BIOINFORMATICS ORIGINAL PAPER Vol. 23 no. 10 2007, pages 1195–1202
doi:10.1093/bioinformatics/btm114

Sequence analysis

A fast and flexible approach to oligonucleotide probe design for genomes and gene families

Shengzhong Feng[1,2] and Elisabeth R.M. Tillier[2,3,*]

[1]Institute of Computing Technology, Chinese Academy of Sciences, China, [2]Ontario Cancer Institute, University Health Network, Toronto, Canada and [3]Department of Medical Biophysics, University of Toronto, Canada

Received on January 3, 2007; accepted on March 15, 2007

Advance Access publication March 28, 2007

Associate Editor: Martin Bishop

ABSTRACT

Motivation: With hundreds of completely sequenced microbial genomes available, and advancements in DNA microarray technology, the detection of genes in microbial communities consisting of hundreds of thousands of sequences may be possible. The existing strategies developed for DNA probe design, geared toward identifying specific sequences, are not suitable due to the lack of coverage, flexibility and efficiency necessary for applications in metagenomics.

Methods: ProDesign is a tool developed for the selection of oligonucleotide probes to detect members of gene families present in environmental samples. Gene family-specific probe sequences are generated based on specific and shared words, which are found with the spaced seed hashing algorithm. To detect more sequences, those sharing some common words are re-clustered into new families, then probes specific for the new families are generated.

Results: The program is very flexible in that it can be used for designing probes for detecting many genes families simultaneously and specifically in one or more genomes. Neither the length nor the melting temperature of the probes needs to be predefined. We have found that ProDesign provides more flexibility, coverage and speed than other software programs used in the selection of probes for genomic and gene family arrays.

Availability: ProDesign is licensed free of charge to academic users. ProDesign and Supplementary Material can be obtained by contacting the authors. A web server for ProDesign is available at http://www.uhnresearch.ca/labs/tillier/ProDesign/ProDesign.html

Contact: e.tillier@utoronto.ca or fsz@ncic.ac.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Microbial sequencing has revealed that the gene complement of closely related species and even strains within a species can vary dramatically due to the ability of bacteria to adapt quickly by modifying their genomic content. For example it has been discovered that bacterial virulence factors are encoded in pathogenicity islands that can readily be exchanged from one species to another through horizontal gene transfer (Garcia-Vallvé *et al.*, 2000). Given this potentially important rate of genetic exchange (Beiko *et al.*, 2005), monitoring the gene content of a microbial community rather than the individual genomes becomes important.

In order to evaluate the activities and functions of microbial communities, it is useful to characterize their genetic diversity and to analyze the individual members of gene families. Recently, metagenomic sequencing is making available sequences from samples from whole environmental communities (Pennisi *et al.*, 2004; Tyson *et al.*, 2004; Venter *et al.*, 2004). Direct complete sequencing is still expensive however, and not necessary for monitoring the presence of specific gene families. The application of microarrays to environmental samples has largely focused on the detection of specific signature sequences for the purpose of detecting particular organisms, and particularly pathogens (Call, 2005; Cho and Tiedje, 2001). Because of the tremendous microbial diversity in the environment, extensive coverage (i.e. the number of sequences we are able to detect) is very important (Guschin *et al.*, 1997; Holben and Harris, 1995; Torsvik and Ovreas, 2002). We propose an approach for the design of gene family microarrays to monitor the gene content of microbial communities. Gene family microarrays will be useful for medical and environmental diagnosis and will provide an alternative to costly genome libraries and to the sequencing of environmental samples.

A gene family is defined as a group of homologous sequences, but this definition is not specific and different clustering schemes can be used to give different groupings of the sequences. Clustering schemes for different levels of sequence identity can be provided by programs such as TribeMCL (Enright *et al.*, 2002) and CD-HIT (Li and Godzik, 2006) which provide non-overlapping clusters (each sequence belongs to a single cluster). In this article, we assume that a reasonable clustering of the sequences has been achieved. Our intent is to determine probes to identify each group of sequences defined by the user, although a re-clustering by merging highly similar groups may be performed when necessary to improve coverage. To reduce the number of probes necessary to cover all members of a gene family, it may be possible to design probes that allow a small number of mismatches with the target sequences (He *et al.*, 2005; Kane *et al.*, 2000; Zhou 2003; Tiqiu *et al.*, 2004; Li *et al.*, 2005; Liebich *et al.*, 2006).

*To whom correspondence should be addressed.

Probes designed for microarrays need to satisfy sensitivity, specificity and consistency requirements. For sensitivity, the self-complementarity of probes should be avoided or they will tend to hybridize to themselves rather than to their intended targets. For specificity, probes should be specific for the intended gene or gene family and not complementary to other sequences, thus avoiding cross-hybridization. Finally, for consistency, the melting temperature for all probes should be within some small range so that they can hybridize to their intended targets at the same temperature within an experiment.

Programs for oligonucleotide probe design are assessed using the three main performance indices of coverage, efficiency and flexibility. The tool should provide a high coverage for the probes, meaning that a large proportion of target genes and gene families are specifically identified. Efficiency then measures the tool's speed in generating specific probes and eliminating probes that cross-hybridize. Most existing algorithms aim to optimize efficiency as speed is not only desirable, but crucial in large-scale applications. Additionally, to accommodate the experimental design, the tool should be flexible and be able to generate probe sets of different lengths and for different hybridization conditions.

