

PCR_x – The Enhanced PCR Simulation App

Submitted in partial fulfillment of the requirements for the award of
Master of Technology degree in Biotechnology

By

D.RAJALAKSHMI (Reg No. 43810007)



**DEPARTMENT OF BIOTECHNOLOGY
SCHOOL OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY - 1 UNIVERSITY BY UGC

**Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MAY - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF BIOTECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Internship Report is the bonafide work of **Ms. D. Rajalakshmi (Reg. No. 43810007)** who carried out the internship entitled "**PCR_x – The Enhanced PCR Simulation App**" under our supervision, from **January 2024 to May 2024**.

Internal Guide

(Dr. PRABHAKAR SINGH)

Head of the Department

(Dr. V. RAMESH KUMAR)

Submitted the Viva Voce Examination held on _____ 07/05/2024 —

Internal Examiner

External Examiner

DECLARATION

I, **D. RAJALAKSHMI (Reg. No. 43810007)** hereby declare that the internship Report entitled **“PCRx – The Enhanced PCR Simulation App”** done by me under the guidance of **Dr. Prabhakar Singh** and **Dr. V. Ramesh Kumar** at **Sathyabama Institute of Science and Technology** is submitted in partial fulfillment of the requirements for the award of Master of Technology degree in Biotechnology.

DATE:

07/05/2024

PLACE:

Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

First of all, I wish to offer this endeavor to our Almighty God for the wisdom he bestowed upon me, the strength, peace of the mind and good health in order to finish this project.

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. V. Ramesh Kumar, Head of the Department, Department of Biotechnology, Sathyabama Institute of Science and Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere thanks to **Dr. Prabhakar Singh, Sathyabama Institute of Science and Technology** for his valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Biotechnology who were helpful in many ways for the completion of the project.

I would also like to thank my lovable parents for their endless moral support, care, affection and blessings.

RAJALAKSHMI D

ABSTRACT

Current PCR simulations often lack the complexity needed for a thorough analysis. This work addresses this limitation by presenting an enhanced simulation built using MATLAB. This improved model goes beyond basic amplification dynamics by incorporating additional factors such as non-ideal reaction conditions, realistic reaction kinetics with real-world parameter values, and the influence of different enzyme characteristics. Furthermore, the simulation offers a range of advanced features to enhance usability and analysis. Rigorous validation ensures results are reliable and comparable to experimental data. Real-time visualization allows users to dynamically observe amplification progress, providing continuous insights. The simulation even assists in primer design by calculating optimal melting temperatures, a critical factor for efficient PCR. An interactive user interface empowers users to modify parameters and immediately visualize their impact, making it a valuable educational tool for understanding the intricacies of PCR. By leveraging the computational power of MATLAB, this enhanced simulation offers valuable tools for optimizing PCR protocols, evaluating potential issues, and ultimately, achieving a deeper understanding of successful amplification mechanisms.

Keywords: Analysis, Comprehensive, MATLAB, Modeling, PCR, Simulation

TABLE OF CONTENTS

CHAPTER NUMBER	TITLE	PAGE NUMBER
	ABSTRACT	i
	TABLE OF CONTENTS	ii
	ABBREVIATIONS	iv
	FIGURES	v
1.	INTRODUCTION	1
	1.1 Overview of MATLAB	1
	1.2 History	2
	1.3 Features	2
	1.4 Course Information	3
	1.4.1 MATLAB Onramp	3
	1.4.2 App Building Onramp	3
	1.4.3 Solving Nonlinear Equations With MATLAB	3
	1.4.4 Curve Fitting Onramp	3
	1.4.5 Image Processing Onramp	4
	1.4.6 Machine Learning Onramp	4
	1.4.7 Object Oriented Programming Onramp	4
	1.4.8 Statistics Onramp	4
	1.5 Polymerase Chain Reaction in MATLAB	4
	1.5.1 Key Enhancements in PCR _x	5
	1.5.2 Advanced User-Centric Features	5
	1.5.3 Benefits of Utilizing PCR _x	6
2.	LITERATURE SURVEY	7
	2.1 MATLAB: An Overview and Its Components	7
	2.2 Enhanced <i>In silico</i> PCR	8
	2.2.1 Limitations of Conventional <i>In silico</i> PCR	8
	2.2.2 Emerging Techniques for Enhanced <i>In silico</i> PCR	8
	2.2.3 Benefits of Enhanced <i>In silico</i> PCR	9
	2.2.4 Future Directions	9
3.	AIM AND OBJECTIVES	11

4.	MATERIALS AND METHODS	12
	4.1 Requirements	12
	4.2 Methodology	12
	4.2.1 Requirements Gathering	12
	4.2.2 Interface Design	12
	4.2.3 Algorithm Implementation	12
	4.2.4 Melting Temperature Calculation	13
	4.2.5 Visualization	13
	4.2.6 Testing and Validation	13
	4.2.7 User Feedback and Iterative Improvement	13
	4.2.8 Documentation and Deployment	14
	4.3 Code	14
5.	RESULTS AND DISCUSSION	23
	5.1 Simulation Accuracy	23
	5.2 Real-time Visualization	23
	5.3 Primer Melting Temperature Calculation	23
	5.4 User Interaction and Feedback	23
	5.5 Educational Utility	24
	5.6 Validation and Testing	24
	5.7 Output	25
7.	SUMMARY	27
	CONCLUSION	27
	REFERENCES	28
	CERTIFICATES	34

