https://dx.doi.org/10.4314/dujopas.v10i3a.25
ISSN (Print): 2476-8316
ISSN (Online): 2635-3490

PyPCRtool: A Python Package for In Silico PCR and Primer Verification

Ibrahim Zubairu Waziri [1,3,4] , Mustapha Ibrahim Usman[2], Zainab Ali Dandalma[1]

[1]Department of Microbiology and Biotechnology,
Federal University Dutse,
Dutse, Jigawa State,
P.M.B 7156,
Nigeria.

[2]Department of Biological Sciences,
Nigeria Police Academy Wudil,
Kano State,
P.M.B. 3474
Nigeria.

[3]Department of Biochemistry,
Ahmadu Bello University, Zaria,
Kaduna State,
P.M.B 1045,
Nigeria.

[4]Africa Centre of Excellence for Neglected Tropical Diseases and Forensic Biotechnology,
Ahmadu Bello University, Zaria,
Kaduna State,
P.M.B 1045,
Nigeria.

Email: ibrahim.z@fud.edu.ng,

Abstract

In Silico PCR is a computational technique used to predict PCR outcomes, improve primer specificity, and optimize experimental conditions prior to conducting laboratory work. Numerous web-based tools with pre-loaded genome templates have been developed for conducting In Silico PCR simulations. However, there is an increasing demand for a flexible, user-friendly package that allows users to upload or define their own custom template sequences and operates offline, ensuring data privacy and security during In Silico PCR simulations and primer verification. This paper introduces PyPCRtool, a Python package designed to perform In Silico PCR simulations and verify primer specificity. The tool aims to offer a flexible, user-friendly solution that handles data locally, facilitating the prediction of DNA fragment amplification and the visualization of PCR product bands through gel electrophoresis simulations. PyPCRtool allows users to input and specify template DNA sequence files, forward and reverse primer sequences and customize mismatch tolerances. An example scenario demonstrates PyPCRtool's functionality, showcasing its ability

Author for Correspondence

to predict PCR product formation and visualize gel electrophoresis results. The tool provides detailed outputs, including PCR product sequences, sizes, and binding site information, assisting in experimental planning and analysis. PyPCRtool offers a robust and versatile solution for In Silico PCR and primer verification. By integrating flexible Python-based operations with local data handling for privacy, it serves as an invaluable resource for students and researchers in molecular biology and biotechnology, enhancing the accuracy and efficiency of PCR experimental planning and result interpretation.

Availability and implementation: PyPCRtool is implemented in python and is compatible with python v3.6 and higher. Source code: (https://github.com/waziiri/pypcrtool). Installation and Usage: from the source, or via PyPi (https://pypi.org/project/pypcrtool/).

Keywords: In Silico PCR, primer specificity, gel electrophoresis, DNA amplification and bioinformatics.

INTRODUCTION

Polymerase Chain Reaction (PCR) has established itself as a crucial method in both molecular biology and biotechnology. This technique employs oligonucleotide primers along with DNA polymerase to amplify DNA sequences that can span several thousand bases (Kalendar, 2022). The effectiveness of PCR is heavily dependent on the careful selection of suitable primers, as these determine the specificity and efficiency of the amplification process (Limberis & Metcalfe, 2023). Over the years, PCR has become an indispensable tool for various applications, including gene cloning, genetic analysis, and diagnostic testing (Van Weezep *et al.,* 2019).

In Silico PCR is a computational technique widely used in molecular biology to predict the outcome of PCR experiments without the need for actual laboratory work (Kalendar *et al.,* 2017). This method allows researchers to assess primer specificity, optimize PCR conditions, and anticipate the size and abundance of PCR products. By simulating PCR reactions on a computer, In Silico PCR saves time and resources, providing valuable insights before performing actual laboratory experiments (Kalendar, 2022). Gel electrophoresis is a common method used to visualize and analyze DNA fragments obtained from PCR amplification. This technique separates DNA fragments based on size, allowing researchers to confirm the presence and size of PCR products or amplicons (Kroczak *et al.,* 2022).

