Selecting Optimum DNA Oligos for Microarrays

Fugen Li and Gary D. Stormo
Department of Genetics, Washington University School of Medicine
St. Louis, MO 63110, USA
Phone: 314-747-5535, FAX: 314-362-7855
lif@ural.wustl.edu
stormo@ural.wustl.edu

Abstract

High-density DNA oligo microarrays are widely used in biomedical research. In this paper we describe algorithms to optimize the selection of specific probes for each gene in an entire genome. Having optimized probes for each gene is valuable for two reasons. By minimizing background hybridization they provide more accurate determinations of true expression levels. And having optimum probes eliminates the need for multiple probes per gene, as is usually done now, thereby decreasing the cost of each microarray and increasing their usage. The criteria for truly optimum probes is easily stated but they are not computable at present. We have developed an heuristic approach that is efficiently computable and should provide a good approximation to the true optimum set. We have run the program on the complete genomes for several model organisms and deposited the results in a database that is available on-line (http://ural.wustl.edu/˜ lif/probe.pl).

1. Introduction

The complete sequences of nearly a dozen microbial genomes and a few of eukaryotic genomes are known. In several years the complete genome sequences of several metazoans, including the human genome, are expected to be completed [6]. Defining the role of each gene in these genomes and understanding of the genome functions as a whole present a great challenge. DNA microarray technology offers a great tool for these tasks [9].

DNA chips are glass surface bearing thousands of DNA fragments at discrete sites at which the fragments are available for hybridization. Hybridization of RNA and DNA-derived samples to DNA chips allows the monitoring of gene expression or occurrence of polymorphisms in genomic DNA [7]. Two DNA chip formats currently in wide use are the cDNA array format [21] and high density synthetic oligonucleotide array format [15, 7]. The oligonucleotide approach has some advantages because it allows the user to design probes for each gene to avoid regions that are repetitive or homologous to other known genes [7, 11]. Oligonucleotide array chips are currently generated by a photolithographic method. In the future other methods for arraying oligonucleotides will likely be available that can produce specifically designed chips rapidly and cheaply. The design of optimum probes would then be of significant benefit.

The challenge is how to identify the optimum probes for each gene. Empirically, the optimum probe for a gene would be the one with minimum hybridization free energy for that gene (under the appropriate hybridization conditions) and, maximum hybridization free energy for all other genes in the hybridizing pool. Unfortunately, those energies depend on knowledge that is not computable from the sequence alone, at least not currently. For example, the hybridization energy of the probe to the correct gene's cDNA depends on the complete structure of that cDNA. DNA (and RNA) structure prediction programs are not reliable enough for us to accurately predict that structure from the sequence. Another limitation is that the hybridization energy for every gene, both the correct one and the incorrect ones, depends on the concentration of those genes because this is a bimolecular interaction. For example, an incorrect gene with somewhat less intrinsic hybridization energy than the correct gene may give a larger signal, i.e. background, than the true gene does simply because it is present in a much greater concentration. We generally do not know in advance the concentration of all the cDNAs; in fact, that is what we are trying to measure. Over time we will learn what genes are present in what concentrations under various conditions, but then we won't need the chips to tell us what we already know. In addition to these issues, which are fundamentally uncomputable at this time, there is also the issue of the complexity of algorithms. In principle we

