An efficient algorithm for minimal primer set selection
Ming-Hua Hsieh, Wei-Che Hsu, Sung-Kay Chiu and Chi-Meng Tzeng*

Department of Research and Development, U-Vision Biotech Inc., 3F 132, Ln 235, Pao-Chiao Rd, Hsin-Tien City 231 Taipei, Taiwan

Received on June 12, 2002; revised on July 29, 2002; accepted on August 6, 2002

ABSTRACT
Summary: We have developed U-PRIMER, a primer design program, to compute a minimal primer set (MPS) for any given set of DNA sequences. The U-PRIMER algorithm, which uses automatic variable fixing and automatic redundant constraint elimination to tackle the binary integer programming problem associated with the MPS selection problem. The program has been tested successfully with 32 adipocyte development-related genes and 9 TB-specific genes to obtain their respective MPSs.
Availability: A free copy of U-PRIMER implemented in C++ programming language is available from http://www.u-vision-biotech.com
Contact: cmtzeng@u-vision-biotech.com

Extensive diagnostic and molecular analyses are often restricted by limited availability of biological materials. Multiplex RT-PCR is a time and reagent saving amplification technique in which multiple primer sets are used to amplify several specific targets simultaneously from a single sample. In microarray experiments, mRNA molecules from cells are reverse transcribed into cDNA by random primers before being hybridized onto the microarray. However, the cDNA generated by random primers gives a significant level of cross hybridization because of the presence of homologous sequences or genes with similar sequences in the genome. One way to overcome this problem is using RT-PCR amplification of the transcripts of interest. In the case of many genes to be monitored in a microarray experiment, many primer pairs are needed for the RT-PCR reactions. If the primer for one gene can be used to prime another gene, called a universal primer, a smaller primer set is thus needed. With this idea, multiple PCR kit and microarray could be more cost effective, convenient, and accurate. Let a complete primer set (CPS) be a primer set that contains primers that can amplify all the DNA sequences as input, and screens primer candidates for both ends of each sequence of any given number of DNA sequences. Since CPS selection is the same in the MPS problem, it is sufficient to consider only the CPS selection problem. Let m be the length of primers and w(i) be the length of the forward primer window of sequence i. We say a primer candidate p is eligible for a DNA fragment if its complementary sequence is a substring of the DNA fragment. However, to increase the chance of finding the universal primers, we allow a maximum of 2 base-pair mismatches between the primer candidate and the DNA fragment at user’s choice. We call a forward primer candidate p_f eligible for a target DNA sequence if p_f is eligible for the 5′ end fragment (forward primer window) of the DNA sequence. Let P_f be a forward primer set. Then P_f selection can be informally stated as follows.

\[
\min_{P_f} |P_f|
\]

subject to  for all s ∈ S, there exists a p_f ∈ P_f such that p_f is eligible for 5′ end of s

where |P| denotes the number of elements of a set P. To formally define this minimization problem, we define primer candidates first. For each forward primer window,
we move the primer candidate one base at a time from the beginning of the window with fragment length $m$ until we run through the whole window. For example, if only a perfect match is allowed, there will be $M(0) = \sum_{i=1}^{n} (u_{i}(i) - m + 1)$ such fragments, and if up to two mismatches are allowed, there will be $M(2) = \sum_{i=1}^{n} (1 + 3m + 9m(m-1)/2) (u_{i}(i) - m + 1)$ such fragments. Let $P_{j}$ be the set of the forward primer candidates defined above and $C_{j}$ be the cover matrix of $P_{j}$. The entries of $C_{j}$ are defined as

$$C_{j}(i, j) = \begin{cases} 1, & P_{j}(j) \text{ is eligible for 5' end of } S(i); \\ 0, & \text{otherwise}. \end{cases}$$