Several web-based tools with pre-defined genomes as templates have been developed for conducting In Silico PCR simulations. Lexa *et al.,* (2001) proposed virtual PCR as a computational tool for simulating PCR experiments. Bikandi *et al.,* (2004) performed an In Silico analysis of complete bacterial genomes using PCR, AFLP-PCR, and endonuclease restriction. Cock *et al.,* (2009) introduced Biopython, a bioinformatics library that includes PCR simulation capabilities. Yu & Zhang, (2011) presented methods for In Silico PCR analysis, while Kumar & Chordia, (2015) discussed primer designing and validation techniques. More recently, Van Weezep *et al.,* (2019) validated PCR diagnostics through In Silico analysis using freely available online software programs. Kalendar, (2022) provided a comprehensive guide to using FASTPCR software for In Silico PCR and oligonucleotide analysis. Additionally, Raney *et al.,* (2024) updated the UCSC Genome Browser database, which includes In Silico PCR functionality.

Despite the availability of these tools and methods, there is a demand for a flexible, user-friendly package that operates offline to ensure data privacy and security during In Silico PCR simulations and primer verification. Many existing tools are either limited in their flexibility or require internet access, posing potential risks to sensitive data.

PyPCRtool stands out from other In Silico PCR tools by offering Python-based flexibility, seamless integration with bioinformatics workflows, and local operation ensuring data privacy. Unlike web-based tools, PyPCRtool can be used offline, eliminating concerns about data security. Furthermore, PyPCRtool provides comprehensive outputs, including product sequence, size, binding site information, and visualization of PCR product bands through gel electrophoresis simulations. These unique features make PyPCRtool a robust and versatile solution for researchers in molecular biology and biotechnology.

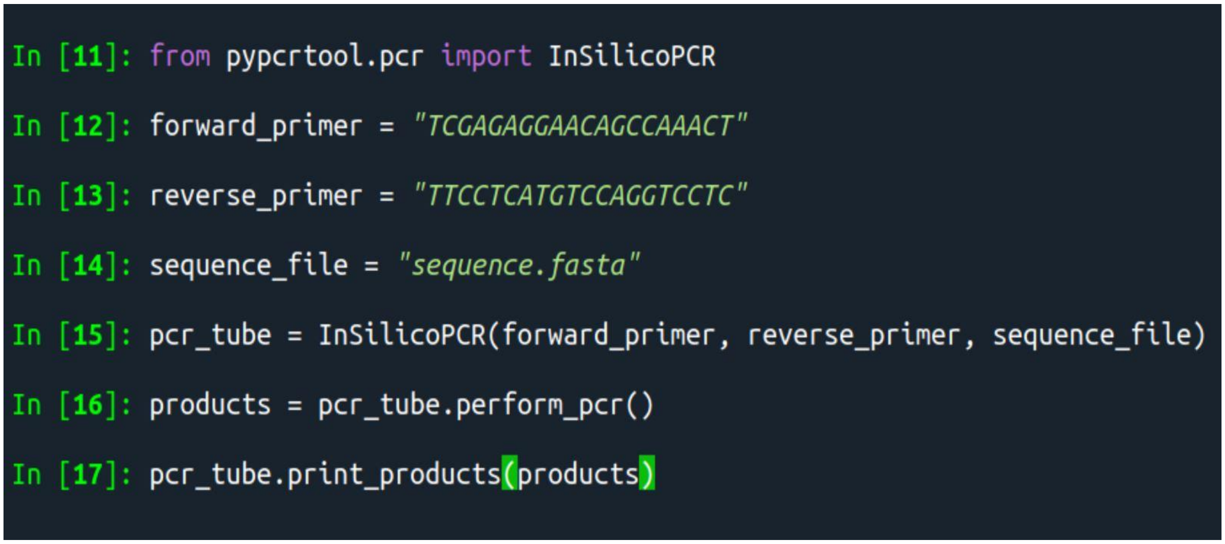
The current work aims to address the gaps identified in existing tools by presenting PyPCRtool, a Python program designed to perform In Silico PCR simulations and visualize the results through gel electrophoresis. The program provides functionalities to specify and load template DNA sequences from files, input forward and reverse primer sequences, set mismatch tolerances, simulate PCR amplification, check primer specificity, and visualize PCR product bands on a simulated gel. By offering these features, PyPCRtool enhances the accuracy and efficiency of PCR experimental planning and analysis, ensuring that students and researchers can effectively design and verify their PCR experiments.

METHOD AND IMPLEMENTATIONS

PyPCRtool is implemented in Python and structured as a comprehensive package with class-based functionalities. Key methods include:

- load_sequence(): Loads DNA sequences from files in FASTA format.
- find_approximate_matches(): Identifies primer binding sites with allowed mismatches.
- perform_pcr(): Simulates PCR amplification and identifies product sequences.
- print_products(): Prints the PCR product sequences to the console.
- visualize_gel(): Generates simulated gel electrophoresis images of PCR products.
- save_products(): Saves the PCR product sequences to a specified file.
- check_primer_specificity(): Evaluates primer specificity by counting binding site occurrences.