Many algorithms have been proposed to solve the probe selection problem. One general approach has been to enumerate all possible probe sequences in a suffix tree or suffix array (see Gusfield, 1997), which is then pruned to meet sensitivity, specificity and consistency requirements. The level of accuracy of these methods is often inversely proportional to their speed, and several heuristic approximations are necessary to improve efficiency. Li and Stormo (2001) proposed such an algorithm and were able to design a length-24 probe set for the *Saccharomyces cerevisiae* genome (6343 genes, 9.5 MB). Kaderali and Schliep (2002) also used a suffix array technique and focused on the accuracy of the probe set generated in their algorithm by using heuristic dynamic programming to compute the most stable alignment between every probe and its target sequence. Although their solution has higher accuracy, their algorithm is very slow and is unsuitable for large-scale data. Rahmann (2003) subsequently presented a fast algorithm that is practical for designing short probes up to 30 nt. By computing a probe's longest common contiguous substring, it only approximates the specificity of a probe, which results in a much higher efficiency. This algorithm allowed the selection of probes for a large genome like *Neurospora crassa* (10082 genes, 38 MB). This approach has several drawbacks, however. First, it can only be used for the design of short probes which may not provide the specificity required in some applications. Furthermore, it can potentially miss some useful probes because of the approximation used for specificity.

Unlike these previous approaches, Sung and Lee (2003) used a gapped-hashing algorithm to enumerate candidate probes and used several smart filtering techniques to reduce the search space. Their consistency and sensitivity filter eliminates probes with high G+C content, extreme melting temperatures and secondary structures. By using the pigeon hole principle, the algorithm avoids redundant comparisons, which reduces the time complexity of specificity filtering.

In the context of identifying particular sequences from related ones (for identifying a particular virus subtype for example), Klau *et al.* (2004) presented an exact approach to the problem of selecting non-unique probes whose hybridization patterns can then be deconvoluted to identify the presence of particular sequences. Their approach is based on integer linear programming mixed with a branch-and-cut algorithm for solving the group separation problem in the general case. Zheng *et al.* (2004) also proposed an algorithm to find unique oligonucleotides in large Unigene clusters from EST databases. Both these algorithms are applicable when probes for specific sequence identification are necessary. For our application where only detecting the presence of a gene family is required, the algorithm also developed by Zheng *et al.* (2004) to find the frequent oligonucleotides in the Unigene clusters is more suitable. However, this algorithm lacks flexibility as the probe lengths are limited to special values (33, 36, etc.) in the range of 20–50.

When sets of target sequences are highly similar to each other, truly sequence-specific probes cannot be found due to potential cross-hybridization. Although it is difficult to define meaningful specific groups for probe design (Behr *et al.*, 2000), a cluster- or group-specific probe concept has been applied in several programs, such as PRIMROSE (Ashelford *et al.*, 2002) and ARB (Ludwig *et al.*, 2004; Meier, 2004; Zhang, 2002). These have been used to design short oligonucleotide probes (20 bases) from a group of sequences; however these programs cannot be used for the design of longer probes (>50 bases).

In addition to these, there are many available software packages implementing some of the previous algorithms that have been developed for oligonucleotide probe selection for different applications. For example OligoWiz (Nielsen *et al.*, 2003), PROBEmer (Emrich *et al.*, 2003), OligoPicker (Wang and Seed, 2003), OligoArray 2.1 (Rouillard *et al.*, 2003), Osprey (Gordon and Sensen, 2004) and Picky (Chou *et al.*, 2004) were developed to generate sequence-specific probes for each gene of a given genome.

A program for designing probes to gene families and sub-families was recently developed by Chung *et al.* (2005). They presented an algorithm named HPD (Hierarchical Probe Design) for designing long oligonucleotide probes for highly conserved gene sequences. HPD uses hierarchical clustering to cluster the sequences into sub-families and automatically generates probes against all nodes (clusters) of the clustering tree for sequences of a conserved functional gene. HPD was implemented on the Microsoft Windows platform using ClustalW (Thompson *et al.*, 1994) and NCBI-BLAST (Altschul *et al.*, 1990, 1997). It is a very slow program particularly for large-scale datasets.

We propose a new approach to gene-specific and cluster-specific probe selection for the detection of genes and gene families, called ProDesign. Since a gene family groups sequences that are homologous but not identical, ProDesign uses spaced seed hashing (Brown *et al.*, 2004; Keich *et al.*, 2004; Ma *et al.*, 2002; Noé and Kucherov, 2004, 2005; Xu *et al.*, 2006) rather than a suffix tree algorithm in order to benefit from the allowance of mismatches between a probe and its targets.

ProDesign provides a new approach for probe selection that builds word lists based on spaced seed hashing with only a

single scanning of all the sequences. All the pairwise similarity scores between sequences are calculated based on the number of shared words. Given the initial list of sequences grouped in gene families, re-clustering according to the pairwise similarity scores may then be used to improve the coverage of probes. Probes are selected such that they are almost complementary with their targets but are dissimilar to sequences outside of their intended cluster (Kane *et al.*, 2000; Rhee *et al.*, 2004; Steward *et al.*, 2004). Additionally, all the probe candidates are filtered with sensitivity and consistency requirements. Subsequently, more accurate melting temperature calculations are done with the OligoArrayAux software package (Markham and Zuker, 2005). Here we show that ProDesign obtains a high coverage of target sequences. It is also time efficient and very flexible.

2 METHODS

2.1 Definitions

We denote the input dataset as $X=\{x_1,x_2,\ldots,x_k\}$, where the generic string x_i is a DNA sequence over the alphabet $\sum=\{A,C,G,T/U\}$ and k is the cardinality of the set. Let n_i denote the length of the ith sequence, then n represents the total size of the input in nucleotides.