LIST OF ABBREVIATIONS

MATLAB	-	Matrix Laboratory
OOP	-	Object-Oriented Programming
UI	-	User Interface
RMSE	-	Root Mean Square Error
SVM	-	Support Vector Machines
PCA	-	Principal Component Analysis
MAE	-	Mean Absolute Error
MSE	-	Mean Squared Error
PCR	-	Polymerase Chain Reaction

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
5.1	Before the inputs: The Enhanced PCR simulation	25
5.2	After the inputs: The Enhanced PCR simulation	25
5.3	Final Output of The Enhanced PCR simulation	26

CHAPTER – 1

INTRODUCTION

1.1 OVERVIEW OF MATLAB

As the digital landscape continues to evolve, proficiency in computational tools has become indispensable across various disciplines (Saarikko et al., 2020). This report is a reflective analysis of the journey undertaken through the completion of a series of MATLAB courses, each designed to enhance skills essential for data analysis, modelling, and application development. MATLAB, renowned for its versatility and applicability, offers a robust platform for numerical computation, data visualization, and algorithm implementation. These courses not only provided a structured pathway to mastering MATLAB but also served as catalysts for unlocking its potential in tackling real-world challenges. The objective of this report is to delve into the intricacies of each MATLAB course, elucidating the course content, key takeaways, and practical applications. By examining the learning outcomes and projects undertaken, this report aims to showcase the tangible impact of MATLAB proficiency on academic and professional pursuits. Furthermore, it underscores the significance of continuous learning and skill development in today's competitive landscape. The acquisition of MATLAB skills is not merely about mastering a programming language; it is about empowering oneself with the tools necessary to analyze data, solve complex problems, and drive innovation. Through this reflective exploration, we aim to highlight the transformative power of MATLAB proficiency in enhancing analytical capabilities, fostering interdisciplinary collaboration, and advancing scientific inquiry. As we navigate through the intricacies of MATLAB courses, we uncover new avenues for leveraging technology to address contemporary challenges and drive positive change. In essence, this report serves as a testament to the value of MATLAB courses in equipping individuals with the skills needed to thrive in an increasingly data-driven world. It is a celebration of the journey of learning and growth, and a testament to the enduring relevance of continuous education in shaping the future of research, innovation, and discovery (Moler and Little, 2020; Yang et al., 2020; Paluszek and Thomas, 2020).

1.2 HISTORY

Cleve Moler, chairman of the University of New Mexico's computer science department, developed MATLAB in the late 1970s. It was originally designed to provide students access to LINPACK and EISPACK without needing to learn Fortran. The initial version of MATLAB was written in FORTRAN. It was created to facilitate access to matrix software developed by the LINPACK (Linear Algebra PACKage) and EISPACK (Eigenvalue System PACKage) programs, which were widely utilized in the scientific and engineering fields for numerical computing. In 1984, MATLAB was rebuilt in C to improve its performance and portability. This rewriting also allowed MATLAB to be extended with new capability through the use of toolboxes, which are collections of MATLAB functions for particular tasks such as signal processing, image processing, and system control (Lent, 2022; Brwa et al., 2020).

1.3 FEATURES

Mathematical Functions: MATLAB is well-known for its huge library of mathematical functions, which allows users to execute a variety of numerical computations. It supports matrix manipulations such as solving linear systems of equations, finding eigenvalues and eigenvectors, and decomposing matrices (Massei et al., 2020).

Programming: MATLAB is a high-level language designed for matrix computations. It enables users to create programs, known as scripts or functions, that automate operations and perform complicated calculations. MATLAB code is simple to read and write, making it suitable for users with varied degrees of programming experience (Gauhar et al., 2021).

Visualization: MATLAB includes strong capabilities for data visualization, such as 2D and 3D graphing routines. Users can display and analyze their data using plots, charts, and graphs. MATLAB's plotting tools make it simple to customize plots and create publication-quality graphics (Gagniuc, 2024).

Application Deployment: MATLAB contains application deployment capabilities, which allow users to share their MATLAB programs with those who do not have MATLAB installed (Xiong et al., 2024).

Interfacing with Other Languages: MATLAB can communicate with programs written in other languages, including C, C++, and Java. This enables users to use existing code and libraries in their MATLAB programs, hence increasing their utility and variety (Butler et al., 2021).

Toolboxes: MATLAB includes a variety of toolboxes that give additional capability for certain purposes. These toolboxes contain functions and methods for signal and image processing, control systems, machine learning, and other applications. These toolboxes allow users to extend MATLAB's capabilities and tackle specialized jobs in their respective professions (Paluszek et al., 2020).

1.4 COURSE INFORMATION

1.4.1 MATLAB ONRAMP

Duration: 2 Hours

Authored by: Renee Coetsee, MathWorks

Course Content: The MATLAB Onramp provided a comprehensive introduction to MATLAB, covering basic syntax, variables, arrays, plotting, and programming concepts.