To demonstrate the capabilities of PyPCRtool for In Silico PCR simulations, a detailed procedure was conducted encompassing several key steps (figure 1). Initially, the InSilicoPCR() class was imported from the pypcrtool package. Next, the forward_primer and reverse_primer sequences were defined alongside the template DNA sequence_file in FASTA format. The primers and template sequence are available for download at https://github.com/waziiri/pypcrtool. Using the InSilicoPCR() class, an instance of the PCR simulation object was created, involving the primer sequences and the template DNA sequence file as arguments. Subsequently, the perform_pcr() method was invoked to simulate the PCR amplification process, predicting the resulting PCR product sequences and sizes. To display these PCR products, the print_products() method was used, printing the predicted product sequences to the console. Additionally, the visualize_gel() method was employed to generate and display a simulated gel electrophoresis image. The comprehensive usage guide for PyPCRtool is available at https://github.com/waziiri/pypcrtool.



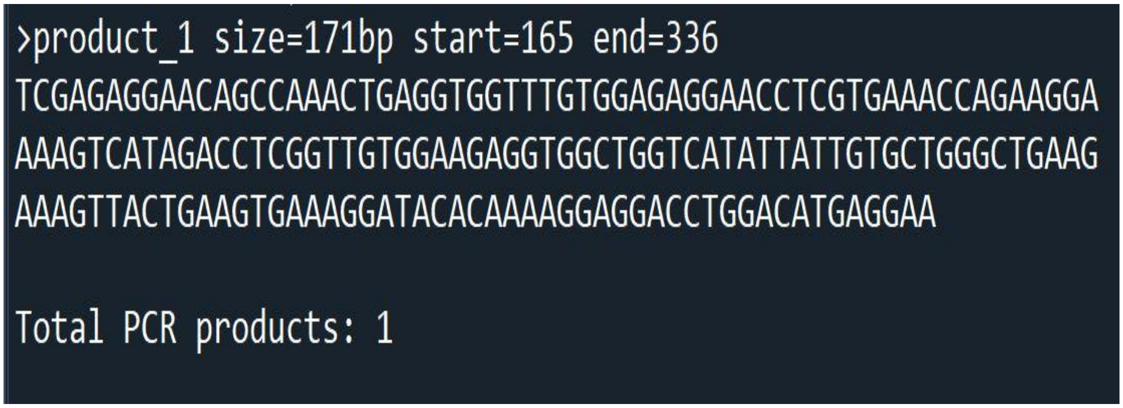
```
In [11]: from pypcrtool.pcr import InSilicoPCR

In [12]: forward_primer = "TCGAGAGGAACAGCCAAACT"

In [13]: reverse_primer = "TTCCTCATGTCCAGGTCCTC"

In [14]: sequence_file = "sequence.fasta"

In [15]: pcr_tube = InSilicoPCR(forward_primer, reverse_primer, sequence_file)

In [16]: products = pcr_tube.perform_pcr()

In [17]: pcr_tube.print_products(products)
```

Figure 1. Pypcrtool codes for simulating PCR amplification and printing PCR products

RESULTS AND DISCUSSION

PyPCRtool was rigorously tested using a set of forward and reverse primer sequences alongside a template DNA sequence file as shown in figure 1. The tool successfully predicted PCR product formation and provided detailed outputs, including product sequences, sizes, and binding positions. For example, the output in Figure 2 shows the sequences of the amplified DNA fragments, their sizes, and the primer binding positions, with the forward primer starting at position 165 and the reverse primer ending at position 336, resulting in a PCR product of 171 bp. This precision ensures that the primers bind specifically to the target DNA regions, which is crucial for reliable PCR results. The capability PyPCRtool to simulate gel electrophoresis results was also validated, offering clear visualization of PCR product bands on a simulated gel. Figure 3 demonstrates this by displaying a 171 bp band on the gel, helping researchers quickly confirm the presence and size of the amplified DNA fragments. This feature allows for an intuitive and straightforward assessment of PCR results, aiding in the validation of experimental setups.