A word is defined as a subsequence w of length m which occurs at position r of sequence x_i if $w_{[1]}=x_{i[r]},\ldots,w_{[m]}=x_{i[r+m-1]}$, where $w_{[i]}$ is the ith letter of the word w, $x_{i[j]}$ is the jth letter of sequence x_i, and $j=r,\ldots,r+m-1$. In order to find non-exactly matching words in sequences, seeds are specified using a seed pattern built over a three-letter alphabet #, @ and _ (Kucherov *et al.*, 2006; Noé *et al.*, 2004, 2005). This alphabet is used to compare and score the bases between two sequences. In this alphabet, positions with # must have an exact nucleotide match, positions with _ allow a mismatch and positions with @ allow a transition mutation (A ↔ G or C ↔ T). The weight of a seed pattern is defined as the number of # plus half the number of @ and $S(m,t)$ represents a seed of length m and weight t. In this work, we use spaced seed patterns previously identified by Kucherov *et al.* (2006) to be appropriate seeds for sensitive sequence comparisons.

A word is said to occur in sequence x_i given a spaced seed $S(m,t)$, if $w_{[j]}=x_{i[r+j-1]}$ at positions # in the seed and $w_{[j]}$ matches or is a transition of $x_{[i+j-1]}$ at @ positions of the seed. For example, the word 'ACACT' does not exactly occur in sequence 'ACGGTCG' (i.e. based on the seed $S(5,5)=$'#####'), but it does occur based on the seed $S(5,3)=$'#@@_#' (at position 1).

Let $f_w(x_i)$ be the occurrence of word w in sequence x_i. If w occurs in x_i, $f_w(x_i)=1$, otherwise $f_w(x_i)=0$. If $f_w(x_i)=1$ only for x_i, while for any other sequence x_l in $X-\{x_i\}$, $f_w(x_l)=0$, then the word w is specific for x_i. If for each sequence x_i in group G, $f_w(x_i)=1$, while for any sequence x_i in $X-G$, $f_w(x_i)=0$, then the word w is specific for the group G, and W_G is the set of specific words for group G. A less stringent threshold for group specificity can also be set. ProDesign requires the presence of the word in over 95% of the sequences in group G and in fewer than 5% of the sequences in other groups.

A probe candidate for group G consists of one or more specific words of G. If a probe candidate contains two or more specific words, then these must be in tandem and they can overlap or have gaps between them (usually gaps should be fewer than 3 bases; Rimour *et al.*, 2005). The probe candidate consists of only group specific words therefore it is itself group specific. A probe candidate for a group with only a single sequence is also sequence specific (gene specific).

2.2 The ProDesign procedure

The input to ProDesign is a list of sequences and an initial list of groups to which those sequences are assigned. The program then proceeds in four stages described below. In the first stage, word lists are built based on spaced seed hashing. The second stage consists of finding the probe candidates for each group using the word lists. In the process, a consistency and sensitivity filter is applied to eliminate probe candidates with too high or low G+C content, extreme melting temperatures and secondary structures. In the third stage, groups without probes are re-clustered and new probes are selected based on the new groups and filtered using the same algorithm as in stage 2. This re-clustering stage can be further iterated as needed. Once probe sets have been generated for each spaced seed, the final stage of the program consists of selecting optimal probe sets by considering their melting temperature and hybridization properties using the package OligoArrayAux. The flowchart for the complete process is shown in Supplementary Figure 1, and stages 1–3 are detailed in the following sections.

2.2.1 Building the word lists and finding group-specific words The aim of this stage is to generate all possible words that are specific for every group of sequences. This is accomplished in three steps. The process starts by building a hash table for each spaced seed. The key size of the table is the weight t of the spaced seed. The nucleotides A, T, G and C are encoded by 00, 01, 10 and 11, and the size of the hash table (the number of keys) is therefore 4^t. The spaced seed hash table is then used to create the hits matrix (Fig. 1) which records the words found in each sequence group according to the seed. If for any sequence j of the kth group, $h_{ij}=1$, then the word is found in the cluster and we say $H_{ik}=1$. For each word w_i a list of groups for which $H_{ik}=1$ is created. If there is only one element in a word list, the word is specific for the corresponding group.

2.2.2 Finding probe candidates A probe candidate for group G consists of one or more group-specific words of G in tandem. The words may overlap or be separated by small gaps. Because it consists of group-specific words, the probe candidate is also specific for group G. This stage can also be broken down into several steps as outlined in Figure 2. All probe candidates are then checked for low complexity, melting temperature and G+C content requirements. If the requirements cannot be satisfied, the candidate is eliminated from the list of probe candidates.

groupID	0		1			2			G
Key \ seqID	0	1	2	3	4	5	6		N
00 00 00 00	−1	1	−1	1	−1	1	−1	~	1
00 00 00 00	1	1	−1	−1	−1	−1	−1		−1
00 00 00 00	−1	−1	1	1	1	−1	−1		−1
00 00 00 00	1	1	−1	−1	−1	1	1		−1
~								h_{ij}	~
11 11 11 10	1	−1	−1	−1	−1	−1	−1	~	−1
11 11 11 11	−1	1	1	1	1	−1	−1		−1

Fig. 1. Example hit matrix (with $t=4$). The hit matrix is used to find words, which are present in over 95% of the sequences of one or more groups, but are found in <5% of the sequences in the other groups. All the elements of the matrix are set to −1 initially. If the ith word is found in the jth sequence based on a specified spaced seed, then $h_{ij}=1$. The table is then used to find group-specific (in dark gray) and shared (in light gray) words, where $h_{ij}=1$ almost exclusively in sequences belonging to specified groups. For example, the word '00 00 00 01', is specific for the sequences (0, 1) of group 0, so group 0 is inserted into the word '00 00 00 01' list. The word '00 00 00 11', is present in all the sequences (sequence 0, 1 and 5, 6) of groups 0 and 2, so both groups are inserted into the word '00 00 00 11' list as a shared word (light gray).

