2010

csPCR: A computational tool for the simulation of the Polymerase Chain Reaction

Sandhya Dasu

April Williams

Yuriy Fofanov
University of Texas Medical Branch, yufofano@utmb.edu

Catherine Putonti
Loyola University Chicago, cputonti@luc.edu

OJB™

Online Journal of Bioinformatics ©

Volume 11 (1): 30-33, 2010.

csPCR: A computational tool for the simulation of the Polymerase Chain Reaction

Sandhya Dasu[1], April Williams[2,3], Yuriy Fofanov[1,4] and Catherine Putonti[2,3,4,5*]

[1]Department(s) of Computer Science and [4]Biology, University of Houston, [2]Department(s) of Biology, [3]Bioinformatics and [5]Computer Science, Loyola University Chicago, USA *Corresponding Author: cputonti@luc.edu

ABSTRACT

Dasu S, Williams A, Fofanov Y, Putonti C., csPCR: A computational tool for the simulation of the Polymerase Chain Reaction, Online J Bioinformatics, 11 (1): 30-33, 2010. Herein we present a computational simulation package PCR (csPCR) which models the PCR reaction, taking into consideration the issues of specificity, sensitivity, potential mishybridizations throughout the primer sequence as well as at the 3' end, and primer interactions, including self-complementarity and primer-primer interactions. A single target sequence or multiple target sequences can be considered simultaneously in addition to multiple primer sequences; thus a complex community and/or a multiplex assay can be simulated in a manner analogous with the actual experiment. This tool leaves primer design to the user, as there is a wealth of existing programs already available, and rather focuses on simulation of the anticipated amplification and expected agarose gel in addition to providing information about the location(s) of amplification in both text and graphical format. The software is freely available at www.bioinfo.uh.edu/csPCR.

INTRODUCTION

The success of Polymerase Chain Reaction (PCR) assays depends largely on the primers used in the experiment. While the primers are designed to hybridize to specific subsequences present within a target sequence, tolerance of base mismatches often occurs and may result in unexpected amplified products. Simulations of PCR experiments can greatly help to predict the

product(s) produced during the reaction in addition to identify unexpected amplicons due to mishybridizations. Applications have been developed to identify where user designed primers will amplify a particular target DNA/RNA sequence, employing a variety of methods for ascertaining the primer's specificity to the target sequence including hash-based searches (ePCR [1]; me-PCR [2]), BLAST-based searches (VPCR [3]; PUNS [4]; Primer-BLAST [5]), information theory-based (SPCR [6]) and the Findpatterns program from the GCG package (simPCR [7]). Moreover, with the exception of the PUNS system, all of the forementioned applications also take into consideration the likelihood of base mispairing between the primer and target sequence by either explicitly inputting the number (or range) of bp mismatches tolerated or an E-value threshold. Screening the primer(s) for potential interactions is left to the user. More sophisticated applications, such as FastPCR [8], include various options of search parameters depending upon the particular type of PCR assay the user wishes to simulate. Typically, the results of the applications are presented as text documents with just two providing the user with a visualization of the products simulating an agarose gel – VPCR [3] and SPCR [6]. Herein we present a computational simulation package PCR (csPCR) which models the PCR reaction, taking into consideration the issues of specificity, sensitivity, potential mishybridizations throughout the primer sequence as well as at the 3' end, and primer interactions, including self-complementarity and primer-primer interactions. A single target sequence or multiple target sequences can be considered simultaneously in addition to multiple primer sequences; thus a complex community and/or a multiplex assay can be simulated in a manner analogous with the actual experiment. This tool leaves primer design to the user, as there is a wealth of existing programs already available, and rather focuses on simulation of the anticipated amplification and expected agarose gel in addition to providing information about the location(s) of amplification in both text and graphical format. The software is freely available at www.bioinfo.uh.edu/csPCR.