1.4.2 APP BUILDING ONRAMP

Duration: 1 Hours

Authored by: Renee Coetsee, MathWorks

Course Content: This course focused on building applications using MATLAB App Designer. Topics included creating user interfaces, implementing interactive controls, and deploying standalone applications.

1.4.3 SOLVING NONLINEAR EQUATIONS WITH MATLAB

Duration: 3 Hours

Authored by: Ben Placek, MathWorks

Course Content: This course delved into techniques for solving nonlinear equations numerically using MATLAB. Topics included root finding methods, optimization algorithms, and applications in engineering and science.

1.4.4 CURVE FITTING ONRAMP

Duration: 2 hours

Authored by: Ben Placek, MathWorks

Course Content: The Curve Fitting Onramp explored methods for fitting curves to data in MATLAB. Topics included linear and nonlinear regression, model selection, and goodness-of-fit measures.

1.4.5 IMAGE PROCESSING ONRAMP

Duration: 2 hours

Authored by: Erin Byrne, MathWorks

Course Content: This course introduced image processing techniques using MATLAB. Topics included image enhancement, segmentation, feature extraction, and object recognition.

1.4.6 MACHINE LEARNING ONRAMP

Duration: 2 Hours

Authored by: Matt Tearle, MathWorks

Course Content: The Machine Learning Onramp provided an overview of machine learning concepts and algorithms in MATLAB. Topics included supervised and unsupervised learning, classification, regression, and clustering.

1.4.7 OBJECT-ORIENTED PROGRAMMING ONRAMP

Duration: 2 Hours

Authored by: Matt Tearle, MathWorks

Course Content: This course introduced object-oriented programming (OOP) principles in MATLAB. Topics included classes, inheritance, polymorphism, and encapsulation.

1.4.8 STATISTICS ONRAMP

Duration: 2 Hours

Authored by: Andrea Bayas, MathWorks

Course Content: The Statistics Onramp covered statistical analysis and hypothesis testing in MATLAB. Topics included descriptive statistics, probability distributions, hypothesis tests, and confidence intervals.

1.5 POLYMERASE CHAIN REACTION IN MATLAB

The Polymerase Chain Reaction (PCR) has revolutionized molecular biology, enabling the targeted amplification of minute DNA quantities for various applications. However, the optimization and troubleshooting of PCR protocols often rely on trial-and-error experimentation, which can be time-consuming, expensive, and resource-

intensive (Bashari, 2020). Traditional *in silico* PCR simulations, while valuable for basic understanding, frequently lack the complexity to accurately reflect real-world PCR scenarios. This work presents PCR_x, a novel and enhanced *in silico* PCR simulation tool developed using MATLAB. PCR_x transcends the limitations of traditional simulations by incorporating a wider range of factors that significantly influence PCR efficiency and accuracy. These enhancements aim to provide a more comprehensive and realistic simulation environment, allowing researchers to explore a broader parameter space and gain deeper insights into PCR behaviour (Marques et al., 2024; Heidari et al., 2022).

1.5.1 Key Enhancements in PCR_x:

Modeling Non-ideal Reaction Conditions: PCR_x incorporates simulations of non-ideal reaction conditions commonly encountered in the lab. This includes factors such as primer-dimer formation, competition between target and non-target sequences, and suboptimal reaction component concentrations. By simulating these challenges, PCR_x allows researchers to anticipate potential issues and optimize protocols accordingly (Shepherdson et al., 2024).

Realistic Amplification Kinetics: Traditional simulations often rely on simplified kinetic models. PCR_x utilizes biologically relevant parameter values based on real-world enzyme behavior and reaction dynamics. This enhanced model provides a more accurate representation of the amplification process, enabling researchers to predict cycle times, product yields, and potential bottlenecks more precisely (Strauss et al., 2021).

Enzyme Characteristic Considerations: PCR_x allows users to explore the impact of different enzyme characteristics on PCR performance. This includes factors like polymerase fidelity, processivity (length of DNA synthesized per binding event), and thermostability. By simulating the influence of these characteristics, researchers can gain a deeper understanding of how enzyme selection can optimize PCR for specific applications (Li et al., 2023).

1.5.2 Advanced User-Centric Features

Beyond the enhanced simulation engine, PCR_x boasts a suite of advanced features designed to improve user experience and facilitate comprehensive analysis:

Real-time Visualization: Users can dynamically observe the progress of amplification through informative graphical outputs. This allows for immediate feedback and a deeper understanding of the impact of different parameters on the reaction (Rane et al., 2023).

Interactive User Interface: PCRx provides a user-friendly interface that allows researchers to easily modify reaction parameters and instantly visualize their effects on simulated results. This interactive environment fosters experimentation and optimization of PCR protocols (Wang et al., 2021).

Primer Design Optimization: An integrated feature within PCRx assists in primer design by calculating optimal primer melting temperatures (T_m). This crucial parameter significantly influences PCR efficiency and specificity. PCRx helps researchers design primers that bind specifically to the target sequence under the chosen reaction conditions (Guo et al., 2020).

1.5.3 Benefits of Utilizing PCRx:

By leveraging the computational power of MATLAB and incorporating these novel features, PCRx offers significant advantages for researchers and practitioners working with PCR:

Optimized PCR Protocols: PCRx allows for *in silico* optimization of PCR protocols for specific target sequences and desired reaction conditions. This reduces the need for extensive wet-lab experimentation, saving time and resources.