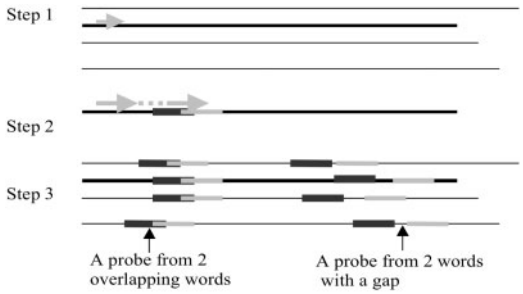

Fig. 2. Probe selection based on group-specific words. For each sequence group, we randomly choose one sequence (step 1, the thick line) and scan it to find the first group-specific word (step 2, the dark gray substring). If the length of the word substring is in the user-specified range, the substring is a probe candidate, otherwise, the scan is extended forward until the second group-specific word is found (step 2, the light gray substring). If the length of the substring of the two joined words is in the user-specified range with all gaps less than 3 bases, then we proceed to step 3, which is to search for that substring in all other sequences of the group. Otherwise the scan of the original sequence is continued. If an appropriate substring is found in all the sequences of the group, this substring is added to the list of the probe candidates for the group.

2.2.3 Re-clustering the groups without a probe The goal of this stage is to re-cluster the groups for which no probe was found. This can happen when groups contain highly divergent sequences, or when there are highly similar sequences found in between groups. We can address the later case by merging similar groups. To do this we use the hits table (Fig. 1) to calculate the similarity of the groups in terms of the number of shared words. Each group without probes is clustered together with other groups similar to it, but only if the new clustering results in words becoming specific to the new larger group. The re-clustering step is optional and the number of clustering iterations is set as user input.

2.3 Time complexity

In stage 1, all the input sequences are scanned once and the word lists are built. The time complexity is $O(n)$, where n is the size of the input dataset. In stage 2, all the sequences of the group are scanned once for every group-specific word. The number of group-specific words should be less than the length of the minimal sequence in the group. Therefore, the time complexity of this stage is less than $O(n^*m)$, where m is the typical length of the sequences. In stage 3, in order to find the shared words, a sample sequence of the group without a probe should be scanned once. Therefore, the time complexity of this stage is $O(k^*m)$, where k is the number of groups. In stage 4, the hybridization prediction (melting temperature calculation, free energy calculation, etc.) for every probe is $O(p^3)$, where p is the typical length of probes. Since several probes can be found for every group, the total time complexity of this stage is $O(kp^3)$. Usually, $k \ll n$, and $p \ll m$, therefore, including the hybridization prediction, the total time complexity is $O(n+n^*m+k^*m+kp^3) \approx O(nm)$.

2.4 Implementation

The algorithm presented here was implemented in a program called ProDesign written in ANSI C. To calculate the folding energy, melting temperature and hybridization, OligoArrayAux was integrated into the ProDesign package. The input data of ProDesign are two files. The first lists the gene families, and the second is a FASTA-formatted file of all sequences. Many parameters can be set by the user to modify the default settings of the command options. Unless otherwise noted, the default parameters were used in this article. The default probe length was set to 20–70 bases. The default spaced seed (with weight $t=10$) was '#@#_#@_#_@#__@###'. The hybridization threshold of heteroduplex formation between the probe and the target sequences was set to −30 kcal/mol of hybridization free energy. To remove the probes having hairpin secondary structures, the default self-annealing energy was set to −3 kcal/mol (Bodrossy *et al.*, 2003). To obtain the melting temperature, free energy rules were applied at 65°C. The allowable G + C content range was set to 35–65%. The number of final probe sets for each group of sequences was limited to one.

The output of ProDesign is a tabulated text file listing sets of probes consistent in melting temperature (e.g. within 5°C). The tabulated data for each probe include cluster ID, probe sequence and melting temperature. The computer used for the design tasks described here has a 3.06 GHz Pentium Xeon CPU with 4 GB physical RAM and runs the Linux operating system.

3 RESULTS

We applied ProDesign to several types of datasets and determined its efficiency in terms of the number of genes or groups for which probes could be found (coverage). For gene-specific oligonucleotide probe design, the gene coverage is the number of genes covered specifically by all the probes; while for group-specific oligonucleotide probe design, the group coverage is the number of groups covered specifically by all the probes.

3.1 Probe design for a single genome

Several computer programs, such as OligoArray, OligoPicker and YODA (Nordberg, 2005), have been developed for finding gene-specific probes for single genomes. A custom similarity search algorithm was also developed in YODA. By setting the size of each cluster to 1 (i.e. each cluster has a single sequence), ProDesign can also design gene-specific probes for single genomes. We considered 17 bacterial genomes and compared the results of ProDesign and YODA (Supplementary Table 1). Unlike ProDesign, YODA generated probes of a fixed length and also required a set narrow range for the melting temperature. The flexibility of ProDesign in probe length and melting temperature resulted in an increased coverage, approaching 100% for all genomes, which is 1–5% higher than that obtained by YODA.

