table 5 shows that probes selected have at least a hamming distance of 27 with respect to the human genome. This is much greater than the minimum acceptable Hamming distance for 70-mer non-cross-hybridization which is 21 mismatches. The hybridization temperature of all these probes are about $42^{\circ}C$. This is a near ideal temperature for hybridization. Since a more positive free energy means that the probe is less self-complementary, probe number 1, 3, 4 and 5 are the better probes compared to probe number 2 in this aspect. The probes are also close to the 3' end since the largest distance from the 3' end of the probes is only 495 bases.

As a further illustration of our improvement in accuracy of the probe set, we compare our probes with a commercialized probe available in the market. This probe is probe number 2 in Table 5. It is a commercialized probe produced by Operon, the microarray oligo division of Qiagen, and used by the National Cancer Institute in Washington DC. In

| Genome Name | E. coli | S. pombe | S. cerevisiae | Neurospora crassa |
|---|---------|----------|---------------|-------------------|
| Genome Length (bases) | 4662239 | 7098029 | 8953158 | 38044343 |
| Number of Genes | 400 | 4987 | 6343 | 10895 |
| k | 10 | 11 | 11 | 11 |
| Time for Hashing (seconds) | 32 | 80 | 104 | 748 |
| Time for Homogeneity and Sensitivity Criteria Filtering (seconds) | 11 | 19 | 24 | 61 |
| Time for Cross Hybridization Criterion Filtering (seconds) | 60 | 1688 | 2857 | 12070 |
| Number of Genes with Probes Found | 400 | 4987 | 6343 | 10398 |
| Total Time for FindProbe (seconds) | 191 | 1787 | 2985 | 12879 |

Table 2. Experiment Results of FindProbe Program with the following parameters. The probe length was 50 bases. Lower bound of C+G content was 40%. Upper bound of C+G content was 60%. Sensitivity segment of each probe was 5 bases. Maximum secondary structure length of each probe was 3 bases. The melting temperature was between 80.0 degrees Celsius and 87.0 degrees Celsius.

| Gene Name | HoxA9 | Nasopharyngeal Carcinoma |
|---|-------|--------------------------|
| Genome Length (bases) | 2119 | 3548397 |
| Number of Genes | 1 | 173 |
| Probe Length | 70 | 60 |
| k | 11 | 11 |
| Total Time for Hashing (seconds) | 24975 | 32798 |
| Total Time for Homogeneity and Sensitivity Criteria Filtering (seconds) | 1 | 31 |
| Total Time for Uniformity Criterion Filtering (seconds) | 79 | 13807 |
| Total Time for Cross Hybridization Criterion Filtering (seconds) | 1582 | 4618 |
| Total Number of Probes Found | 17 | 1162 |
| Number of Genes with Probes Found | 1 | 173 |
| Total Time for FindProbe w/o Statistics(seconds) | 18124 | 51254 |

Table 3. Experiment results of finding probes for genes in human genome. These genes are compared against the 1.4×10^9 bps human genome. Lower bound of C+G content was 40%. Upper bound of C+G content was 60%. Sensitivity segment of each probe was 5 bases. Maximum secondary structure length of each probe was 3 bases. The melting temperature was between 77.0 degrees Celsius and 83.5 degrees Celsius.

comparison, all probes that we found are better than probe number 2 in terms of the statistic values. They hybridize closer to the experiment temperature, have better uniformity, have higher global Hamming distance and higher free energy. Thus our algorithm is able to find better quality probes and in larger quantity than other algorithms.

In our experiments, we have noted that there are some genes with no probes. An investigation of these genes revealed that some of these genes are duplicates or very similar to some other genes in the genome. These genes may have similar functions resulting in a similar genetic makeup. Another reason is that the length of some of these genes are even shorter than the probe length. Apart from these reasons, our algorithm is able to select probes for all genes.

8 Discussion

Microarray technology promises to revolutionize the way scientists examine gene expression. With its ability to analyze the expression of hundreds or thousands of genes

at the same time, the microarray promises to revolutionize the way scientists examine gene expression. The quality of a microarray is determined by the quality of the probes it uses. To realize the full potential microarray technology promises, the way at which probes used by microarrays are selected is crucial.

