BIOINFORMATICS APPLICATIONS NOTE
Vol. 19 no. 18 2003, pages 2486–2488
DOI: 10.1093/bioinformatics/btg350

PRIMEX: rapid identification of oligonucleotide matches in whole genomes

Matej Lexa[1,*] and Giorgio Valle[2]

[1]Laboratory of Functional Genomics and Proteomics, Masaryk University Brno, Kotlarska 2, 61137 Brno, Czech Republic and [2]CRIBI Biotechnology Center, University of Padova, via U.Bassi, 58/b, 35131 Padova, Italy

Received on March 24, 2003; revised on May 30, 2003; accepted on July 1, 2003

ABSTRACT

Summary: PRIMEX (PRImer Match EXtractor) can detect oligonucleotide sequences in whole genomes, allowing for mismatches. Using a word lookup table and server functionality, PRIMEX accepts queries from client software and returns matches rapidly. We find it faster and more sensitive than currently available tools.

Availability: Running applications and source code have been made available at http://bioinformatics.cribi.unipd.it/primex

Contact: m-lexa@sci.muni.cz

INTRODUCTION

The latest developments in genomics have provided researchers with a number of whole-genome sequences and a new generation of bioinformatic tools. The prediction of non-trivial PCR amplification products from genomic DNA is a field that despite its considerable practical value, has not been satisfactorily investigated at the bioinformatic level.

To predict the outcome of an arbitrary PCR reaction, it is crucial to identify all the potential priming sites for the primers in use. The search for candidate matches must be sensitive enough to extract all the relevant candidate sequences. At the same time it should be relatively fast, because with large genomes it tends to be the time-limiting step in PCR simulation. Using BLAST in our first algorithms to predict PCR products (VPCR 1.0; Lexa *et al*., 2001), we realized several shortfalls of that program for our purposes. We decided to write a new program that would be better suited for this purpose. The new program called PRIMEX (PRImer Match EXtractor) may also find applications in other areas, such as oligonucleotide probe selection for hybridization experiments or genome alignment.

We provide interfaces allowing one to rapidly query whole genomes for occurrences of short oligonucleotide sequences with mismatches. These include a CGI script, a Perl script with a developer's library and a standalone C++ program. We propose a distributed network of PRIMEX servers to provide search capabilities for most of the sequenced genomes in a short period of time.

SYSTEM AND METHODS

PRIMEX has been written in C++, compiled with gcc and executed under Debian Linux 2.4.16 on a dual-processor Athlon 1600 machine. The server/client functions of the program are built around a socket library written by Tougher (2002, http://www.linuxgazette.com/issue74/tougher.html).

ALGORITHM

In designing the algorithm, we were inspired by some of the ideas behind BLAT (Kent, 2002) and SSAHA (Ning *et al*., 2001) that give our software speed, such as lookup table use and server functionality.

Lookup table

The program creates a word-sized window positioned at the first nucleotide of the searched genomic sequence. It moves the window along the sequence, recording the position of each word encountered into a lookup table. The lookup table is an array of lists. The indices of the array represent the words, while the lists contain all the positions at which the word has been encountered. Lookup table may be saved for future use.

Queries

The program enters a server mode and starts listening to queries. After receiving a query, it extracts words from the query sequence and searches for matches, using the lookup table. Pre-defined parameters set the allowed number of mismatches to be tolerated. The collected matches are then filtered for the allowed number of mismatches, duplicates are eliminated and the result is written out.

[*] To whom correspondence should be addressed at CRIBI Biotechnology Center, University of Padova, via U.Bassi, 58/b, 35131 Padova, Italy.