Since the coverage obtained with YODA and ProDesign are correlated (Fig. 3, $R^2 = 0.55$), we considered whether the coverage was affected by the degree of duplication in the genomes (as measured by the percent of genes sharing ≥90% sequence identity). We found that for the highly duplicated *Rhodopirellula baltica* genome, ProDesign and YODA did have reduced coverage (Supplementary Table 1). However, YODA gave highly variable results that were not correlated with the sequence identity (Fig. 3). Because groups of highly similar sequences will result in decreased coverage, several programs employ some sort of clustering strategy. OligoArray clusters highly similar sequences for which no probe was found as a group, and a probe specific for the new group is then found. In YODA, a manual iterative procedure is used through relaxing some clustering requirements, but we found it was not

always successful. As described in the previous section, an automated clustering strategy was implemented in ProDesign.

To compare ProDesign with OligoArray, OligoPicker and YODA, we used the same dataset of the yeast *S.cerevisiae* genome used by Nordberg (2005) (Table 1). For this dataset, OligoPicker, YODA and ProDesign finished the probe design process in <15 min. Because of its slow clustering strategy and similarity search algorithm, OligoArray2.1 took 201 min to finish this design task. ProDesign, although it also uses clustering, is still the fastest, because it was developed with a fast heuristic clustering algorithm and an alignment-free custom similarity search algorithm based on spaced seed hashing.

No clustering methods were used in the packages YODA and OligoPicker. With a clustering strategy, the coverage was improved significantly (for OligoArray, from 92.6 to 99.4%; for ProDesign, from 97.4 to 99.4%), and the final coverage was higher than that without a clustering strategy (for OligoPicker, 92.6%; for YODA, 90.3%).

3.2 Probe design for gene families

In the algorithm of ProDesign, the basic data structure is a gene group, instead of a sequence. Hence, ProDesign is specifically designed to select probe sets for gene families. HPD (Chung *et al.*, 2005) is analogous in function with ProDesign. HPD integrates BLAST for sequence similarity searching, and ClustalW for multiple sequence alignment and clustering. We attempted to use HPD on a single genome scale but found that for thousands of sequences, the sequence clustering and specificity searching was very time-consuming, and sometimes even failed to complete for larger genomes (for example, for *S.cerevisiae*).

To compare ProDesign with HPD, we selected the same test data used by Chung *et al.* (2005): the nitrite reductase (*nirS*) and methane monooxygenase (*pmoA*) gene sequences. The sequences were downloaded from GenBank (http://www.ncbi.nlm.nih.gov). A total of 421 *nirS* sequences of length >699 bp and 490 *pmoA* sequences of length >449 bp were selected. The *nirS* sequences had an average sequence identity of 68.4% ± 7.5 (SD) and the *pmoA* sequences had an average identity of 69.5% ± 6.4.

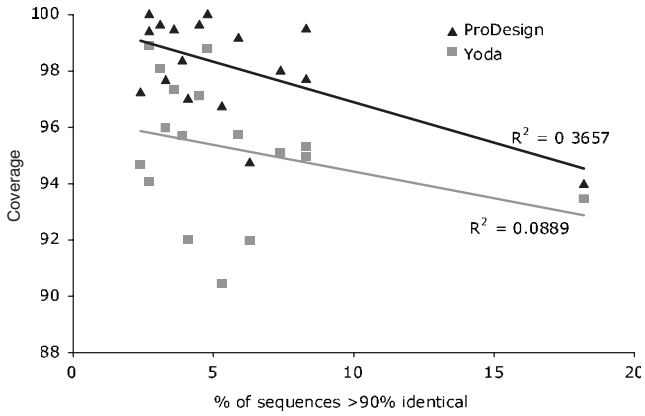

Fig. 3. Effect of gene duplication on probe coverage. The percent coverages obtained by ProDesign and YODA are plotted against the frequency (%) of genes with >90% sequence identity within each genome.

Initially, the sequences were not clustered so that each group contained only one sequence, and the probes generated in Stage 2 of ProDesign were sequence specific. After re-clustering, a total of 679 probes (364 sequence-specific probes and 315 group-specific probes) were found in the *nirS* set, and 655 probes (323 sequence-specific probes and 332 group-specific probes) in the *pmoA* set. Sequence-specific probes covered 86.4% of *nirS* sequences and 65.7% of *pmoA* sequences. ProDesign creates larger clusters only when required to increase coverage in a non-hierarchical manner that is quite different from the HPD approach. HPD tries to find probes for each node (cluster or sequence) in a hierarchical tree. For HPD, we report the group-specific coverage for only those probes found at the top node of each cluster determined by HPD with an identity threshold, and the sequence-specific coverage, which counts the probes found at the terminal nodes. For ProDesign, we report the coverage with and without clustering (Table 2). Although the different approaches lead to different clustering schemes, we obtained similar coverage of group-specific probes with both programs. Without clustering, the coverage of sequence-specific probes generated by ProDesign was much higher than that found with HPD, indicating that more genes can be detected specifically with ProDesign. We think the

Table 1. Comparison of ProDesign and other oligonucleotide design tools on the 5875 sequences of the *S.cerevisiae* genome

Program	Length	C#[1]	C%[1]	C#[2]	C%[2]	Time (min)
OligoArray2.1	50	5440	92.6	5841	99.4	201.0
OligoPicker	70	5440	92.6	–	–	14.5
YODA	70	5305	90.3	–	–	3.1
ProDesign	20–70	5721	97.4	5841	99.4	2.3

For OligoPicker, OligoArray2.1 and YODA, default parameters were used as much as possible. For ProDesign, the following parameters were used: the probe length, from 20 to 70 bases; G + C content, from 35 to 65%; for all the program, the melting temperature range is 5°C. The initial sequences were not clustered. The coverage (C) in number of genes (#) and percentage of genes (%) is given before (1) and after (2) re-clustering.