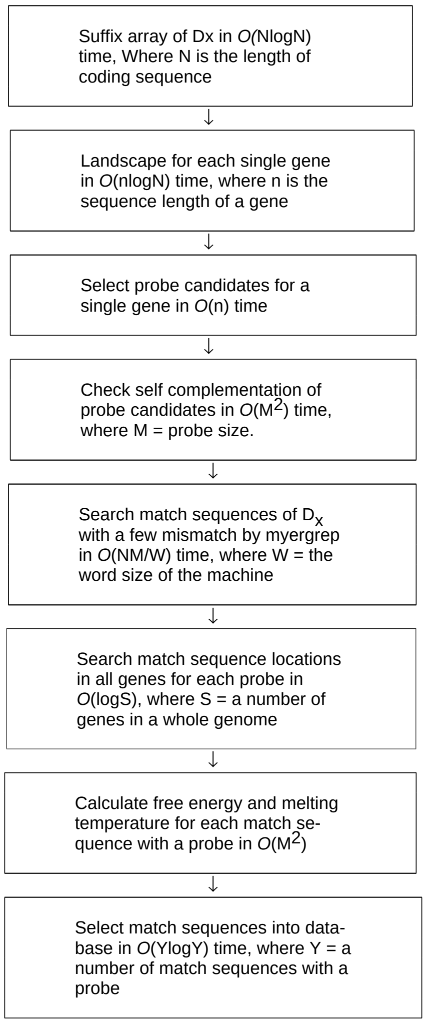
Suffix array of Dx in O(NlogN) time, Where N is the length of coding sequence
Landscape for each single gene in O(nlogN) time, where n is the sequence length of a gene
Select probe candidates for a single gene in O(n) time
Check self complementation of probe candidates in O(M2) time, where M = probe size.
Search match sequences of Dx with a few mismatch by myergrep in O(NM/W) time, where W = the word size of the machine
Search match sequence locations in all genes for each probe in O(logS), where S = a number of genes in a whole genome
Calculate free energy and melting temperature for each match sequence with a probe in O(M2)
Select match sequences into database in O(YlogY) time, where Y = a number of match sequences with a probe

Figure 1. The diagram describes the whole procedure of ProbeSelect program.

could compute the theoretical energy (i.e. ignoring internal structure and concentration) for every potential probe from each gene to every possible hybridization partner, but that task would be computationally intractable.

Only one program which is developed to design oligo probes for DNA chips by affymetrix [12] exists currently. This program has been used for commercial purpose. There is no way to find the algorithm of the program except that probe criteria have been described [12]. Our approach, described in detail below, is to attack the probe design problem in two major steps (composed of several minor steps), which are each tractable and should provide good, although not perfect, predictions of the optimal probes purely based on sequence information, probe criteria used in affymetrix program and standard stack energy. The first major step is to identify, for each gene, the set of candidate probes that maximize the minimum number of mismatches to every other gene in the genome. Since mismatch positions affect stability of oligo DNA hybridization structure those candidate probes are then compared more rigorously to determine their hybridization energy, both for their target sequences and for all of the other positions in the genome that match with some allowable number of mismatches. The optimum probes are then picked based on having free energy (ΔG) for the correct target in an acceptable range, and maximizing the difference in free energy to every other mismatched target.

By identifying optimal probes we may be able to reduce a number of probes per gene based on hybridization theory, without a reduction in sensitivity. However, we need experiments to determine what minimal number of probes per gene is need to be arrayed on a DNA chip. This problem is computationally intractable. This paper describes all of the algorithms used for probe candidate selection and probe data generated by the program. We have run the program on some model organisms and the selected probe data and gene information are stored in the ChipProbe database which is accessible on-line (http://ural.wustl.edu/ ˜ lif/probe.pl).

2. System and methods

ProbeSelect is written in C++ and was developed on Sun workstations running Solaris. The code is portable and has been implemented on HP workstations. The program consists of eight major components, described in detail below: 1) making a suffix array of the coding sequences from the whole genome; 2) building sequence landscapes for every gene based on the sequence suffix array; 3) choosing probe candidates based on the sequence word rank values; 4) checking self-complementarity of each probe candidate; 5) searching for matching sequences in the whole genome, allowing a certain number of mismatches by the program myersgrep [14]; 6) locating match sequence positions in all genes; 7) calculating the free energy (ΔG) and melting temperature (T_m) for each valid match sequence; 8) selecting match sequences that have stable hybridization structures with a probe based on free energy data and allow good discrimination with other targets in the genome. The architecture of the program is shown in Figure 1.

Sybase is used to design the ChipProbe database to store all probe information generated by ProbeSelect. ChipProbe database is linked with the web by CGI.pm [22] and Sybperl interface [16]. Several genomes are available in ChipProbe database.