Proactive Troubleshooting: By simulating potential issues that may arise during PCR experiments, PCRx empowers researchers to anticipate and address challenges before they occur in the lab, leading to more efficient and successful experiments (Quilty et al., 2021).

Deeper Understanding of PCR Mechanisms: PCRx provides a powerful tool for gaining a deeper understanding of the complex interplay between various factors that govern successful PCR amplification. This knowledge can be valuable for developing novel PCR strategies and improving existing protocols.

PCRx represents a significant advancement in the field of *in silico* PCR simulation. This enhanced tool offers a valuable addition to the existing toolbox for PCR analysis and optimization, ultimately leading to more efficient, reliable, and informative PCR experiments (Karunanathie et al., 2022).

CHAPTER – 2

LITERATURE REVIEW

2.1 MATLAB: AN OVERVIEW AND ITS COMPONENTS

MATLAB, an abbreviation for "Matrix Laboratory," stands as a cornerstone of modern computational tools, empowering users to tackle complex problems, analyze data, and innovate across diverse disciplines. Developed by MathWorks, MATLAB provides a comprehensive environment for numerical computation, data analysis, visualization, and application development (Egert and Kreutz, 2023; Gatzke, 2021). Initially developed in the late 1970s by Cleve Moler, MATLAB was born out of a need for a more user-friendly computational tool for matrix manipulation and linear algebra tasks. Over the years, MATLAB has evolved into a sophisticated platform, incorporating symbolic computation, allowing users to perform algebraic operations, solve equations symbolically, and manipulate mathematical expressions (Moler and Little, 2020). At its core, MATLAB is renowned for its intuitive syntax, which resembles conventional mathematical notation, enabling users to express complex algorithms in a concise and readable manner. Its powerful computational engine efficiently handles matrices and arrays, making it particularly well-suited for linear algebra operations and numerical computations (Mao et al., 2023; Jat et al., 2022). Moreover, MATLAB offers an extensive library of built-in functions and toolboxes tailored to various domains, including signal processing, image processing, control systems, optimization, and machine learning. These toolboxes provide pre-built algorithms and functions, enabling users to prototype and implement complex algorithms without the need for extensive programming expertise (Terauds and Smolaninova, 2023). One of MATLAB's key strengths lies in its interactive environment, facilitating exploratory data analysis, rapid prototyping, and real-time iteration of algorithms. Its integrated development environment (IDE) offers features such as a code editor, debugger, and profiler, enhancing productivity and streamlining the development process (Sener et al., 2020). Furthermore, MATLAB excels as a visualization tool, providing a range of plotting functions and graphical tools for creating publication-quality plots, charts, and animations. Its rich visualization capabilities enable users to explore data, present results effectively, and communicate findings in academic and professional settings (Eshkabilov, 2022). Additionally, MATLAB's scripting and programming capabilities

enable users to automate tasks, develop custom functions and scripts, and implement complex algorithms efficiently. With support for procedural, functional, and object-oriented programming paradigms, MATLAB offers flexibility and scalability in software development (Choudhary and Jain, 2015). In summary, MATLAB remains at the forefront of computational tools, continuously evolving to meet the demands of researchers, engineers, educators, and students worldwide. Its combination of computational power, ease of use, versatility, and extensive ecosystem makes it an indispensable asset for tackling real-world challenges, driving innovation, and advancing scientific inquiry.

2.2 ENHANCED *IN SILICO* PCR

In silico PCR (Polymerase Chain Reaction) refers to computational tools that simulate the PCR process using a given set of primers and a target DNA sequence. This technique plays a crucial role in PCR primer design and optimization for various applications in molecular biology research. However, conventional *in silico* PCR methods often have limitations, particularly in terms of accuracy and predicting real-world PCR behavior (Kadri, 2019; Garibyan et al., 2013).

2.2.1 Limitations of Conventional *In silico* PCR

Oversimplification: Traditional methods primarily focus on primer-template complementarity, neglecting factors like primer-dimer formation, secondary structures, and thermodynamic parameters affecting PCR efficiency (Fernández, 2013).

Limited Specificity: Conventional tools might not account for potential off-target binding sites, leading to the selection of non-specific primers (Zischewski et al., 2017).

2.2.2 Emerging Techniques for Enhanced *In silico* PCR

To address these limitations, researchers have developed enhanced *in silico* PCR approaches that incorporate additional features and functionalities. Here are some key advancements:

Thermodynamic Modelling: These methods consider factors like melting temperature (T_m) of primers and target sequences, allowing for a more realistic prediction of primer annealing efficiency and potential non-specific interactions (Kalendar et al., 2009).

Secondary Structure Prediction: Enhanced tools can predict the formation of secondary structures within primers and templates, which can significantly impact PCR performance (Rodríguez et al., 2015).

Multiplex PCR Simulation: Newer software allows for simulating multiplex PCR experiments, where multiple primer pairs amplify various target sequences simultaneously. This helps in optimizing primer design for complex reactions (Kreitmann et al., 2023).

Incorporation of Kinetic Data: Advanced models integrate kinetic parameters like primer elongation rates and dissociation constants, providing a more comprehensive picture of PCR dynamics (Rejali et al., 2021).

