BIOINFORMATICS APPLICATIONS NOTE Vol. 20 no. 15 2004, pages 2473–2475
doi:10.1093/bioinformatics/bth259

G-PRIMER: greedy algorithm for selecting minimal primer set

Jiren Wang[1,*], Kuo-Bin Li[1] and Wing-Kin Sung[2]

[1]BioInformatics Institute, 30 Biopolis Street, #07-01 Matrix, Singapore 138671 and [2]Department of Computer Science, National University of Singapore, 3 Science Drive 2, Singapore 117543

Received on January 25, 2004; revised on February 27, 2004; accepted on March 9, 2004
Advance Access publication April 8, 2004

ABSTRACT

Summary: G-PRIMER, a web-based primer design program, has been developed to compute a minimal primer set specifically annealed to all the open reading frames in a given microbial genome. This program has been successfully used in the microarray experiment for analyzing the expression of genes in the *Xanthomonas campestris* genome.

Availability: It is available at http://mammoth.bii.a-star.edu.sg/gprimer/. Its source code is available upon request.

Contact: jiren@bii.a-star.edu.sg

DNA microarray technology has received considerable attention in recent years due to its ability to analyze gene expression pattern at genomic scale. An important step of DNA microarray technology is the generation of labeled probes for hybridization. mRNA molecules from bacterial cells are usually reverse transcribed into cDNA by random primers before being hybridized onto the microarray. However, the signal in hybridization result is relatively weak when using random primers to generate labeled cDNAs (Talaat *et al*., 2000).

G-PRIMER is used to generate the minimal primer set (MPS) specifically annealed to all open reading frames (ORFs) in a given microbial genome to improve the hybridization signals of microarray experiments.

Here, the MPS problem is formulated as the set covering problem (SCP) described below: suppose $S = \{S_1, S_2, \ldots, S_n\}$ be the set of all ORFs in a given bacterial genome, and L be the length of the primers. The MPS problem is to find a minimal number of primers P_1, $P_2, \ldots$, and P_k, whose lengths are equal to L, that can cover all ORFs S_1, $S_2, \ldots$, and S_n. In this paper, it is said that a primer covers an ORF if this primer is a substring of the reverse complementary sequence of this ORF at the specified search window from the 3′ end. The ORF can be reverse transcribed into cDNA by using this primer.

Suppose C_1, $C_2, \ldots$, and C_m, called primer candidates whose length is equal to L are all unique substrings of the reverse complementary sequences of S_1, $S_2, \ldots, S_n$ at the specified search window from the 3′ end, and a zero–one matrix $M[1..n, 1..m]$ is constructed, where n and m are the number of rows and columns, respectively. Each row represents an ORF S_j $(1 \leq j \leq n)$ while each column represents a primer candidate C_i $(1 \leq i \leq m)$. If C_i covers an ORF S_j then $M(j,i) = 1$, otherwise $M(j,i) = 0$. Here, only the information related to elements '1' in the zero–one matrix will be stored to save space.

Now the MPS problem becomes finding the minimal subset of $\{C_1, C_2, \ldots, C_m\}$ that covers all ORFs S_1, $S_2, \ldots$, and S_n. As the MPS problem can be formulated to the SCP and the SCP is NP-hard (Cormem *et al*., 1990), the MPS problem is also NP-hard. Thus, a greedy algorithm is used to find the MPS.

The proposed greedy algorithm, G-PRIMER, is divided into three main steps described below.

The first step is identifying each primer candidate and a list of the ids of ORFs covered by this primer candidate by checking the reverse complementary of each L-mer in all ORFs in the specified search window at the 3′ end and storing the above information in two hash tables named as Counter and Cover. Each key of Counter and Cover represents a primer candidate while its corresponding values are equal to the number of ORFs covered by this primer candidate, and the list of ids of ORFs covered by this primer, respectively. The chosen hash tables are very suitable for solving the MPS problem as the repeated primer candidates within one ORF and among ORFs can be easily eliminated. Moreover, it only takes $O(1)$ time to check the existence of a primer and to retrieve its corresponding list of ids of ORFs covered by this primer.

Suppose the lengths of S_1, $S_2, \ldots$, and S_n are equal to L_1, $L_2, \ldots$, and L_n, the length of primers is equal to L and L_G is equal to $L_1 + L_2 + \cdots + L_n$, the size of all ORFs, then the total number of unique candidate primers is less than $(L_1 - L + 1) + (L_2 - L + 1) + \cdots + (L_n - L + 1) = L_G - n * (L - 1)$ due to the repeated candidate primers. Thus, both the total number of keys, i.e. the primer candidates, in two hash tables and the total number of the ORF ids in the elements of Cover,

*To whom correspondence should be addressed.