3. Algorithm

3.1. Sequence Suffix Array

A suffix array [13] is a sorted list of all the suffixes of a sequence. It takes $O(N \log N)$ time to build a suffix array, where N is the length of the sequence. A suffix array permits on-line string searches in time $O(p + \log N)$, where p is the length of the sequence word. The details about suffix arrays are described by Manber and Myers (1993).

3.2. Sequence Landscape

Detailed information about a landscape is described by Levy et al.(1998). A landscape represents the frequency of all words in the query sequence that can be found in some database, in our case the coding and complementary sequences from the genome. A landscape of a simple 15 base query sequence, using the same sequence as a database, shows the basic elements of a landscape (Figure 2). Every cell in the landscape, $f(j, i)$, indicates where a word begins j, and its length i, and the value of a cell is the frequency of that word in the database. For example, the frequency of 2 at $f(3, 3)$ indicates that the 3-base word TCC starting at position 3 occurs twice in the sequence, the other location being at position 8. . Building a landscape for one gene takes $O(n \log N)$ time, where n is the length of the gene and N is a total length of coding sequence of a genome.

3.3. Selection of Probe Candidates for Each Gene

A sequence landscape contains a large amount of information that can be used to select probes. If we were only interested in the frequency in the genome of exact matches to the probe sequence we could simply look it up in the landscape, or we could get a good estimate by multiplying together the frequencies of the sub-words. But we need to know how many approximate matches there are, allowing for some number of mismatches. Fast approximate string matching algorithms, such as agrep, are too slow for our needs when we are comparing every potential probe in every gene to all possible target sites in the entire genome. Instead we use the landscape information to get an estimated rank of the number of approximate matches for each potential probe (i.e. sub-word) of each gene. We save the top Q (typically from 10 to 20) probe candidates for further analysis.

We have found that summing the frequencies of words at each of several different word lengths (i.e. heights in the landscape) gives a good predictor of the number of matches in the genome, allowing mismatches. Therefore candidate probes are chosen by finding the probe-sized words within the gene sequence that minimize the sum of its sub-words.

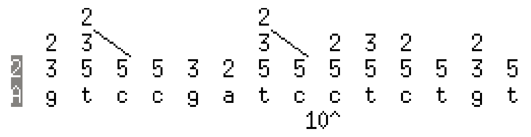

Figure 2. Landscape of a 15-base sequence showing the frequencies of all words compared to itself.

For example, if the word s is comprised of the sequence ATGCCA, ATG is the beginning sub-word and $f(ATG)$ is the frequency of the word which occurs in coding sequence and its complement of the whole genome. The "overlapping frequency" of word s is defined to be

$$F(s) = f(ATG) + f(TGC) + f(GCC) + f(CCA)$$

So the general formula for all words in a gene landscape is:

$$F_i(s_j) = \sum_{k=j}^{j+M-i} F_{ik}$$

where i is column height of the gene landscape, j is the word position, M is the probe size and $j < n - M$. n is the sequence length of the gene.

The first ten words will be saved based on the minimum overlapping frequency value of words at the different sizes of starting sub-words from 2 to probe size. The rank value of each word is how many times the word has been selected at different size of starting sub-words. A word is selected as a probe candidate from low to high rank values; rank of 1 is predicted to have the fewest approximate matches elsewhere in the genome.

3.4. Self Complementary Prediction of Each Probe

The algorithm for self complementary prediction of DNA primer sequences has been described [18]. This algorithm has been modified in the ProbeSelect program to predict self complementarity of each probe sequence. A half matrix is filled by comparing the probe sequence and its complement. For complementary bases the matrix value is 1, otherwise the value is 0. If the maximum value of a diagonal line is more than 30% of the probe sequence, the probe is substantially self complementary and is removed from the candidate list.