We presented a new algorithm to select oligo probes for microarrays. Our algorithm makes use of several smart filtering techniques to reduce the search space for probes. Our homogeneity and sensitivity filter eliminates probes with rich CG-content, extreme melting temperatures and secondary structures. Hamming distance is used as the specificity measure of the oligos. By using the Pigeon Hole Principle, we avoided redundant comparisons for probes and greatly reduced the time complexity of specificity filtering.

Due to the smart filtering techniques described above, our algorithm is capable of finding oligo probes for large genomes. For genomes whose size exceeds the memory limit of a computer, we use the divide-and-conquer approach for the selection of probes of such genomes. Thus,

| Probe Sequence | BP | CG | MT | HT | FE | HD | DE |
|--|-----|------|-------|-------|--------|----|-----|
| GACGACTACGACTTAAACGGAATGC TGCTAAGCATCGCCAGTTACGTGAG | 622 | 50.0 | 82.06 | 41.56 | -68.31 | 21 | 328 |
| TTAGCACTGAAGATGGAAGGGAGGC TTCGAATTACTATGTTGCTCCGTTG | 706 | 46.0 | 81.86 | 41.36 | -65.52 | 20 | 250 |

Table 4. 50-mer probes for pB10D8₁00324|SPBP35G2.13c gene in the *S. pombe* genome.

| Probe Number | Probe Sequence | BP | CG | MT | HT | FE | HD | DE |
|--------------|---|------|----|-------|-------|--------|----|-----|
| 1 | TTAAGTGTTCCTCGG GGATGCATAGATTC ATCATTTTCTCCAC CTTAAAAATGCGGG CATTTAAGTCTGTC | 1580 | 40 | 83.41 | 42.91 | -81.06 | 29 | 495 |
| 2 | GACGAACAGTGAGG AAATTCGGAGCTAT ACATATGTGCAGAA GGTTACTACCTAGG GTTTATGCTTAATT | 1749 | 40 | 85.54 | 45.04 | -83.78 | 27 | 326 |
| 3 | ACACTATGAAACCG CCATTGGGCTACTG TAGATTTGTATCCT TGATGAATCTGGGG TTCCATCAGACTG | 1878 | 44 | 84.38 | 43.88 | -82.19 | 30 | 197 |
| 4 | TGAAACCGCCATTG GGCTACTGTAGATT TGATCCTTGATGA ATCTGGGGTTTCCA TCAGACTGAACCTA | 1884 | 42 | 84.65 | 44.15 | -82.60 | 28 | 191 |
| 5 | CTTGATGAATCTGG GGTTCCATCAGAC TGAACCTACACTGT ATATTTGCAATAG TTACCTCAAGGCCT | 1918 | 40 | 82.65 | 42.15 | -80.31 | 29 | 157 |

Table 5. 70-mer probes for the *HoxA9* gene in the human genome.

our program is less limited by the host computer's hardware constraint.

Further research includes generalizing the algorithm so that edit distance can be used as a specificity measure of the oligos. In addition, we also hope to find and enforce other biological factors besides homogeneity, sensitivity, uniformity and specificity that affects the goodness of probes. We will continue to work closely with the people in the Genome Institute of Singapore to quantify these biological factors. This may further improve the quality of the probes selected by our algorithm.

Also, it is quite important to further reduce the time complexity of *FindProbe* while maintaining its accuracy. Such improvement can enable practical probes selection for the human genome and other larger genomes such as the tree genome.

Currently, *FindProbe* runs in Solaris sunfire and Linux operating systems. We hope to come out with a version that can run in Windows operating systems. This will facilitate a wider use of our software.