Table 1. Experimental result

Primer length	Without biological constraints		With some biological constraints specified	
	No. of primer candidates	No. of primers	No. of primer candidates	No. of primers
6	2797	15	1966	16
8	59 929	67	28 510	77
10	4 53 505	247	2 14 827	318
12	12 53 717	712	6 83 802	817
14	17 23 682	1473	8 24 721	1747

i.e. the total number of '1' elements in the zero–one matrix, are less than L_G.

The second step is choosing the primer candidate with the maximal number of ORFs covered by this primer candidate from Counter as a primer.

It was observed that there would be cases when two or more chosen primer candidates cover the same number of ORFs but different ORF sets. In such a situation, choosing a different one of these primer candidates may produce different number of primers in the final results.

Hence, the following heuristic is proposed to choose the better primer candidate for the case where two or more chosen primer candidates cover the same number of ORFs. If there are more than five chosen primer candidates covering the same number of ORFs, only the first five chosen primer candidates will be checked further to reduce the searching time. For each of these primer candidates, the maximal number of elements in set difference between the set of ids in each element of Cover and the set of ids whose corresponding ORFs covered by this primer is calculated. Then the primer candidate that produces this maximal number is selected as the primer. If there are two or more such primer candidates, the first one processed becomes the primer in the proposed algorithm without further searching.

The third step is updating Counter and Cover. Once the primer candidate is selected as a primer, the hash table elements whose keys are equal to this primer are deleted from Counter and Cover. Meanwhile, the ids of all ORFs covered by this primer are removed from the id list in all elements of the Cover. The affected elements of Counter are re-calculated based on the current number of ids of ORFs covered in the corresponding elements of Cover.

The second and third steps will be repeated until all ORFs are covered by the chosen primer set.

The inputs of G-PRIMER include the nucleotide sequences of all the ORFs (in FASTA format) in a given microbial genome, the primer length, the size of search window from the 3′ end of ORFs, and biological constraints, and its output is the MPS. Biological constraints can be optionally specified by users and applied to the primer candidates. Currently, biological constraints include the percentage range of GC content, deleting self-complementary primers to avoid hybridization of primers with themselves, and forming the 'G/C' clamp at the 3′ end of a primer to ensure a tight localized hybridization bond.

Table 1 shows the experimental result by using the proposed algorithm to generate the MPS of *Xanthomonas campestris* genome. The size of all ORFs is 4.3 MB while the number of ORFs is 4181. The size of search window from the 3′ end of ORFs is set to 500. The biological constraints include setting 30–70% GC content, deleting self-complementary primers, and forming the 'G/C' clamp at the 3′ end.

Both the cost and the specificity of a primer are proportional to its length. Moreover, the longer the primer length is, the more primers are required to cover all ORFs in a genome as shown in Table 1, and thus more money is required to synthesize these primers. Users need to balance the cost with the biological specificity to choose a specific primer length.

Compared with the method of Talaat *et al*. (2000), G-PRIMER can deal with the cases when two or more chosen candidate primers cover the same number of ORFs. Moreover, the biological constraints specified by users can be used to filter unwanted primer candidates. Unlike the method of Hsieh *et al*. (2003), where there are one or two mismatches between the generated primers and the covered nucleotide sequences, G-PRIMER generates the MPS that can anneal to all ORFs in a given microbial genome without any mismatch between them.

In the proposed algorithm there is some restriction either on the size of total ORFs or the length of primers as the space complexity of G-PRIMER is the minimum of L_G and 4^L, where L is the length of the primers. Currently this algorithm works well for most of the microbial genomes as the sizes of total ORFs in these genomes are relatively small.

G-PRIMER provides a web-based tool for microbiologists to generate the MSP used for genome-directed primer design. By increasing the memory capacity, this algorithm can be applied to generate the MSP for other large genomes as well.

ACKNOWLEDGEMENT

The authors are grateful to reviewers for the useful comments.

REFERENCES

Cormen,T.H., Leiserson,C.E. and Rivest,R.L. (1990) *Introduction to Algorithms*. MIT Press.

Hsieh,M.-H., Hsu,W.C., Chiu,S.K. and Tzeng,C.M. (2003) An efficient algorithm for minimal primer set selection. *Bioinformatics*, **19**, 285–286.

Talaat,A.M., Hunter,P. and Johnston,S.A. (2000) Genome-directed primers for selective labeling of bacterial transcripts for DNA microarray analysis. *Nat. Biotechnol.*, **18**, 679–682.