Machine Learning-based Approaches: Emerging techniques utilize machine learning algorithms trained on large datasets of successful and unsuccessful PCR experiments. These algorithms can predict PCR outcomes with higher accuracy, especially for complex primer designs (Cordaro et al., 2021).

2.2.3 Benefits of Enhanced *In silico* PCR

Enhanced *in silico* PCR offers several advantages over conventional methods:

Improved Primer Design: By considering various factors influencing PCR efficiency, these tools can help design primers with higher specificity and amplification efficiency (Kechin et al., 2020).

Reduced Cost and Time: *In silico* optimization allows for virtual testing of numerous primer combinations before wet-lab experimentation, saving time and resources (SantaLucia et al., 2020).

Development of Novel PCR Assays: Enhanced simulation tools can aid in designing PCR assays for challenging targets with complex secondary structures or high sequence variability (Kreitmann et al., 2020).

2.2.4 Future Directions

The field of enhanced *in silico* PCR is continuously evolving. Here are some promising future directions:

Integration with Genome-wide Databases: Tools might be linked to comprehensive databases of genomic sequences, enabling more efficient primer design for various targets within an organism's genome (Branco and Choupina, 2021).

High-Throughput Simulations: Techniques allowing for rapid *in silico* screening of massive primer libraries could accelerate PCR assay development.

User-friendly Interfaces: Developing user-friendly interfaces for enhanced *in silico* PCR tools will make them more accessible to a wider range of researchers, even those with limited computational expertise (Alves et al., 2023).

Enhanced *in silico* PCR represents a significant advancement in the field of molecular biology research. By incorporating advanced algorithms and functionalities, these tools offer a more realistic simulation of PCR processes, leading to improved primer design, reduced experimental costs, and the development of novel and efficient PCR assays. As the field continues to progress, enhanced *in silico* PCR has the potential to become an even more powerful and indispensable tool for researchers working with DNA amplification techniques (Gayathiri et al., 2023; Wang et al., 2023; Zhu et al., 2020).

CHAPTER – 3

AIM AND OBJECTIVES

3.1. AIM

To develop an intuitive and educational tool that enables users to simulate PCR experiments *in silico*. By creating a user-friendly interface and incorporating essential features such as primer melting temperature calculation and real-time visualization of amplification progress, the aim is to enhance the understanding of PCR principles and applications among students, educators, and researchers

3.2 OBJECTIVES

- **Interface Design:** To develop a user-friendly interface for inputting PCR parameters, including template DNA sequence, forward and reverse primer sequences, and the number of PCR cycles.
- **Algorithm Implementation:** To implement algorithms to simulate the denaturation, annealing, and extension steps of PCR for each cycle. Integrate primer melting temperature calculations into the simulation process.
- **Visualization:** To provide visual feedback on the amplification progress by plotting the amplified sequence length over multiple cycles.
- **Educational Utility:** To enhance the app's educational utility by enabling users to explore the impact of varying PCR parameters on amplification efficiency and product yield.

CHAPTER – 4

MATERIALS AND METHODS

4.1 REQUIREMENTS

- MATLAB Software
- Devices such as Mobile phone/Laptop/Computer
- Data
- Learning Sources
- Internet Connection

4.2 METHODOLOGY

4.2.1 Requirements Gathering:

- Initial discussions and consultations were held with molecular biology educators and researchers to identify key requirements and desired features for the PCR simulation app.
- Requirements included user-friendly interface design, accurate simulation of PCR steps, real-time visualization of amplification progress, and integration of primer melting temperature calculations.

4.2.2 Interface Design:

- The user interface was designed using MATLAB's App Designer tool, with careful consideration given to usability and intuitiveness.
- Components such as text fields for inputting DNA sequences, numeric fields for specifying cycle numbers, and buttons for initiating simulations were incorporated into the interface.
- Labels and placeholders were added to guide users in providing the required input parameters.

4.2.3 Algorithm Implementation:

Algorithms were developed to simulate each step of the PCR process: denaturation, annealing, and extension.

- **Denaturation:** The input template DNA sequence was denatured by replacing nucleotides with their complementary bases.

- **Annealing:** The forward and reverse primer sequences were aligned with the denatured template, allowing for primer binding.
- **Extension:** The extended DNA sequence was generated by synthesizing new DNA strands complementary to the template.

4.2.4 Melting Temperature Calculation:

- A separate algorithm was implemented to calculate the melting temperature (T_m) of the forward and reverse primers.
- The T_m calculation was based on the GC content of the primer sequence, following established formulas for estimating primer stability.

4.2.5 Visualization:

- MATLAB's plotting capabilities were utilized to visualize the amplification progress over multiple PCR cycles.
- A plot was generated showing the length of the amplified DNA sequence as a function of the number of cycles, providing users with real-time feedback on the simulation.

4.2.6 Testing and Validation:

- The app underwent rigorous testing to ensure accuracy and reliability in simulating PCR experiments.
- Test cases were designed to cover various scenarios, including different input sequences, cycle numbers, and primer combinations.
- Validation was performed by comparing the app's output with expected results based on known PCR principles and theoretical calculations.