3.5. Approximate String Searching

The approximate string searching problem is to find all locations at which a query of length m matches a substring of a text of length n with k-or-fewer differences. A fast bit-vector algorithm for approximate string searching has been designed to run in $O(nm/w)$ time by Myers(1998), where w is the word size of the machine. Detailed information about this algorithm has been described by Myers(1998) . This algorithm has been used in the ProbeSelect program to find all match locations for the probe in the coding and its complementary sequences of the genome with a number of 4 or fewer mismatches allowed. We also tried 5 or fewer, which returns many more locations in the genome, but the extra matches almost never contribute competing sites once the T_m is considered. The sites with 4 or fewer mismatches occasionally contribute competing sites with similar T_ms, so all of those are considered when determining the optimum set of probes.

3.6. Localization of the Match Sequences in Each Gene

The fast approximate string searching is performed on the entire coding and non-coding sequences combined into a single large database. The matches can then be assigned to their exact positions within the specific genes or ORFs. All gene positions can be built into the sorted array. All match positions to the probe can be located in the specific gene by a binary search of the sorted array, which only takes $O(\log S)$ time where S is the number of genes or ORFs in the whole genome. If the match sequences are across two genes, they are not valid.

3.7. Free Energy and Melting Temperature Calculation

It is unlikely to calculate free energy and melting temperature between a probe and target on a solid chip, because DNA hybridization behavior on a chip is not the same as that in solution and the whole parameters on a chip are not available currently. Partial thermodynamic parameters for stack energy on gel matrix [8] have an approximate linar relationship with those in solution, although absolute data is different. Therefore, using thermodynamic parameters measured in solution could predict stability of DNA oligo duplexs theoretically.

As DNA oligonucleotide nearest-neighbor themodynamic parameters [20, 2, 19, 3, 4, 5, 17] are available, they provide a way to predict oligonucleotide DNA hybridization secondary structure. The parameters for free energy calculation in ProbeSelect are very similar to that in Zuker's single strand DNA secondary structure prediction program Mfold [24]. We could find the minimum energy hybridization structure using a standard dynamic programming approach based on the free energy rules. However, we already have the alignment of the probe sequence and its target and, since there are at most four mismatches between them, the lowest energy structure is usually the same as that alignment. Exceptions involve slight rearrangements in, and adjacent to, the mismatched positions. Therefore we designed a fast heuristic to test various alternatives to the alignment. This is M times faster than the full dynamic programming method where M is probe size, and nearly always ends up with the same optimum structure. In the few cases where they are different, the change in free energy is very small.

T_m could be used as a parameter to evaluate probe hybridization behavior. Since it is impossible to know target DNA concentration, the calculation can not be accurate. However, T_m calculation is still good for probe evaluation. With standard conditions, T_m [1, 23] is calculated for each probe based on

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

where ΔH and ΔS are the enthalpy and entropy for helix formation, respectively, and R is the molar gas constant($1.987 cal/^oC \times mol$). c is the total molar concentration of the annealing oligonucleotides when oligonucleotides are not self-complementary. As described earlier, this is an intermolecular hybridization so the free energy depends on the concentration. This is not generally known, so we set it to a constant of $1 \times 10^{-6} M$. If concentrations of individual genes were known those could be included in the calculation of T_m. The important thing we have to remind is that ΔH and ΔS are calculated based on standard hybridization solution conditions, because stack energy parameters for chip conditions are not available currently. If desired, the user can substitute other parameters that are more appropriate for the conditions of hybridization to the chips.

3.8. Selecting Match Sequences into Database Based on Free Energy Data

Since each probe may have many match regions in the whole genome when allowing four mismatches, it is not necessary to store all match sequences. The hybridization structures between some match sequences and a probe have high free energy and are very unstable. These structures may not be formed under general hybridization conditions, and even if they are formed, they do not contribute significantly to the background. Therefore, these sequences have no effects on the selection of the probe. However, some match sequences have large effects if they hybridized with a probe because they have free energies close to that of the probe to its target. The difference of free energy between the hybridization structures with and without mismatches is used as an index to filter the sequence and other data. Sites with a difference of 10 kcal/mole or less in free energy usually have a difference in T_m of less than 20^oC and can contribute to background hybridization. Therefore all of those mismatch sites are also stored in the database.

