A Computer Program for Generating Gene-Specific Fragments for Microarrays

Dong Xu[1] xud@ornl.gov
Ying Xu[1] xyn@ornl.gov
Gary Li[2] iqu@ornl.gov
Jizhong Zhou[2] zhouj@ornl.gov

[1] Computational Biosciences Section, Life Science Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[2] Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

Keywords: microarray, hybridization, primer design, sequence alignment, dynamic programming

1 Introduction

Various genome-scale sequencing projects create vast amount of genomic data. The next key step is to define the function of each gene. One of the powerful tools to investigate gene function is DNA microarray, which is based on DNA-DNA hybridization at high throughput mode [2]. If there is a certain degree of DNA sequence similarity between two genes, both of them may give hybridization signal with the same probe on a microarray. In this case, it is difficult to assign functions for them according to array data. To target a particular gene, it is necessary to design a gene-specific probe, i.e., a segment of a sequence that does not have high sequence similarity to any other sequence in the sequence pool. Traditionally this task is carried out manually through visually checking the output of sequence alignments. It takes a long time for researchers and the results may not be reliable. We have developed a computer program that solves this problem reliably and efficiently. This program is useful in selecting gene-specific fragments, which can be used as the input to design PCR primer pair for PCR amplification and microarrays.

2 Method

2.1 Mathematical Formulation

The problem can be formulated as follows. Assume we have a collection of N sequences S, which consists of $s^1, s^2, ..., s^N$. For sequence s^m $(1 \leq m \leq N)$ with n_m genetic bases s^m_k $(1 \leq k \leq n_m)$, the goal is to find the longest segment of genetic bases $s^m_{a,b}$, i.e., $a \leq k \leq b$, such that the following three conditions are satisfied: (1) the maximum sequence similarity between $s^m_{a,b}$ and s^k $(k \neq m)$ is smaller than a user specified value sim_{max}; (2) the segment has at least l_{min} bases, i.e., $b - a > l_{min}$; (3) if s^m and any s^k $(k \neq m)$ have a local sequence alignment with a minimum length of f_{mix} and an expectation value of E_{max} or less, the aligned portion of s^m should not overlap with the $s^m_{a,b}$ segment.

2.2 Algorithm

Our algorithm combines the fast local alignment using the BLAST [1] for all the pairwise sequence comparisons, and a global alignment using dynamics programming [4], which is slow but guarantees to find a global optimal solution, for selected sequence comparisons. The following steps are carried out to solve the problem formulated above:

- Carry out a local alignment for each s^m $(1 \leq m \leq N)$ against the whole sequence database S using the BLAST to find all the local alignment with an expectation value of E_{max} or less. If no such a local alignment is found other than the self alignment, we will use the whole sequence of s^m as the gene-specific fragment. Otherwise it is included in the sequence set Q.

- Assume a sequence q^u with n_u genetic bases in Q finds local alignments (other than the self alignment) with w_u sequences p_v^u $(1 \leq v \leq w_u)$ that have an expectation value of E_{max} or less. We carry out a global alignment between q^u and each p_v^u using dynamics programming. For a segment $q_{i,j}^u$ $(1 \leq i \leq n_u - l_{min} + 1,\ i + l_{min} - 1 \leq j \leq n_u)$ in q^u, if the three conditions described in **Mathematical Formulation** are satisfied for the global optimal alignment between q^u and every p_v^u $(1 \leq v \leq w_u)$, $q_{i,j}^u$ is recorded as a possible gene-specific fragment. Then we choose the longest segment $q_{a,b}^u$ among all possible solutions. If no $q_{i,j}^u$ is found to satisfy the three conditions, the user may need to adjust the specified parameters to redo the calculation, or it may indicate that two sequences are too similar so that finding a gene-specific fragment is impossible.

- Verify whether $q_{a,b}^u$ can have an alignment (other than the global optimal alignment) with a p_v^u that yields a sequence similarity of sim_{max} or larger. We carry out the global alignment between $q_{a,b}^u$ (instead of q^u in the last step) and each p_v^u $(1 \leq v \leq w_u)$ using dynamics programming. If none of p_v^u has a sequence similarity of sim_{max} or larger, then $q_{a,b}^u$ is selected as the gene-specific fragment for q^u. Otherwise we use the next longest segment in the possible gene-specific fragments of $q_{i,j}^u$ to check. However, the latter case is rare, since we have not found it in our preliminary test.

3 Implementation

The computer program is written in the C programming language. It has been tested on various Unix/Linux platforms, including Silicon Graphics, Sun, DEC, and Pentium PC. On average, it takes a few seconds for each sequence to find a solution. The program reads a file of all sequences in the FASTA format. A user has options to specify parameters either at the command line or in a parameter file. The default values for the parameters are

$$sim_{max} = 80\%; \quad l_{min} = 100; \quad E_{max} = 10^{-15}; \quad f_{mix} = 50.$$

We are using the program to design gene-specific fragments in actually experiments. Preliminary results indicate that it meets the need for the design. The program also links to the *primer3* program [3] for the automated primer design.

References

[1] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J., Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Res.*, 25:3389–3402, 1997.

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

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

[4] Smith, T. F. and Waterman, M. S., Comparison of Biosequences, *Adv. Appl. Math.*, 2:482–489, 1981.