Table 1. Performance of various search programs when looking for oligonucleotide AAAAAATGATCAATTTACAT in the *A.thaliana* genome

Program	Total	Mismatches							Search time (s)
		0	1	2	3	4	5	6	
BLAST	162	1							12
BLAST-O	2	1							5
FASTA	72	1	0	4	13	23	17	12	53
FASTA-O	1	1							19
BLAT	0	0							80
SSAHA	0	0							71
SSAHA-O	3	1							10
CGC FP-O	1	1							10 (estimate)
EMBOSS	983	1	0	5	104	86	8		18
EMBOSS-O	1	1							14
TACG	1	1							49
AGREP	1204	1	0	5	100	779	3632		34
AGREP	1	1							2
PRIMEX (3)	4517	1	0	14	199				56
PRIMEX-S (5)	12140	1	0	14	199	1686	10240		19
with insertions	24902	1							38
PRIMEX-S (4)	1900	1	0	14	199	1686			4
with insertions	3621	1							12
PRIMEX-S (3)	214	1	0	14	199				1
with insertions	286	1							1
PRIMEX-S (2)	15	1	0	14					<1
with insertions	19	1							<1
PRIMEX-S (1)	1	1	0						≪1
PRIMEX-SO (0)	1	1							≪1

The -O suffix represents searches for high-similarity matches. The -S suffix indicates that the program ran in server mode. The numbers in parentheses are mismatch limits. References: BLAST (Altschul *et al*., 1990), FASTA (Pearson and Lipman, 1988), BLAT (Kent, 2002), SSAHA (Ning *et al*., 2001), CGC FindPatterns (Accelrys, San Diego, CA, USA), EMBOSS Fuzznuc (http://www.hgmp.mrc.ac.uk/Software/EMBOSS/), TACG (Mangalam, 2001) and AGREP (Wu and Manber, 1994).

IMPLEMENTATION

Important server functions

query_remote S	Find matches for oligonucleotide S
dump_state_remote	Report the current settings of the server
get_seq_remote M:N	Report the sequence between M and N

The results of a query are returned in lines. Each line contains the following data: primer number, query sequence, matched sequence, clone name, position within the clone, orientation and number of matching basepairs. For example:

```
0 AAAAATTTTTCCCCCGGGGG AAAAATTCTGCC-ACCGGGGG
  15237134 111102206 + 17
```

Performance

We carried out a series of performance tests. Table 1 shows, which of the related routines were able to find an exact or approximate match to a 20 bp oligonucleotide in the more than 100 MB-long genomic sequence of *Arabidopsis thaliana*. It also lists the times in seconds necessary to provide the answers. Without the server speed-up, AGREP was the best program. However, this program does not provide server functionality that could be used to accelerate the search in DNA sequences, therefore, PRIMEX remains currently the only high-speed choice for oligonucleotide searches in whole genomes.

DISCUSSION

The future of PRIMEX depends on the applications that could benefit from its abilities. For instance, primer design software may query the server and check primers against whole genomes rapidly. PCR simulation software (Rubin and Levy, 1996; Lexa *et al.*, 2001) could use PRIMEX. Genome alignment and analysis software could use repeated PRIMEX queries to find large-scale similarities between two or more genomes.

We currently run two PRIMEX servers. A master-list file defining how to access these services is available at http://bioinformatics.cribi.unipd.it/primex/primex_master.txt

Current example: *147.162.3.227|30000|Arabidopsis thaliana |10|NCBI A.thaliana genome release 147.251.24.2|30000| Heliobacter species|8|NCBI Helicobacter sp. genome release*

This file is also consulted by the Perl PRIMEX communication library available for download from our website. Each line specifies the IP number, port, organism, word size and optional notes for running servers. Please, contact the authors, if would like to include your server.

REFERENCES

Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Kent,W.J. (2002) BLAT—the BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.

Lexa,M., Horak,J. and Brzobohaty,B. (2001) Virtual PCR. *Bioinformatics*, **17**, 192–193.

Mangalam,H.J. (2001) tacg—a grep for DNA. *BMC Bioinformatics*, **3**, 8.

Ning,Z., Cox,A.J. and Mullikin,J.C. (2001) SSAHA: a fast search method for large DNA databases. *Genome Res.*, **11**, 1725–1729.

Pearson,W.R. and Lipman,D.J. (1988) *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Rubin,E. and Levy,A.A. (1996) A mathematical model and a computerized simulation of PCR using complex templates. *Nucleic Acid Res.*, **24**, 3538–3545.

Tougher,R. (2002) Linux socket programming in C++. Linux Gazette 74.

Wu,S. and Manber,U. (1994) A fast algorithm for multi-pattern searching. Technical Report, The Computer Science Department, The University of Arizona.