Table 2. Comparison of HPD and ProDesign on the *nirS* and *pmoA* datasets

Design method	Type of probe	Number of probes	
		nirS	*pmoA*
HPD (with UPGMA)	Sequence specific	145(145)[a]	63(63)[a]
	Group specific	235(410)[b]	171(479)[b]
ProDesign	Sequence specific	364(364)[a]	323(323)[a]
	Group specific	315(410)[b]	332(480)[b]

[a]Number in parenthesis is the sequence coverage of the sequence-specific probes.
[b]Number in parenthesis is the sequence coverage of the sequence- plus group-specific probes.

spaced seed hashing and the flexibility of probe lengths implemented in ProDesign were the main reasons for the higher coverage obtained.

3.3 Probe design for a microbial community

To examine the performance of ProDesign on a set of genomes, we selected 11 related genomes (*Escherichia coli* CFT073, *E.coli* K12, *E.coli* O157H7, *E.coli* O157H7_EDL933, *E.coli* UTI89, *E.coli* W3110, *Shigella boydii* Sb227, *S.dysenteriae, S.flexneri* 2a, *S.flexneri* 2a 2457T, *S.sonnei* s046) with a total of 51519 sequences, as a mock example of a microbial community. Based on their BLAST pairwise E-values, all the sequences were clustered to 7157 groups obtained with TribeMCL. More than 98% of the clusters have fewer than 14 sequences.

For this large dataset, it was not possible to use HPD for comparison with ProDesign and only ProDesign was used to find probes of length 20–70 bp. A total of 3822 probes were found, corresponding to a coverage of 52.9%. This was improved by re-clustering, to 5723 probes and 79.3% coverage. A very low coverage was obtained for the small number of larger clusters containing 14 or more sequences (27.5 and 40.2% before and after re-clustering, respectively).

This low coverage can be explained because many clusters contain very short sequences of length fewer than 200 bases. Another reason for the low coverage is that some clusters contained sequences that have little or no overlap between them. For the 11 genomes under consideration, the proportion of sequences of length fewer than 200 bases was 12.71%, and the proportion of clusters containing sequences with overlap fewer than 200 bases was 12.38%. Under these circumstances it is difficult to obtain a high coverage, as the sequence space for which probes can be designed to detect the cluster is unavoidably small.

The previous result indicates that the original clustering strategy can also have a strong impact on the probe coverage that can be obtained. To investigate different levels of clustering on probe design, we used CD-HIT to cluster the 11 genomes, which allows us to set various similarity thresholds for the clustering. The effect of similarity threshold on coverage is shown in Figure 4. We see that peak coverage is obtained when setting the similarity threshold to 0.90, and is reduced for lower or higher thresholds of similarity. This is because lower similarity thresholds allow more diverse sequences into a cluster, which makes it harder to find a probe to detect all the sequences of the cluster; on the other hand, by setting a higher similarity threshold, the distance between clusters is smaller, which makes it harder to find probes that are unique to each cluster. The re-clustering strategy used in ProDesign, because it merges close clusters, is not as strongly affected by an overly stringent clustering strategy as it is an overly lenient one. This is shown in Figure 4, where we see that coverage is more strongly increased by re-clustering when the similarity threshold for CD-HIT is high.

There has been extensive work in the area of optimal spaced seed design, and we used seeds suggested by Kucherov *et al.* (2006). When a variety of seed patterns was tried on the 11 genomes (and CD-HIT threshold 0.9), seeds of weight 10 yielded coverages ranging from 80.4 to 84.8% whereas those seeds of weight 11 yielded a coverage of 93.3 to 94.2% (see Fig. 5 and Supplementary Table 2 for details).

Another factor that affects coverage is the allowed range of melting temperatures. When the allowable range was increased from 4 to 10°C, for TribeMCL clusters of the collection of 11 bacterial genomes, coverage was increased from 75 to 90%. In ProDesign, the default temperature range is 5°C, but the actual temperature is automatically chosen by the program to obtain the highest coverage. Because of this flexibility higher coverage can be obtained.

4 CONCLUSION

In summary, in ProDesign we have put forward a basic data structure based on gene groups instead of single sequences to realize the selection of sequence-specific as well as group-specific probes. We also developed a word-based

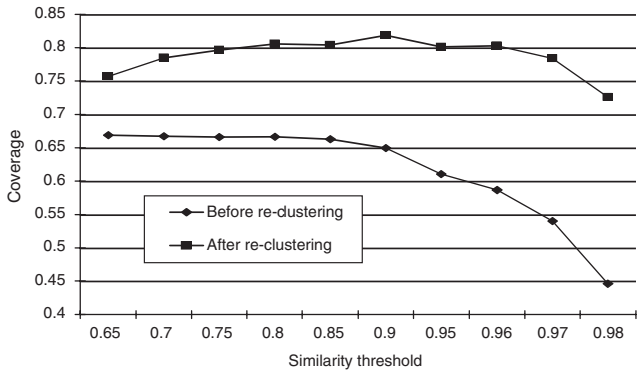

Fig. 4. The effect of the similarity threshold on coverage. The choice of similarity clustering threshold used in CD-HIT affects coverage of the probes that are obtained with ProDesign. By setting a lower similarity threshold, more diverse sequences can go into a cluster, which makes it harder to find a probe to detect all the sequences of the cluster; on the other hand, by setting a higher similarity threshold, the distance between clusters is smaller, which makes it harder to find a probe to distinguish this from other still similar clusters.