References

- [1] Baggerly K.A., Hess K.R., Stivers D.N., Abruzzo L.V., Coombes K.R. and Zhang W. Identifying differentially expressed genes in cDNA microarray experiments. *Journal of Computational Biology*, 8(6):639–659, 2001.
- [2] Bailey W. F. and Monahan A. S. In *J. Chem. Ed.*, pages 489–493, 1978.
- [3] Beheshti B., Braude I., Park P.C. and Squire J.A. Microarray cgh. *Methods Mol Biol*, 204:191–207, 2002.
- [4] Blohm Dietmar H and Guiseppi-Elie Anthony. New developments in microarray experiments. *Biotechnology*, 12:4147, 2001.
- [5] Bosch J. T., Seidel C. H., Lam S. B., Tuason N., Salijoughi S., and Saul R. Validation of sequence-optimized 70-base oligonucleotides for use on DNA microarrays. In *The TIGR genome sequencing and analysis conference (Poster)*, 2000.
- [6] Cherie Bond, Tian Mingting, Melia Dorothy, Zhang Shengwen, Borg Lisa, Gong Jianhua, Schluger James,

- Strong Judith A., Leal Suzanne M., Tischfield Jay A., Kreek Mary Jeanne, Laforge K. Steven and Yu Lei. Single-nucleotide polymorphism in the human mu opioid receptor gene alters β -endorphin binding and activity: Possible implications for opiate addiction. *Neurobiology*, 95:9608–9613, 1998.
- [7] Debouck C. and Goodfellow P.N. Dna microarrays in drug discovery and development. *Nat Genet*, 21(1):48–50, 1999.
- [8] Epstein Charles B. and Butow Ronald A. Microarray technology - enhanced versatility, persistent challenge. *Biotechnology*, 11:3641, 2000.
- [9] Gerhold D., Rushmore T. and Caskey C. T. Dna chips: promising toys have become powerful tools. In *Trends in biochemical sciences*, pages 168–173, 1999.
- [10] Kaderali L. and Schliep A. Selecting signature oligonucleotides to identify organisms using DNA arrays. *Bioinformatics*, 2002.
- [11] Kane M. D., Jatkoe T. A., Stumpf C. R., Lu J., Thomas J. D. and Madore S. J. Assessment of sensitivity and specificity of oligonucleotide (50mer) microarrays. *Nucleic Acid Res*, 28:4552–4557, 2000.
- [12] Keller GH. *Keller GH Manak MM. DNA Probes Second Edition*, chapter Section 1: Molecular hybridization technology, pages 1–9. Stockton Press, 1993.
- [13] Lipson D., Webb P., and Yakhini Z. Designing Specific Oligonucleotide Probes for the Entire *S.cerevisiae* Transcriptome. *WABI*, LNCS 2452:491–505, 2002.
- [14] Li F. and Stormo G. Selection of optimal DNA oligos for gene expression analysis. *Bioinformatics*, 17:1067–1076, 2001.
- [15] Lockhart D. J., Dong H., Byrne M. C., Follettie M. T., Gallo M. V., Chee M. S., Mittmann M., Wang C., Kobayashi M., Horton H. and Brown E. L. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14:1675–1680, 1996.
- [16] Manber U. and Myers G. Suffix arrays: a new method for online string searches. *SIAM Journal of Computing*, 22(5):935–948, 1993.
- [17] Myers E. W. A fast bit-vector algorithm for approximate string matching based on dynamic programming. In *Ninth Combinatorial Pattern Matching Conference*, pages 1–13, 1998.
- [18] Raddatz G., Dehio M., Meyer T. F. and Dehio C. Primearray: genome-scale primer design for dna-microarray construction. *Bioinformatics*, 17:98–99, 2001.
- [19] Rahmann S. Rapid large-scale oligonucleotide selection for microarrays. In *Proc. of the Second Workshop on Algorithms in Bioinformatics (WABI)*, pages 302–311, 2002.
- [20] Rouillard J. M., Herbert C. J. and Zuker M. Oligoarray: Genome-scale oligonucleotide design for microarrays. *Bioinformatics (Applications Note)*, 18:486–487, 2002.
- [21] SantaLucia J. J., Allawi H. T. and Seneviratne P. A. Improved Nearest-Neighbor Parameters for Predicting DNA Duplex Stability. *Biochemistry*, 35:3555–3562, 1996.