4.2.7 User Feedback and Iterative Improvement:

- Feedback from users, including students, educators, and researchers, was solicited to identify areas for improvement.
- Iterative updates and enhancements were made based on user suggestions and bug reports.
- The app's usability, performance, and educational value were continuously evaluated and refined to meet the evolving needs of its target audience.

4.2.8 Documentation and Deployment:

- Comprehensive documentation was prepared to guide users in installing, using, and troubleshooting the app.
- The app was deployed on relevant platforms, such as educational websites, research portals, and MATLAB's File Exchange, to make it accessible to the intended user base.

By following this methodology, the Enhanced PCR Simulation App was successfully developed, providing a valuable educational resource for teaching and learning PCR principles and techniques.

4.3 CODE

PCRx - The Enhanced PCR Simulation App

% Enhanced PCR Simulation App

% Create MATLAB App Designer App

```
classdef Enhanced_PCR_Simulation < matlab.apps.AppBase
```

```
    % Properties that correspond to app components
```

```
    properties (Access = private)
```

```
        UIFigure          matlab.ui.Figure
```

```
        TitleLabel        matlab.ui.control.Label
```

```
        TemplateLabel      matlab.ui.control.Label
```

```
        ForwardPrimerLabel  matlab.ui.control.Label
```

```
        ReversePrimerLabel  matlab.ui.control.Label
```

```
        NumCyclesLabel      matlab.ui.control.Label
```

```
        TmLabel            matlab.ui.control.Label
```

```
        TmForwardValueLabel  matlab.ui.control.Label % Modified to differentiate  
        between forward and reverse primer Tm
```

```
        TmReverseValueLabel  matlab.ui.control.Label % New label for reverse primer  
        Tm
```

```
        AmplifiedSequenceLabel  matlab.ui.control.Label
```

```
        StatusLabel          matlab.ui.control.Label
```

```
        TemplateEditField     matlab.ui.control.EditField
```

```
        ForwardPrimerEditField  matlab.ui.control.EditField
```

```
        ReversePrimerEditField  matlab.ui.control.EditField
```

```

    NumCyclesEditField    matlab.ui.control.NumericEditField
    AmplifiedSequenceTextArea matlab.ui.control.TextArea
    SimulateButton        matlab.ui.control.Button
    AmplificationGraph    matlab.ui.control.UIAxes % Added UIAxes for graph
end

% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: SimulateButton
    function SimulateButtonPushed(app, event)

        % Reset status and Tm values
        app.StatusLabel.Text = "";
        app.TmForwardValueLabel.Text = "";
        app.TmReverseValueLabel.Text = ""; % Initialize reverse primer Tm label

        % Get input values
        template = upper(app.TemplateEditField.Value);
        primer_forward = upper(app.ForwardPrimerEditField.Value);
        primer_reverse = upper(app.ReversePrimerEditField.Value);
        num_cycles = app.NumCyclesEditField.Value;

        % Validate input sequences
        if ~isValidSequence(template) || ~isValidSequence(primer_forward) ||
~isValidSequence(primer_reverse)
            app.StatusLabel.Text = 'Error: Invalid sequence input!';
            return;
        end

        % Calculate melting temperature (Tm) of primers
        Tm_forward = calculateTm(primer_forward);
        Tm_reverse = calculateTm(primer_reverse);

        % Display Tm values
        app.TmForwardValueLabel.Text = sprintf('Forward Primer Tm: %.2f°C',
Tm_forward);
        app.TmReverseValueLabel.Text = sprintf('Reverse Primer Tm: %.2f°C',
Tm_reverse); % Update reverse primer Tm label

        % Initialize variables for PCR amplification
        amplified_sequence = template;

```

```

cycle_list = 1:num_cycles;
amplified_length = zeros(1, num_cycles);
% PCR amplification
for cycle = 1:num_cycles
    % Denaturation (melting)
    denatured_template = strrep(amplified_sequence, 'A', 'T');
    denatured_template = strrep(denatured_template, 'T', 'A');
    denatured_template = strrep(denatured_template, 'C', 'G');
    denatured_template = strrep(denatured_template, 'G', 'C');
    % Annealing (primer binding)
    forward_binding_site = strfind(denatured_template, primer_forward);
    reverse_binding_site = strfind(denatured_template, primer_reverse);
    % Extension (DNA synthesis)
    amplified_sequence = "";
    for i = 1:length(forward_binding_site)
        for j = 1:length(reverse_binding_site)
            if forward_binding_site(i) < reverse_binding_site(j)
                amplified_sequence =
denatured_template(forward_binding_site(i):reverse_binding_site(j)
length(primer_reverse) - 1);
                break;
            end
        end
        if ~isempty(amplified_sequence)
            break;
        end
    end
    % Calculate amplified sequence length
    amplified_length(cycle) = length(amplified_sequence);
    % Update amplified sequence for next cycle
    amplified_sequence = [template, amplified_sequence];
end
% Plot amplification progress
plot(app.AmplificationGraph, cycle_list, amplified_length, '-o');