PyPCRtool stands out for its simplicity, user-friendliness, and flexibility compared to existing tools. Unlike the comprehensive and complex Biopython (Cock *et al.,* 2009; Hu *et al.,* 2022), which can be challenging due to its extensive functionalities, PyPCRtool is specifically designed for In Silico PCR, making it more accessible for users focused on PCR simulation and primer verification. PyPCRtool also offers significantly more flexibility than online In Silico PCR tools (Carrasco-Acosta & Garcia-Jimenez, 2024; San Millán *et al.,* 2013), which come with pre-loaded genome sequences and limit users to uploading a maximum of four user-defined templates, PyPCRtool allows users to define their own genome or DNA sequence template input files with no limit on the number of templates. This feature greatly enhances the options for experimental design. Additionally, unlike the UCSC Genome Browser (Raney *et al.,* 2024), which only supports pre-loaded sequences for In Silico PCR, PyPCRtool enables researchers to input custom DNA sequences. This flexibility is crucial for studies involving unique or less commonly studied genomes, allowing for comprehensive analysis tailored to specific research needs. A key advantage of PyPCRtool is its ability to operate locally, ensuring data privacy. Many In Silico PCR tools are web-based, requiring users to upload their template and primers sequences, which can lead to data privacy concerns. In contrast, PyPCRtool runs entirely on the user's local machine, safeguarding sensitive genetic information and ensuring data security. This local operation is a significant advantage for researchers working with confidential or proprietary genetic data, providing peace of mind and compliance with data protection regulations. The example scenario showcased in this paper highlights PyPCRtool's capabilities, illustrating its potential to assist researchers in optimizing PCR experimental conditions and improving result reliability. By offering detailed outputs and visualization features, PyPCRtool aids in experimental planning and analysis, ensuring that researchers can effectively design and verify their PCR experiments. These unique features make PyPCRtool an invaluable resource for students and researchers, providing comprehensive outputs and visualization capabilities that enhance experimental planning and analysis, thereby optimizing PCR conditions and improving reliability.



```
>product_1 size=171bp start=165 end=336
TCGAGAGGAACAGCCAAACTGAGGTGGTTTGTGGAGAGGAACCTCGTGAAACCAGAAGGA
AAAGTCATAGACCTCGGTTGTGGAAGAGGTGGCTGGTCATATTATTGTGCTGGGCTGAAG
AAAGTTACTGAAGTGAAAGGATACACAAAAGGAGGACCTGGACATGAGGAA

Total PCR products: 1
```

Figure 2. Output of the PCR products simulation produced by PyPCRtool

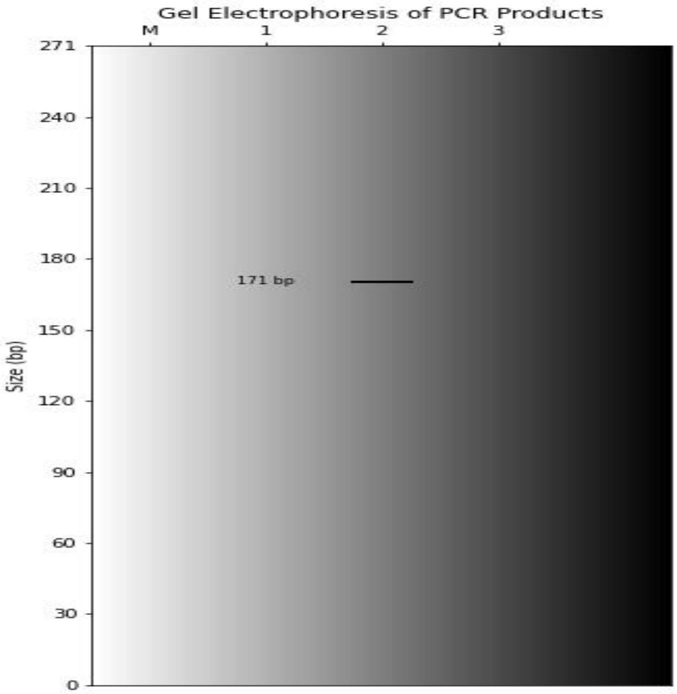


Figure 3. Output of gel electrophoresis simulation produced by PyPCRtool

CONCLUSION

PyPCRtool demonstrates significant utility in performing In Silico PCR simulations and primer verification. Through an example scenario, we showed how inputting forward and reverse primer sequences along with a DNA sequence file allows PyPCRtool to accurately predict PCR product formation and visualize gel electrophoresis results. The program outputs comprehensive information, including PCR product sequences, sizes, and binding positions, aiding in experimental planning and analysis. Additionally, PyPCRtool enables thorough primer specificity assessment, essential for designing effective PCR experiments. By offering functionalities for primer specificity analysis and gel electrophoresis visualization, PyPCRtool supports researchers in optimizing PCR experiments and interpreting results, thereby serving as an indispensable resource in computational biology research.

