me-PCR: a refined ultrafast algorithm for identifying sequence-defined genomic elements

Kevin Murphy1, Towfique Raj1, R. Scott Winters1 and Peter S. White1,2,∗

1Division of Oncology, Children’s Hospital of Philadelphia and 2Department of Pediatrics, University of Pennsylvania, 324 S. 34th Street, Rm 1407 CHOP North, Philadelphia, PA 19104-4318, USA

Received on July 31, 2003; revised on August 27, 2003; accepted on September 29, 2003
Advance Access publication January 22, 2004

ABSTRACT
Summary: We have adapted the originally described electronic PCR (e-PCR) algorithm to perform string searches more accurately and much more rapidly than previously possible. Our implementation [multithreaded e-PCR (me-PCR)] runs sufficiently fast to allow even desktop machines to query quickly large genomes with very large genomic element sets. In addition, me-PCR is multithreaded, interprets all IUPAC nucleotide symbols, allows searches with elements specified by long sequences (such as SNPs), accepts ranges in the expected PCR size input field, requires substantially less memory for analysis of large sequences and corrects a number of minor flaws causing misreporting of hits in exceptional cases. Thus, me-PCR provides increased annotation capabilities for complex genomes to non-expert laboratories.

Availability: me-PCR has been compiled for Linux, Solaris, AIX, Windows and Macintosh platforms. Both source and executable code are freely available to the research community at http://genome.chop.edu/mePCR/

Contact: white@genome.chop.edu

INTRODUCTION
The rate of generation of large genomic sequence tracts for complex organisms is increasing dramatically. This has created an acute need for more rapid methodologies to identify and position pre-determined, sequence-defined genomic elements, especially from smaller laboratories. Traditionally, these tasks have been performed using sequence alignment tools such as BLAST or BLAT (Altschul et al., 1990; Kent, 2002). However, sequence alignment is inefficient as a placement tool, especially where elements are well-defined at the sequence level. This makes routine placement of large element sets, such as complete SNP or marker collections, difficult or impossible for many laboratories. Another disadvantage of alignment algorithms is that the results require significant post-processing, due to their high sensitivity.

ALGORITHM
The central e-PCR algorithm allows rapid identification of sequence elements defined by oriented primer sequences (STSs) flanking a genomic element of specified length or length range. The algorithm scans the database sequence, and at each base position, it is able to determine via a single hash lookup whether the left primer of any STS contains W adjacent bases that match the database sequence at that position. For each such match, direct string comparison is used to determine whether the entire left primer matches at any of the allowed corresponding positions (determined by the known amplimer size range and a margin parameter M). The reason why this simple approach works so well is that the two primers that determine an STS contain W adjacent bases that match the database sequence at that position. For each such match, direct string comparison is used to determine whether the entire left primer matches at that position, and, if so, whether the right primer also matches at any of the allowed corresponding positions (determined by the known amplimer size range and a margin parameter M). The reason why this simple approach works so well is that the two primers that determine an STS are short (averaging around 21 bases each), and the hash word size W can be relatively long compared

588 Bioinformatics 20(4) © Oxford University Press 2004; all rights reserved.
with the length of a primer, leading to efficient hash lookups (i.e. on average, not many STSs share the same hash values), and fast string comparisons.

The algorithm can be expressed in more detail as follows. For each STS primer, a hash word consisting of $W$ adjacent bases compressed into a 32-bit integer is computed, where each base occupies two binary bits. Pointers to the STSs are placed in a hash table, an array of $X$ pointers where $X$ is the number of possible hash words ($2^{2W}$). Primers sharing the same hash value are joined by a linked list extending from the hash table entry.

When searching, the algorithm slides a $W$-sized window across the underlying sequence. For each $W$-sized window, the algorithm computes the hash value and determines which primers have a matching hash word via a simple array index of the hash table. If multiple primers match, the algorithm looks at each in turn by following a linked list from the main hash table entry. If the hash value of a primer matches, a direct string comparison is made against the entire primer. If this comparison succeeds, then direct string comparisons of the second primer against the underlying sequence are made. The second primer has an expected offset relative to the first primer (implied by a known PCR amplicon size), but the margin parameter $M$ allows the position of the second primer to vary relative to the first, at the cost of $2 \times M$ additional string comparisons. The $N$ parameter specifies the cumulative number of allowed mismatches in both primer sequences. The algorithm’s memory usage is primarily determined by the target sequence and hash table (16 MB for me-PCRs default hash word size of 11).

me-PCR uses the same core algorithm and has the same theoretical complexity as e-PCR but differs in certain details that increase performance and sensitivity. The maximum hash word size has been increased from 8 to 16 nt, which results in enormous performance gains at the cost of additional memory usage. Another algorithmic change allows me-PCR to choose a hash word anywhere in the primer, as opposed to strictly at the 5’ end, which allows it to search for almost all STSs with indeterminate nucleotides (Ns). An additional sensitivity-increasing change allows me-PCR to identify correctly multiple STS hits within locally duplicated or repetitive sequence search windows. Various errors in the original implementation of e-PCR have been corrected, including one which resulted in the implementation having a different complexity than intended. It should be noted that this error was subsequently repaired in e-PCR after the development of me-PCR.