Where $P_{j}(j)$ denotes the $j$th element of $P_{j}$ $(1 \leq j \leq k_{f})$ and $S(i)$ denotes the $i$th element of $S$ $(1 \leq i \leq n)$. Set $k_{f} = |P_{f}|$ and $x_{j}$'s $(1 \leq j \leq k_{f})$ be the decision variables for $P_{j}$. In particular, $x_{j} = 1$ if $P_{j}(j)$ is selected; otherwise $x_{j} = 0$. Then, the above minimization problem can be formulated as an binary integer programming (BIP) problem (see Hillier and Lieberman (2001) for more details)

$$\min_{x_{1}, \ldots, x_{k_{f}}} \sum_{j=1}^{k_{f}} x_{j}, \quad (P1)$$

subject to $C_{j}x \geq 1_{n}, \quad x_{j} = 0 \text{ or 1, } \quad 1 \leq j \leq k_{f},$

where $x = (x_{1}, x_{2}, \ldots, x_{k_{f}})^{T}$ and $1_{n}$ denotes the $n$-vector with all elements equal to $1$.

The main difficulty of computing the exact solution of (P1) is that the number of decision variables is large even when $n$ and $m$ are moderately small; e.g. for $n = 32$, $m = 15$, and $u_{i}(i) = 100 \ (i = 1, \ldots, n)$, then $M(2) = 2727232$. It is well known that a general BIP problem with more than 1000 decision variables is computationally intractable unless a special algorithm is designed to exploit the structure of the BIP problem. To solve this problem, we adopt automatic variable fixing and automatic redundant constraint elimination techniques (see, Crowder et al. (1983)). We summarize the basic idea of the U-PRIMER algorithm as follows.

1. for $k = 1$ to $k_{f}$ do
   1) for $j \neq k$ do
      if $c_{j} \geq c_{k}$ (where $c_{j}$ and $c_{k}$ denote the $j$th and $k$th column of $C_{j}$) set $x_{k}$ to 0
   2) for $k = 1$ to $k_{f}$ do
      if $\sum_{j=1}^{k-1} C_{j}(i, j) = 1$ and $C_{j}(k, j)$ = 1
      set $x_{k}$ to 1
3. for all the $x_{k}$ that is fixed to 1 at Step 2, remove constraint $i$ if $C_{i}(i) = 1$. If there is any variable that can be fixed in Step 1 or 2 go to 1
4. if all variables have been fixed and then the algorithm will be terminated; otherwise, use a regular BIP solver (such as a branch- and bound-solver) for the remaining variables.

Step 1 and 2 are automatic variable fixing. $x_{k}$ can be fixed to zero if we can find a $j \neq k$, such that $c_{j} \geq c_{k}$. Since if \{\ldots, $k$, \ldots\} is an MPS, then \{\ldots, $j$, \ldots\} will be an MPS, too. On the other hand, $x_{k}$ = 1 if $\sum_{j=1}^{k-1} C_{j}(i, j) = 1$ and $C_{j}(k, j)$ = 1, since any set without $k$ can not be a CPS and thus not an MPS. Step 3 is an automatic redundant constraint elimination.

In our Type II diabetes (NIDDM) research, we used U-PRIMER program to design an MPS for 32 genes involved in adipocyte differentiation. Also, in another application, a minimal primer set is needed for the PCR amplification of nine novel markers from the genome of Mycobacterium tuberculosis (or TB), generated by U-GET (U-Vision Biotech Inc.), a computer program that can find unique sequences in the genome of any species. With a choice of primer-size 15 and one allowable base-pair mismatch, 6 pairs of primers from the $S'$ and $S'$' ends for the 9 U-GET verified sequences were found and evaluated. The combination of U-PRIMER program to generate genespecific primers and the selection of unique sequences in a genome by U-GET program makes the multiplex PCR reaction specific and accurate in clinical diagnosis.

There are several previous studies for the MPS problem: Pearson et al. (1996) and Talaat et al. (2000). They considered primer lengths from 5 or 12. By probability, the chance of matching of a random 8mer is 61 times in a 4 Mb genome. Thus the primer set of short primers is not specific enough in real biological applications. In real world applications, short primers cannot generate unique PCR products because they can cross-prime to many other sites (homologues). We increase the size up to 17 basepairs, which can enhance the specificity for the priming reaction. We also allow one to two mismatches between the primers and the DNA sequence, for increasing the chance to find a second priming site without sacrificing the specificity.

**ACKNOWLEDGEMENTS**

This work was partly supported by a fund from SBIR Grant, MOEA, Taiwan (No. 12900033).

**REFERENCES**