All sequences matched with a probe are sorted into an array based on free energy by Quicksort which takes $O(Y \log Y)$ time where Y is the number of sequences. An output filter checks the difference of free energy between all sequences and the probe to determine whether this sequence should be stored in the database or not. This algorithm takes $O(Y)$ time in a worst case.

4. Results

4.1. Performance for a few genomes

ProbeSelect was tested on a few of genomes to generate probes for each gene on a Sun workstation. It took under 2 minutes to finish T7 phage genome which is 39937 bps and includes 60 genes. About ten probes with 20 bases were generated for each gene and match sequences to each probe were searched with four mismatches. However it took one and half days to finish *E. coli* genome which is about 4.6×10^6bps and includes about 4300 genes, and almost four days to finish *Saccharomyces cerevisiae* genome which is about 1.2×10^7 bps and includes about 6000 genes. The size of a probe for *E. coli* and *Saccharomyces cerevisiae* is 23 bases and the match sequences were searched with 4 mismatches.

4.2. Probes generated by ProbeSelect

Probes are generated for each gene by ProbeSelect. Ten probes for the yeast CDC28 gene are shown in Table 1. Table 2 describes the detailed information about each probe for this gene and a part of match sequences with 4 or fewer mismatches. Most of probes from T7 phage have under 3 match positions with 4 or fewer mismatches, however, most probes from *E. coli* and *Saccharomyces cerevisiae* have under 5 match positions with 4 or fewer mismatches when the probe size is increased to 23 bases. Even some unique probes have been selected for many genes in both genomes. Therefore, increasing the probe size makes it much easier to find unique probes for each gene in a large genome. Determination of probe size and a number of mismatches for one of the genomes will help design good probes to limit nonspecific hybridization on a DNA array.

It is reasonable to compare probes from ProbeSelect with those from affymetrix program. However, we could not get any affymetrix probe sequence information from public domain. Another question which may be raised is why we do not use PCR primers as oligo chip probes. Many programs exist to select PCR primers for a given DNA sequence. PCR primer selection is less stringent than our task because in order to get an inappropriate PCR product both primers must hybridize within a relatively short distance of each other. But for microarrays, an inappropriate hybridization anywhere in the genome can lead to high background. We tested one PCR primer selection program (http://genome-www2.standford.edu/cgi-bin/SGD/web-primer) to see how its selections compare to those generated by our program. Table 3 lists the selected primers for three yeast genes (other genes give similar results) and the number of matching sites in the whole genome if four mismatches are allowed. In general each of them has multiple potential target sites in the genome, whereas our program returned primarily sites that are unique, and in general have many fewer alternative hybridization sites. So while the program is adequate for identifying good primers for PCR, it would not work nearly as well for identifying optimum probes for microarrays.

4.3. Free energy, melting temperature and secondary structure of probes

Free energy parameters are used to predict the stability of a hybridization structure between a probe and its match sequence. Our algorithm, described above, can determine the free energy of hybridization for many different secondary structures and the one with the lowest free energy is selected. In table 2, ΔG and T_m for some probe hybridization structures of yeast gene CDC28 are listed. Of the ten probes listed in Table 1, only probe-6 has an alternative target in the genome with a free energy difference within 10 kcal/mole of the true target free energy. Based on the data in Table 2 one could pick the optimum probe (or small set of probes) for the CDC28 gene.

This algorithm has been compared with the standard dynamic programming algorithm with free energy rules to fill the matrix. 95% of the free energy calculation and hybridization structure prediction are identical. Only 5% of them have minute differences which do not affect the probe selection. The algorithm which is included in ProbeSelect takes $O(M^2)$ time, but the standard dynamic programming algorithm with free energy rules takes $O(M^3)$ time.