METHOD

The csPCR has application was developed using the C#.NET framework. The number and size of PCR products can be calculated for a given DNA sequence(s), primer(s) and specificity of hybridization. Since many processes are involved during a PCR experiment, the application takes into consideration possible interactions between primers such as formation of secondary structure and cross-hybridization. The agarose gel is simulated using one of two standard ladders (100bp ladder including thirteen bands from 100bp to 3000bp and a 1Kb ladder including eleven bands from 500bp to 10,000bp) based on amplicon size. The use of a hash-based algorithm for finding hybridization sites significantly reduces the runtime. The program also has a cleanly separated back and front end so that the back end could be exported to a cluster for faster execution if so desired.

The time complexity of the algorithm is $O(mn+ x2)$, where n is the number of primers, m is the length of the template DNA, and x is the number of hybridization sites. The simulator stores and reports to the user both the list of the hybridization sites and the amplicon details. Thus, the memory complexity of the software is $O(x + y)$, where x is the number of hybridization sites and y is the number of amplicons computed. The number of hybridization sites and amplicons

produced, which directly corresponds to the time and memory required to perform the simulation, is a factor of the size and number of primer sequence(s) relative to the size of the target sequence as well as the number of base mishybridizations allowed [9]. The time and memory complexity have been tested using a Pentium IV processor machine by running the software using randomly generated DNA sequences of different sizes as well as real genomic sequences. The execution time is linearly proportional to the target genome size from less than a second (viral and some bacterial genomes) to minutes for the 3Gbp complete human genome. The RAM required is minimal (<<1MB) for even the complete human genome.

Figure 1A shows the interface of the application. For a given target sequence(s) and primer(s), each of the hybridization sites and the expected amplicon lengths are listed. The interface offers the user the ability to evaluate the primer(s) included with respect to the likelihood of primer-dimer formation and possible self-complementarity (hair-pin formation). The user is presented with the opportunity to remove particular primers either manually or automatically. In the event that the amplified region between two pairs of primers overlaps, it is difficult to predict what size or if any amplicon will be produced; thus, all of the putative amplicons are reported.