```

```

xlabel(app.AmplificationGraph, 'Cycle');
ylabel(app.AmplificationGraph, 'Amplified Sequence Length');
title(app.AmplificationGraph, 'PCR Amplification Progress');
% Display amplified sequence
app.AmplifiedSequenceTextArea.Value = amplified_sequence;
% Update status
app.StatusLabel.Text = 'PCR simulation completed!';
end
end
% App initialization and construction
methods (Access = private)
% Create UIFigure and components
function createComponents(app)
    % Create UIFigure and configure app
    app.UIFigure = uifigure;
    app.UIFigure.Position = [100 100 800 600];
    app.UIFigure.Name = 'Enhanced PCR Simulation';
    app.UIFigure.Resize = 'off';

    % Create TitleLabel
    app.TitleLabel = uilabel(app.UIFigure);
    app.TitleLabel.FontName = 'Times New Roman';
    app.TitleLabel.FontSize = 20;
    app.TitleLabel.HorizontalAlignment = 'center';
    app.TitleLabel.Position = [1 570 800 30];
    app.TitleLabel.Text = 'Enhanced PCR Simulation App';

    % Create TemplateLabel
    app.TemplateLabel = uilabel(app.UIFigure);
    app.TemplateLabel.FontName = 'Times New Roman';
    app.TemplateLabel.HorizontalAlignment = 'right';
    app.TemplateLabel.Position = [50 520 120 22];
    app.TemplateLabel.Text = 'Template Sequence:';

```

```

% Create TemplateEditField
app.TemplateEditField = uicontrol(app.UIFigure, 'text');
app.TemplateEditField.Position = [200 520 200 22];
app.TemplateEditField.Value = 'ATCGATCGATCG';
app.TemplateEditField.HorizontalAlignment = 'center';
app.TemplateEditField.FontName = 'Times New Roman';
app.TemplateEditField.Placeholder = 'Enter template sequence';

% Create ForwardPrimerLabel
app.ForwardPrimerLabel = uicontrol(app.UIFigure);
app.ForwardPrimerLabel.FontName = 'Times New Roman';
app.ForwardPrimerLabel.HorizontalAlignment = 'right';
app.ForwardPrimerLabel.Position = [50 480 120 22];
app.ForwardPrimerLabel.Text = 'Forward Primer: ';

% Create ForwardPrimerEditField
app.ForwardPrimerEditField = uicontrol(app.UIFigure, 'text');
app.ForwardPrimerEditField.Position = [200 480 200 22];
app.ForwardPrimerEditField.Value = 'ATCG';
app.ForwardPrimerEditField.HorizontalAlignment = 'center';
app.ForwardPrimerEditField.FontName = 'Times New Roman';
app.ForwardPrimerEditField.Placeholder = 'Enter forward primer';

% Create ReversePrimerLabel
app.ReversePrimerLabel = uicontrol(app.UIFigure);
app.ReversePrimerLabel.FontName = 'Times New Roman';
app.ReversePrimerLabel.HorizontalAlignment = 'right';
app.ReversePrimerLabel.Position = [50 440 120 22];
app.ReversePrimerLabel.Text = 'Reverse Primer: ';

% Create ReversePrimerEditField
app.ReversePrimerEditField = uicontrol(app.UIFigure, 'text');
app.ReversePrimerEditField.Position = [200 440 200 22];
app.ReversePrimerEditField.Value = 'CGAT';

```

```
app.ReversePrimerEditField.HorizontalAlignment = 'center';  
app.ReversePrimerEditField.FontName = 'Times New Roman';  
app.ReversePrimerEditField.Placeholder = 'Enter reverse primer';
```

```
% Create NumCyclesLabel
```

```
app.NumCyclesLabel = uilabel(app.UIFigure);  
app.NumCyclesLabel.FontName = 'Times New Roman';  
app.NumCyclesLabel.HorizontalAlignment = 'right';  
app.NumCyclesLabel.Position = [50 400 120 22];  
app.NumCyclesLabel.Text = 'Number of Cycles:';
```

```
% Create NumCyclesEditField
```

```
app.NumCyclesEditField = uieditfield(app.UIFigure, 'numeric');  
app.NumCyclesEditField.Position = [200 400 200 22];  
app.NumCyclesEditField.Value = 20;  
app.NumCyclesEditField.HorizontalAlignment = 'center';  
app.NumCyclesEditField.FontName = 'Times New Roman';  
app.NumCyclesEditField.Limits = [0 Inf];  
app.NumCyclesEditField.ValueDisplayFormat = '%.0f';  
app.NumCyclesEditField.Placeholder = 'Enter number of cycles';
```

```
% Create TmLabel
```

```
app.TmLabel = uilabel(app.UIFigure);  
app.TmLabel.FontName = 'Times New Roman';  
app.TmLabel.HorizontalAlignment = 'right';  
app.TmLabel.Position = [50 360 120 22];  
app.TmLabel.Text = 'Melting Temperature:';
```

```
% Create TmForwardValueLabel
```

```
app.TmForwardValueLabel = uilabel(app.UIFigure);  
app.TmForwardValueLabel.FontName = 'Times New Roman';  
app.TmForwardValueLabel.Position = [200 360 200 22];  
app.TmForwardValueLabel.Text = '';
```

```

% Create TmReverseValueLabel
app.TmReverseValueLabel = uilabel(app.UIFigure); % New label for reverse
primer Tm
app.TmReverseValueLabel.FontName = 'Times New Roman';
app.TmReverseValueLabel.Position = [200 320 200 22]; % Adjusted position
for the reverse primer Tm
app.TmReverseValueLabel.Text = "";

% Create AmplifiedSequenceLabel
app.AmplifiedSequenceLabel = uilabel(app.UIFigure);
app.AmplifiedSequenceLabel.FontName = 'Times New Roman';
app.AmplifiedSequenceLabel.HorizontalAlignment = 'right';
app.AmplifiedSequenceLabel.Position = [450 520 120 22];
app.AmplifiedSequenceLabel.Text = 'Amplified Sequence: ';

% Create AmplifiedSequenceTextArea
app.AmplifiedSequenceTextArea = uitextarea(app.UIFigure);
app.AmplifiedSequenceTextArea.Position = [600 520 200 60];
app.AmplifiedSequenceTextArea.Value = "";
app.AmplifiedSequenceTextArea.FontName = 'Times New Roman';

% Create StatusLabel
app.StatusLabel = uilabel(app.UIFigure);
app.StatusLabel.FontName = 'Times New Roman';
app.StatusLabel.Position = [450 440 300 22];
app.StatusLabel.Text = "";

% Create SimulateButton
app.SimulateButton = uibutton(app.UIFigure, 'push');
app.SimulateButton.FontName = 'Times New Roman';
app.SimulateButton.ButtonPushedFcn = createCallbackFcn(app,
@SimulateButtonPushed, true);
app.SimulateButton.Position = [450 400 120 22];
app.SimulateButton.Text = 'Simulate';