4.4. Database searching on the Web

Since ProbeSelect produces tremendous data for each genome, and a database has been built to handle all the information. Also, it is very important for people who are interested in gene expression arrays to access the data. This web oriented database is available for a few of genomes, such as T7 phage, *E. coli*, *S. cerevisiae* and *C. elegans*. The database can be expanded when more genome sequences are finished. The probes may be searched for each gene based on your requirement. Also, detailed information about each gene and its probe can be searched in this database.

Table 1. Probes for yeast cdc28 gene generated by ProbeSelect*

gene-name	probe-name	probe-sequence	b-pos	num-matches
CDC28	PROBE-1	AAACTCCTCGCGTATGACCCTAT	820	1
CDC28	PROBE-2	AAGTCTAGATCCACGCGGTATTG	786	4
CDC28	PROBE-3	TCCTCGCGTATGACCCTATTAAC	824	1
CDC28	PROBE-4	AGACGAGGGTGTTCCCAGTACAG	141	1
CDC28	PROBE-5	TGCATACTGCCACTCACACCGTA	372	2
CDC28	PROBE-6	GGTATAGAGCTCCGGAGGTATTA	527	2
CDC28	PROBE-7	TATTAACCGGATTAGCGCCAGAA	840	3
CDC28	PROBE-8	GAGCAGCCATCCACCCCTACTTC	863	1
CDC28	PROBE-9	AGGACCAACCGTTAGGAGCTGAT	311	1
CDC28	PROBE-10	GAGCTCCGGAGGTATTACTGGGT	533	1

*b-pos means the probe is selected from the beginning position of the gene. num-matches means how many of positions a probe can match a whole genome with 4 or fewer mismatches

Table 2. Detail information of all probes for yeast cdc28 gene*

gene-name	probe--name	probe-match-seq alignment	m-gene	mb-pos	mis	ΔG	$\Delta\Delta$G	T_m	h-strand
CDC28	probe-1	AAACTCCTCGCGTATGACCCTAT TTTGAGGAGCGCATACTGGGATA	CDC28	820	0	-30.5	0	79.8	noncode
CDC28	probe-2	AAGTCTAGATCCACGCGGTATTG TTCAGATCTAGGTGCGCCATAAC	CDC28	786	0	-30.1	0	78.9	noncode
CDC28	probe-3	TCCTCGCGTATGACCCTATTAAC AGGAGCGCATACTGGGATAATTG	CDC28	824	0	-29.8	0	78.5	noncode
CDC28	probe-4	AGACGAGGGTGTTCCCAGTACAG TCTGCTCCCACAAGGGTCATGTC	CDC28	141	0	-31.6	0	82.6	noncode
CDC28	probe-5	TGCATACTGCCACTCACACCGTA ACGTATGACGGTGAGTGTGGCAT	CDC28	372	0	-32.1	0	82.8	noncode
CDC28	probe6	GGTATAGAGCTCCGGAGGTATTA CCATATCTCGAGGCCTCCATAAT	CDC28	527	0	-28.7	0	77.6	noncode
CDC28	probe-6	GGTATAGAGCTCCGGAGGTATTA CCATATCTCGAGGCCTTTATTAC	SLT2	590	4	-20.0	-8.7	66.0	noncode
CDC28	probe-7	TATTAACCGGATTAGCGCCAGAA ATAATTGGCCTAATCGCGGTCTT	CDC28	840	0	-29.9	0	78.3	noncode
CDC28	probe-8	GAGCAGCCATCCACCCCTACTTC CTCGTCGGTAGGTGGGGATGAAG	CDC28	863	0	-32.3	0	84.5	noncode
CDC28	probe-9	AGGACCAACCGTTAGGAGCTGAT TCCTGGTTGGCAATCCTCGACTA	CDC28	311	0	-31.4	0	82.2	noncode
CDC28	probe-10	GAGCTCCGGAGGTATTACTGGGT CTCGAGGCCTCCATAATGACCCA	CDC28	533	0	-31.4	0	82.5	noncode

*m-gen means a gene name of a sequenc a probe match. mb-pos is the starting position of the gene that a probe match. mis is the number of mismatches when a probe matches the sequence. h-strand means what sequence strand a probe hybridizes with.