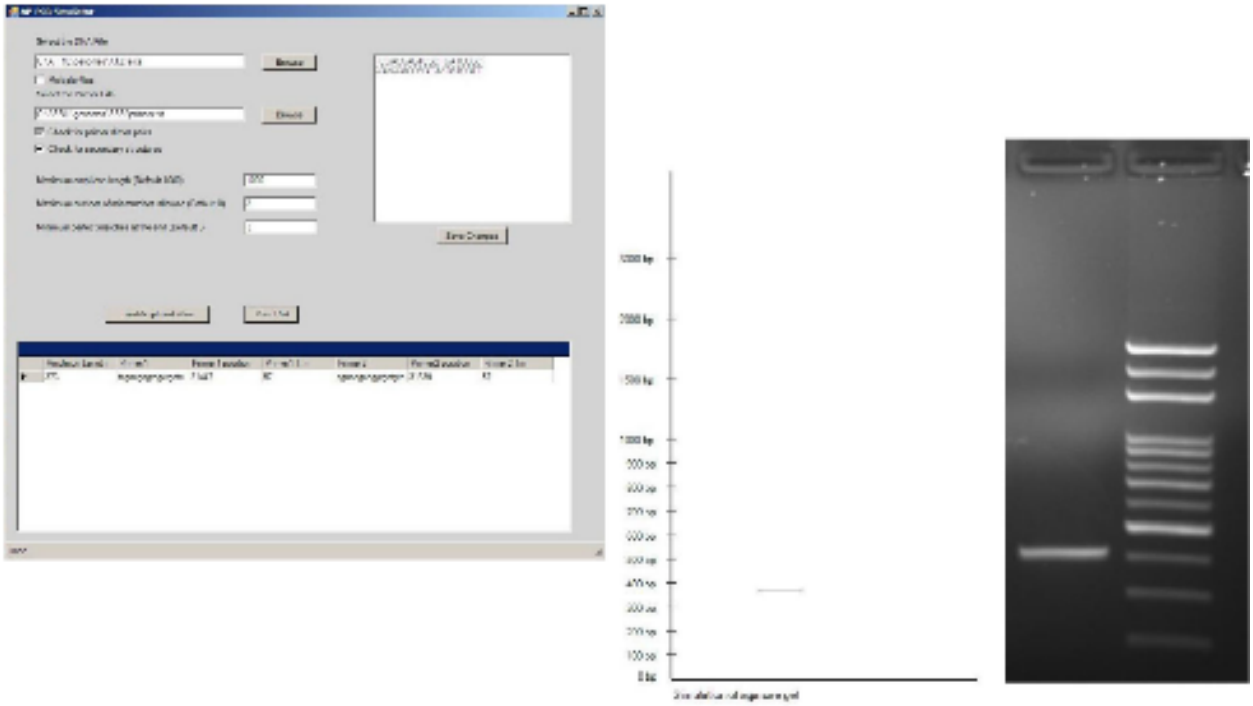

Figure 1. (a) Application interface. (b) Predicted amplification band (center) compared to an agarose gel (right) for the particular primer and target.

A set of primers had been designed to specifically detect the carbamoylphosphate synthetase large subunit (carB) of Pseudomonads (5’ ¨3’ forward primer: CGACCGCCGGTGATGTCGTT and 5’ ¨3’ reverse primer: TCGACGAGTCGCTGATCGGC). To test the specificity of the primer pair, they were used to amplify *E. coli* K12 MG1655 which resulted in the unexpected amplified

product ~400bp (the same size as expected for the Pseudomonad targets). BLAST results for the primer pair against the *E. coli* genome (NC_000913) listed only partial complementarity of the primer sequence to the genomic sequence, insinuating that several base misprimings were tolerated in the experimental reaction. The reaction was simulated using the csPCR application allowing for mismatches between the primer sequences and *E. coli* genome; the predicted gel and actual agarose gel are shown in Figure 1B.

The software developed simulates a PCR-based assay providing a visualization of not only the amplified regions but also the expected agarose gel result, taking into consideration possible mishybridizations. Moreover, preliminary screening of primer interactions is provided. The PCR simulator can be used for a wide variety of applications in addition to providing a useful PCR troubleshooting tool.

Acknowledgements

This work was supported by the Department of Homeland Security Science and Technology Directorate [NBCHC070063, NBCHC070054 to Y.F.]. The authors would like to thank Maia Larios-Sanz for her insightful comments and suggestions.

REFERENCES

1. Schuler GD. (1997) Sequence mapping by electronic PCR. *Genome Research,* 7**,** 541-550.
2. Murphy K, Raj T, Winters RS, White PS. (2004) me-PCR: a refined ultra fast algorithm for identifying sequence-defined genomic elements. *Bioinformatics*, 20, 588-590.
3. Lexa M, Horak J, Brzobohaty B. (2001) Virtual PCR. *Bioinformatics*, **17**, 192-193.
4. Boutros PC, Okey AB. (2004) PUNS: transcriptomic- and genomic-in silico PCR for enhanced primer design. *Bioinformatics,* 20, 2399-2400.
5. Primer Blast, NCBI, http://www.ncbi.nlm.nih.gov/tools/primer-blast/
6. Cao Y, Wang L, Xu K, Kou C, Zhang Y, Wei G, He J, Wang Y, Zhao L. (2005) Information theory-based algorithm for *in silico* prediction of PCR products with whole genomic sequences as templates. *BMC Bioinformatics*, 6, 190.
7. Rubin E, Levy AA. (1996) A mathematical model and a computerized simulation of PCR using complex templates. *Nucleic Acids Research*, 24**,** 3538–3545.
8. Kalendar R, Lee D, Schulman AH. (2009) FastPCR Software for PCR primer and probe design and repeat search. *Genes, Genomes and Genomics*, 3: 1-14.
9. Reed C, Fofanov V, Putonti C, Chumakov S, Slezak T, Fofanov Y. (2007) Effect of the mutation rate and background size on the quality of pathogen identification. *Bioinformatics,* 23, 2665-2671.