REFERENCES

Bikandi, J., Millán, R. S., Rementeria, A., & Garaizar, J. (2004). In silico analysis of complete bacterial genomes: PCR, AFLP–PCR and endonuclease restriction. *Bioinformatics*, *20*(5), 798–799. https://doi.org/10.1093/bioinformatics/btg491

Carrasco-Acosta, M., & Garcia-Jimenez, P. (2024). Development of Multiplex RT qPCR Assays for Simultaneous Detection and Quantification of Faecal Indicator Bacteria in Bathing Recreational Waters. *Microorganisms*, *12*(6), 1223. https://doi.org/10.3390/microorganisms12061223

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., & de Hoon, M. J. L. (2009). Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423. https://doi.org/10.1093/bioinformatics/btp163

Hu, Y., Xu, F., Huang, B., Chen, X., & Lin, S. (2022). A Python script to design primers for overlap extension PCR to ligate two DNA fragments. *PeerJ*, *10*, e14283. https://doi.org/10.7717/peerj.14283

Kalendar, R. (2022). A Guide to Using FASTPCR Software for PCR, In Silico PCR, and Oligonucleotide Analysis. *Methods in Molecular Biology (Clifton, N.J.)*, *2392*, 223–243. https://doi.org/10.1007/978-1-0716-1799-1_16

Kalendar, R., Muterko, A., Shamekova, M., & Zhambakin, K. (2017). In Silico PCR Tools for a Fast Primer, Probe, and Advanced Searching. *Methods in Molecular Biology (Clifton, N.J.)*, *1620*, 1–31. https://doi.org/10.1007/978-1-4939-7060-5_1

Kroczak, A., Wierzbicki, H., & Urantówka, A. D. (2022). In Silico Analysis of Seven PCR Markers Developed from the CHD1, NIPBL and SPIN Genes Followed by Laboratory Testing Shows How to Reliably Determine the Sex of Musophagiformes Species. *Genes*, *13*(5), 932. https://doi.org/10.3390/genes13050932

Kumar, A., & Chordia, N. (2015). In silico PCR primer designing and validation. *Methods in Molecular Biology (Clifton, N.J.)*, *1275*, 143–151. https://doi.org/10.1007/978-1-4939-2365-6_10

Lexa, M., Horak, J., & Brzobohaty, B. (2001). Virtual PCR. *Bioinformatics*, *17*(2), 192–193. https://doi.org/10.1093/bioinformatics/17.2.192

Limberis, J. D., & Metcalfe, J. Z. (2023). primerJinn: A tool for rationally designing multiplex PCR primer sets for amplicon sequencing and performing in silico PCR. *BMC Bioinformatics*, *24*, 468. https://doi.org/10.1186/s12859-023-05609-1

Raney, B. J., Barber, G. P., Benet-Pagès, A., Casper, J., Clawson, H., Cline, M. S., Diekhans, M., Fischer, C., Navarro Gonzalez, J., Hickey, G., Hinrichs, A. S., Kuhn, R. M., Lee, B. T., Lee, C. M., Le Mercier, P., Miga, K. H., Nassar, L. R., Nejad, P., Paten, B., … Haeussler, M. (2024). The UCSC Genome Browser database: 2024 update. *Nucleic Acids Research*, *52*(D1), D1082–D1088. https://doi.org/10.1093/nar/gkad987

San Millán, R. M., Martínez-Ballesteros, I., Rementeria, A., Garaizar, J., & Bikandi, J. (2013). Online exercise for the design and simulation of PCR and PCR-RFLP experiments. *BMC Research Notes*, *6*, 513. https://doi.org/10.1186/1756-0500-6-513

Van Weezep, E., Kooi, E. A., & van Rijn, P. A. (2019). PCR diagnostics: In silico validation by an automated tool using freely available software programs. *Journal of Virological Methods*, *270*, 106–112. https://doi.org/10.1016/j.jviromet.2019.05.002

Yu, B., & Zhang, C. (2011). In silico PCR analysis. *Methods in Molecular Biology (Clifton, N.J.)*, *760*, 91–107. https://doi.org/10.1007/978-1-61779-176-5_6