5. Discussion

The probe size is usually 25 bases on the DNA chips made by Affymetrix. However, the necessary size depends on the probe hybridization stability and genomic size of the organism. For practical reasons, 20 bases are minimal size for the probe. Most probes selected for T7 phage by probeSelect do not have any other match in the genome even if four mismatches are allowed. However, for *E. coli*, most probes with 20 bases have more than 30 matches in the genome if four mismatches are allowed, and about 70 matches in the genome if five mismatches are allowed. The probe size should increase as the gemone size becomes larger. For yeast, most probes with 20 bases have about 70 matches in the genome if four mismatches are allowed. However, when probe size increases to 23 bases, most probes have less than 5 matches in the *E. coli* genome and more than 30% of the probes are unique even if four mismatches are allowed. Thus, the probe size has to be determined for each organism before ProbeSelect is run for a whole genome.

Setting the number of mismatches is very important for searching similar regions with each probe. A few genes have been tested for *E. coli*, when the probe size is 23 bases. Free energy (ΔG) of all match regions with the probes is calculated. The difference of free energy ($\Delta\Delta G$) between the hybridization structure with perfect match and that with mismatches is also calculated. The $\Delta\Delta G$ values from 0 to -10 for four or less mismatches contain all $\Delta\Delta G$ values for five or less mismatches for all testing genes. This means all hybridization structures with four or less mismatches have enough information to select probes with 23 bases for *E. coli*. Also, we have tested a few yeast genes and had the same results as *E. coli*. We chose 23 bases as the probe size and 4 for the number of mismatches. Thus, for differ-

Table 3. Primers for three yeast genes generated by web-primer*

gene-name	primer-type	primer-sequence	num-matches
CDC28	forward	ATGAGCGGTGAATTAGCAAATTA	2
CDC28	reverse	TTATGATTCTTGGAAGTAGGGGT	5
CYS3	forward	ATGACTCTACAAGAATCTGATAA	14
CYS3	reverse	TTAGTTGGTGGCTTGTTTCAAGG	6
ACT1	forward	ATGGATTCTGGTATGTTCTAGCG	4
ACT1	reverse	TTAGAAACACTTGTGGTGAACGA	3

*The URL of web-primer program is http://genome-www2.stanford.edu/cgi-bin/SGD/web-primer. Only two best primers for each gene are shown here. num-matches means how many of positions a primer can match a whole genome with 4 or fewer mismatches

ent genomes, we may have different sizes for a probe and different number of mismatches allowed. Choosing the reasonable probe size and the number of mismatches can save significant computation time and computer memory, which is critical for a large genome.

Our program has the same criteria as affymetrix program [12] to select probes. Further, we even add free energy calculation to strengthen our probe selection. ProbeSelect uses higher criteria to select probes than the program used by affymetrix. Affymetrix chips have been confirmed experimentally. Although laboratory conditions limit a test of probes selected by ProbeSelect our probes at least should work as good as affymetrix probes.

It takes 4 days for ProbeSelect to finish the yeast genome when on average 8 probes with 23 bases are selected for each gene and 4 mismatches are allowed to search similar regions to each probe. It may take a few weeks to finish worm genome. Thus, it is unlikely for ProbeSelect to run the human genome in the reasonable time if the algorithm is not improved. Based on the algorithm used in ProbeSelect, we could improve it to change the performance for the program. The myersgrep algorithm in ProbeSelect for approximate matching takes about 50% of total time. Since a suffix array for the coding sequence has been built for the gene landscapes, it could be used for approximate string search which should be much faster than myersgrep. Work on optimizing that approach is currently underway.

Acknowledgements

This work reported here has been supported by a grant (ER61606) from the Department of Energy. We thank Jon Sauer for the use of his HP computers for testing various parts of the program.

References

[1] F. Aboul-ela, D. Hoh, and I. J. Tinoco. Base-base mismatches - thermodynamics of double helix formation for dCA3XA3G + dCT3YT3G (X, Y = A, C, G, T). *Nucleic Acid Res.*, 13:4811–4825, 1985.