**ADDITIONAL NEW FEATURES**

In addition to the enhancements described above, me-PCR offers several additional features not ordinarily thought of as algorithmic changes:

- Multithreading. The new $T$ parameter controls the number of threads used during the search. For computers with multiple CPUs, increasing $T$ increases the number of CPUs that can be allocated to the algorithm.
- Optional interpretation of IUPAC nucleotide base symbols in STSs. The new I parameter allows me-PCR to interpret ambiguous nucleotide symbols in STSs.
- Reporting of sequence orientation of hits.
- User-selectable default STS size. The new Z option assigns a default STS size if left unspecified.
- Ability to handle large primers (>100 bp; e.g. SNP flanking sequence).
- Ability to specify a margin ($M$) value of 0.
- Reporting of invalid command-line parameters as errors.

**INPUTS AND OUTPUTS**

me-PCR requires an input target DNA sequence in FASTA format and an input STS file corresponding to the UniSTS format. Command-line execution requires the specification of program options, STS text file path and sequence file path. In addition to the $M, N$ and $W$ parameters, several additional options are available, including the number of threads ($T$), output file name ($O$), message verbosity ($Q$), maximum STS line length ($S$), default amplimer size ($Z$) and IUPAC flag ($I$). More detailed descriptions can be found at [http://genome.chop.edu/mePCR](http://genome.chop.edu/mePCR).

**PERFORMANCE**

me-PCR was compared with e-PCR v1.2 ([ftp://ftp.ncbi.nih.gov/pub/schuler/e-PCR/](ftp://ftp.ncbi.nih.gov/pub/schuler/e-PCR/)), which addresses many of e-PCR v1.0’s limitations but is unpublished. Both programs outperformed e-PCR v1.0 by two orders of magnitude. me-PCR performed up to 10 times faster than e-PCR v1.2 for large genomic sequences. A word size of 11 (possible only with me-PCR) was determined to be optimal. Using a search margin of 1000 bases and a word size of 11, me-PCR running on a single 750 MHz RS64 IV CPU identified sequence positions for 130,650 STSs in the entire human genome in 53.1 min. A similar single-processor-based search for 3.5 million STSs took 277.2 min. Using a margin of 50 increased this performance 263% at the cost of 2.4% fewer localizations, while using eight threads further increased elapsed performance 320%. Using parameter settings optimal for e-PCR v1.2 ($W = 8$, $M = 10$ and a single thread), me-PCR was 9% faster for the human genome. When me-PCR was used with $W = 11$ and a single thread, it was 9.6 times faster than e-PCR v1.2. Finally, me-PCR’s use of multithreading allows the user to take advantage of multiple CPUs with maximum efficiency and convenience.

We also compared me-PCR with the specialized sequence alignment algorithm SSAHA ([Ning et al., 2001](http://onlinelibrary.wiley.com/doi/10.1002/1558-7864(200101)17:1/10.1002/1558-7864(200101)17:1), although the latter program has not yet been adapted to search for STSs. As SSAHA pre-hashes the target sequence, it is expected that
a SSAHA-based STS localizer would be far faster than me-PCR. However, SSAHAs very large memory consumption and lack of portability make it untenable for researchers with typical computing resources.

CONCLUSIONS
The speed and versatility improvements provided by me-PCR allow the possibility of accomplishing large, well-defined genome annotation projects within research environments of modest computational means. As large-scale sequencing technology, capacity and utility increase, rapid and streamlined initial annotation of sequence tracts will be crucial for research and diagnostic purposes, me-PCR can be utilized as an effective component in a sequence analysis pipeline. Furthermore, the rapidity and high specificity of the implementation suggests potential as an initial means for locating any genomic sequence-defined feature, especially in finished sequencing projects.

ACKNOWLEDGEMENTS
The authors gratefully acknowledge G. Schuler for the original e-PCR algorithm and helpful comments. This work was supported in part by NIH grant MH60240.

REFERENCES