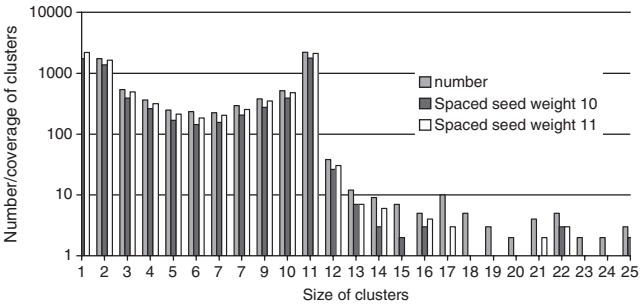

Fig. 5. The effect of spaced seed weight on coverage obtained with ProDesign. Numbers on the *y*-axis represent the total number of clusters, or the number of covered clusters. With spaced seeds of weight 11, we can get more specific words and higher coverage than with seeds of weight 10. This is because higher weight spaced seeds have lower hit probability. The figure also shows that low coverage was obtained for the larger clusters.

heuristic clustering method to improve the probe coverage, and implemented a similarity search algorithm based on spaced seed hashing to cope with large-scale datasets. With these methods, the coverage has been improved without compromising computational time. The program is highly efficient in dealing with complete genomes and even genome communities of tens of thousands of sequences.

For ProDesign, the input dataset is clusters of homologous sequences; therefore the coverage of the probe set generated by ProDesign depends on the clustering strategy and related criteria. Although ProDesign implements a re-clustering strategy to group closely related clusters which can significantly improve coverage, we find that the original input clustering still has a strong effect on the final coverage obtained. In particular, short sequences and sequences with little overlap will result in low coverage. We are currently developing a primary clustering algorithm and sequence filtering criteria optimal for group-specific probe design to subsequently include in the ProDesign package.

Because the user originally determines the clusters, flexibility is maintained for particular requirements of the experimental design that can be addressed in ProDesign. For example, for some gene families it may be important to determine the presence of very close homologs to a particular sequence, whereas for other gene families, detecting any remote homolog is desirable.

The strategy to design group-specific probes to detect gene families instead of single genes is undoubtedly helpful in detecting genes and gene families in highly complex microbial communities. The program provides biologists with a powerful tool for an easy, rapid and flexible design of oligonucleotide probes for environmental microarrays and related applications.

ACKNOWLEDGEMENTS

We thank Bin Ma of the University of Western Ontario for helpful discussion on the spaced seed hashing algorithm, and thank Junni Zhang and Lin Liu of Beijing University for their kind suggestions on the algorithm. We thank Paulo Nuin for design and implementation of the ProDesign interface, and YongBai Xu and Zhuozhi Wang for helpful comments and discussion. We also thank Robert L. Charlebois for critical reading of the manuscript. Grants NSFC 60372040 and NSFC 90612019, State Scholarship Fund Award of China to S.F., and a CIHR grant to E.R.M.T. supported this work.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Ashelford,K.E. *et al.* (2002) PRIMROSE: a computer program for generating and estimating the phylogenetic range of 16S rRNA oligonucleotide probes and primers in conjunction with the RDP-II database. *Nucleic Acids Res.*, **30**, 3481–3489.

Behr,T. *et al.* (2000) A nested array of rRNA targeted probes for the detection and identification of Enterococci by reverse hybridization. *Syst. Appl. Microbiol.*, **23**, 563–572.

Beiko,R.G., Harlow,T.J. and Ragan,M.A. (2005) Highways of gene sharing in prokaryotes. *Proc Natl Acad Sci USA*, **102**, 14332–14337.

Bodrossy,L. (2003) Development and validation of a diagnostic microbial microarray for methanotrophs. *Environ. Microbiol.*, **5**, 566–582.

Brown,D.G. *et al.* (2004) A tutorial of recent developments in the seeding of local alignment. *J. Bioinform. Comput. Biol.*, **2**, 819–842.

Call,D.R. (2005) Challenges and opportunities for pathogen detection using DNA Microarrays. *Critical Reviews in Microbiology*, **31**, 91–99.

Cho,J.C. and Tiedje,J.M. (2001) Bacterial species determination from DNA-DNA hybridization by using genome fragments and DNA microarrays. *Appl. Environ. Microbiol.*, **67**, 3677–3682.

Chou,H.H *et al.* (2004) Picky: oligo microarray design for large genomes. *Bioinformatics*, **20**, 2893–2902.

Chung,W.H. *et al.* (2005) Design of long oligonucleotide probes for functional gene detection in a microbial community. *Bioinformatics*, **21**, 4092–4100.

Emrich,S.J. *et al.* (2003) PROBEmer: a web-based software tool for selecting optimal DNA oligos. *Nucleic Acids Res.*, **31**, 3746–3750.

Enright,A.J. *et al.* (2002) TribeMCL: An eficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.*, **30**, 1575–1584.

Garcia-Vallvé,S. *et al.* (2000) Horizontal gene transfer in bacterial and archaeal complete genomes. *Genome Res.*, **10**, 1719–1725.

Gordon,P.M. and Sensen,C.W. (2004) Osprey: a comprehensive tool employing novel methods for the design of oligonucleotides for DNA sequencing and microarrays. *Nucleic Acids Res.*, **32**, e1331–e133(1–9).

Guschin,D.Y. *et al.* (1997) Oligonucleotide microchips as genosensors for determinative and environmental studies in microbiology. *Appl. Environ. Microbiol.*, **63**, 2397–2402.

Gusfield,D. (1997) *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, NY, USA.

He,Z. *et al.* (2005) Empirical establishment of oligonucleotide probe design criteria. *Appl. Environ. Microbiol.*, **71**, 3753–3760.

Holben,W.E. and Harris,D. (1995) DNA-based monitoring of total bacterial community structure in environmental samples. *Mol. Ecol.*, **4**, 627–631.

Kaderali,L. and Schliep,A. (2002) Selecting signature oligonucleotides to identify organisms using DNA arrays. *Bioinformatics*, **18**, 1340–1349.