[2] H. T. Allawi and J. J. SantaLucia. Thermodynamics and NMR of internal G.T mismatches in DNA. *Biochemistry*, 36:10581–10594, 1997.

[3] H. T. Allawi and J. J. SantaLucia. Nearest neighbor thermodynamic parameters for internal G.A mismatches in DNA. *Biochemistry*, 37:2170–2179, 1998a.

[4] H. T. Allawi and J. J. SantaLucia. Thermodynamics of internal C.T mismatches in DNA. *Nucleic Acids Res.*, 26:2694–2701, 1998b.

[5] H. T. Allawi and J. J. SantaLucia. Nearest-neighbor thermodynamics of internal A.C mismatches in DNA: Sequence dependence and pH effects. *Biochemistry*, 37:9435–9444, 1998c.

[6] J. L. DeRisi, V. R. Lyer, and P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–686, 1997.

[7] D. Gerhold, T. Rushmore, and C. T. Caskey. DNA chips: promising toys have become powerful tools. *Trends in biochemical sciences*, 24:168–173, 1999.

[8] A. Kunitsyn, S. Kochetkova, and V. Florentiev. Partiel thermodynamic parameters for prediction stability and washing behavior of dna duplexes immobilized on gel matrix. *Journal of Biomolecular Structure and Dynamics*, 14:239–244, 1996.

[9] E. S. Lander. Array of hope. *Nature Genetics*, 21:3–4, 1999.

[10] S. Levy, L. Compagnoni, E. W. Myers, and G. D. Stormo. Xlandscape: the graphical display of word frequencies in sequences. *Bioinformatics*, 14:74–80, 1998.

[11] R. J. Lipshutz, S. P. A. Fodor, T. R. Gingeras, and D. J. Lockhart. High density synthetic oligonucleotide arrays. *Natural Genetics*, 21:20–24, 1999.

[12] 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.

[13] U. Manber and E. W. Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal of Computing*, 22:935–948, 1993.

[14] E. W. Myers. A fast Bit-vector algorithem for approximate string matching based on dynamic programming. In *Ninth*

Combinatorial Pattern Matching Conference, pages 1–13, Piscataway, NJ, 1998.

[15] A. C. Pease, D. Solas, E. J. Sullivan, M. T. Cronin, C. P. Holmes, and S. P. Fodor. Lightgenerated oligonucleotide arrays for rapid DNA sequence analysis. *Proc. Natl. Acad. Sci. U. S. A.*, 91:5022–5026, 1994.

[16] M. Peppler. Sybperl2.0: using the Sybase::CTlib module. *http://www.mbay.net/ mpeppler*, 1996.

[17] N. Peyret, P. A. Senerviratne, H. T. Allawi, and J. J. SantaLucia. Nearest-neighbor thermodynamics and NMR of DNA sequences with internal A.A, C.C, G.G and T.T mismatches. *Biochemistry*, 38:3468–3477, 1999.

[18] R. Rozen and H. J. Skaletsky. Primer3. *http://www-genome.wi.mit.edu/genome_software/other/primer3.html*, 1998.

[19] J. J. SantaLucia. A unified view of polymer, dumbbel, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc. Natl. Acad. U.S.A*, 95:1460–1465, 1998.

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

[21] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470, 1995.

[22] L. Stein. *Official Guide to Programming with CGI.pm - the standard for building web scripts*. John Wiley & Sons, Inc., New York, NY., 1998.

[23] W. J. S. W. Rychlik and R. E. Rhoads. Optimization of the annealing temperature for DNA amplification in vitro. *Nucleic Acids Res.*, 18:6409–6412, 1990.

[24] M. Zuker, D. H. Mathews, and D. H. Turner. Algorithms and thermodynamics for RNA secondary structure prediction. In Barciszewski, J & Clark, B.F.C. Clark, editor, *A Practical Guide in RNA Biochemistry and Biotechnology*. Kluwer Academic Publishers, NATO ASI Series edition, 1999.