Kane,M.D. *et al.* (2000) Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays. *Nucleic Acids Res.*, **28**, 4552–4557.

Keich,U. *et al.* (2004) On spaced seeds for similarity search. *Discrete Appl. Math.*, **138**, 253–263.

Klau,G.W. *et al.* (2004) Optimal robust non-unique probe selection using integer linear programming. *Bioinformatics*, **20**, i186–i193.

Kucherov,G. *et al.* (2006) A unifying framework for seed sensitivity and its application to subset seeds. *J. Bioinform. Comput. Biol.*, **4**, 553–569.

Li,F. and Stormo,G. (2001) Selection of optimal DNA oligos for gene expression arrays. *Bioinformatics*, **17**, 98–99.

Li,W. and Godzik,A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**, 1658–1659.

Li,X. *et al.* (2005) Selection of optimal oligonucleotide probes for microarrays using multiple criteria, global alignment and parameter estimation. *Nucleic Acids Res.*, **33**, 6114–6123.

Liebich,J. *et al.* (2006) Improvement of oligonucleotide probe design criteria for functional gene microarray in environmental applications. *Appl. Environ. Microbiol.*, **72**, 1688–1691.

Ludwig,W. *et al.* (2004) ARB: a software environment for sequence data. *Nucleic Acids Res.*, **32**, 1363–1371.

Ma,B. *et al.* (2002) Patternhunter: faster and more sensitive homology search. *Bioinformatics*, **18**, 440–445.

Markham,N.R and Zuker,M. (2005) DINAMelt web server for nucleic acid melting prediction. *Nucleic Acids Res.*, **33**, W577–W581.

Meier,H. *et al.* (2004) Development and implementation of a parallel algorithm for the fast design of oligonucleotide probe sets for diagnostic DNA microarrays. *Concurr. Comput.*, **16**, 873–893.

Nielsen,H.B. *et al.* (2003) Design of oligonucleotides for microarrays and perspectives for design of multi-transcriptome arrays. *Nucleic Acids Res.*, **31**, 3491–3496.

Noé,L. and Kucherov,G. (2004) Improvement hit criteria for DNA local alignment. *BMC Bioinformatics*, **5**, 149.

Noé,L. and Kucherov,G. (2005) YASS: enhancing the sensitivity of DNA similarity. *Nucleic Acids Res.*, **33**, W540–W543.
Nordberg,E.K. (2005) YODA: selecting signature oligonucleotides. *Bioinformatics*, **21**, 1365–1370.
Pennisi,E. (2004) The Biology of Genomes meeting. Surveys reveal vast numbers of genes. *Science*, **304**, 1591.
Rahmann,S. (2003) Fast large scale oligonucleotide selection using the longest common factor approach. *J. Bioinformatics Comput. Biol.*, **1**, 343–361.
Reymond,N. *et al.* (2004) ROSO: optimizing oligonucleotide probes for microarrays. *Bioinformatics*, **20**, 271–273.
Rhee,S.K. *et al.* (2004) Detection of biodegradation and biotransformation genes in microbial communities using 50-mer oligonucleotide microarrays. *Appl. Environ. Microbiol.*, **70**, 4303–4317.
Rimour,S. *et al.* (2005) GoArrays: highly dynamic and efficient microarray probe design. *Bioinformatics*, **21**, 1094–1103.
Rouillard,J.M. *et al.* (2002) OligoArray: genome-scale oligonucleotide design for microarrays. *Bioinformatics*, **18**, 486–487.
Rouillard,J.M. *et al.* (2003) OligoArray 2.0: design of oligonucleotide probes for DNA microarrays using a thermodynamic approach. *Nucleic Acids Res.*, **31**, 3057–3062.
Steward,G.F. *et al.* (2004) Development and testing of a DNA microarray to assess nitrogenase (nifH) gene diversity. *Appl. Environ. Microbiol.*, **70**, 1455–1465.
Sung,W.K. and Lee,W.H. (2003) Fast and accurate probe selection algorithm for large genomes. In *IEEE Computer Society Bioinformatics Conference (CSB)* In *Proceedings of the CSB2003 IEEE Computer Society*. pp. 65–74.
Thompson,J.D. *et al.* (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.
Tiquia,S.M. *et al.* (2004) Evaluation of 50-mer oligonucleotide arrays for detecting microbial populations in environmental samples. *Biotechniques*, **36**, 664–670, 672, 674–675.
Torsvik,V. and Ovreas,L. (2002) Microbial diversity and function in soil: from genes to ecosystem. *Curr. Opin. Microbiol.*, **5**, 240–245.
Tyson,G.W. *et al.* (2004) Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature*, **28**, 37–43.
Venter,J.C. *et al.* (2004) Environmental genome shotgun sequencing of the Sargasso Sea. *Science*, **304**, 66–74.
Wang,X. and Seed,B. (2003) Selecting of oligonucleotide probes for protein coding sequences. *Bioinformatics*, **19**, 796–802.
Xu,J. *et al.* (2006) Optimizing multiple spaced seeds for homology search. *J. Comput. Biol.*, **13**, 1355–1368.
Zhang,Z. *et al.* (2002) Identification of characteristic oligonucleotides in the bacterial 16S ribosomal RNA sequence dataset. *Bioinformatics*, **18**, 244–250.
Zheng,J. *et al.* (2004) Efficient selection of unique and popular oligos for large EST databases. *Bioinformatics*, **20**, 2101–2113.
Zhou,J. (2003) Microarrays for bacterial detection and microbial community analysis. *Curr. Opin. Microbiol.*, **6**, 288–294.