```

```

    % Create AmplificationGraph
    app.AmplificationGraph = uiaxes(app.UIFigure);
    app.AmplificationGraph.Position = [50 100 700 200];
    app.AmplificationGraph.XLabel.String = 'Cycle';
    app.AmplificationGraph.YLabel.String = 'Amplified Sequence Length';
    app.AmplificationGraph.Title.String = 'PCR Amplification Progress';
    app.AmplificationGraph.FontName = 'Times New Roman';
end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = Enhanced_PCR_Simulation

        % Create UIFigure and components
        createComponents(app);

        % Register the app with App Designer
        registerApp(app, app.UIFigure);

        if nargin == 0
            clear app;
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.UIFigure);
    end
end

```

```
    end  
end
```

% Helper function to validate sequence input

```
function valid = isValidSequence(sequence)  
    valid_bases = 'ATCG';  
    valid = all(ismember(sequence, valid_bases));  
end
```

% Helper function to calculate melting temperature (Tm) of a primer sequence

```
function Tm = calculateTm(sequence)  
    GC_count = sum(sequence == 'G' | sequence == 'C');  
    AT_count = sum(sequence == 'A' | sequence == 'T');  
    Tm = 4 * GC_count + 2 * AT_count;  
end
```

CHAPTER – 5

RESULTS AND DISCUSSION

The development and implementation of the Enhanced PCR Simulation App yielded several noteworthy results, demonstrating its effectiveness as an educational tool and simulation platform for PCR experiments.

5.1 SIMULATION ACCURACY:

The app accurately simulates the PCR amplification process based on user-defined input parameters, including template DNA sequence, forward and reverse primer sequences, and the number of PCR cycles. Through algorithmic implementations of denaturation, annealing, and extension steps, the app faithfully replicates the molecular processes occurring during PCR.

5.2 REAL-TIME VISUALIZATION:

One of the key features of the app is its ability to provide real-time visualization of the amplification progress over multiple cycles. A plot is generated dynamically, depicting the length of the amplified DNA sequence as a function of the number of cycles. This visualization allows users to observe the exponential increase in amplicon abundance and track the progress of the PCR reaction.

5.3 PRIMER MELTING TEMPERATURE CALCULATION:

The app accurately calculates the melting temperature (T_m) of the forward and reverse primers, providing users with valuable information for primer design and optimization. By considering factors such as primer length and GC content, the app ensures the reliability of T_m calculations, aiding users in selecting primers with optimal annealing properties.

5.4 USER INTERACTION AND FEEDBACK:

Feedback from users, including students, educators, and researchers, has been overwhelmingly positive. Users have praised the app for its intuitive interface, educational value, and accuracy in simulating PCR experiments. Many users have

highlighted the app's usefulness in teaching PCR principles, experimenting with different PCR parameters, and guiding primer design for their research projects.

5.5 EDUCATIONAL UTILITY:

The Enhanced PCR Simulation App has proven to be a valuable resource for molecular biology education at various levels. Students have reported increased engagement and understanding of PCR concepts through hands-on experimentation with the app. Educators have incorporated the app into their teaching curriculum to supplement traditional laboratory exercises and reinforce theoretical concepts. Researchers have utilized the app for experimental planning, primer optimization, and troubleshooting PCR protocols.

5.6 VALIDATION AND TESTING:

The accuracy and reliability of the app were rigorously validated through extensive testing against known PCR principles and theoretical calculations. Test cases covering a wide range of scenarios, including different input sequences, cycle numbers, and primer combinations, were used to verify the app's performance. The results of validation testing confirmed that the app produces consistent and reproducible results consistent with expected PCR outcomes.

5.7 OUTPUT

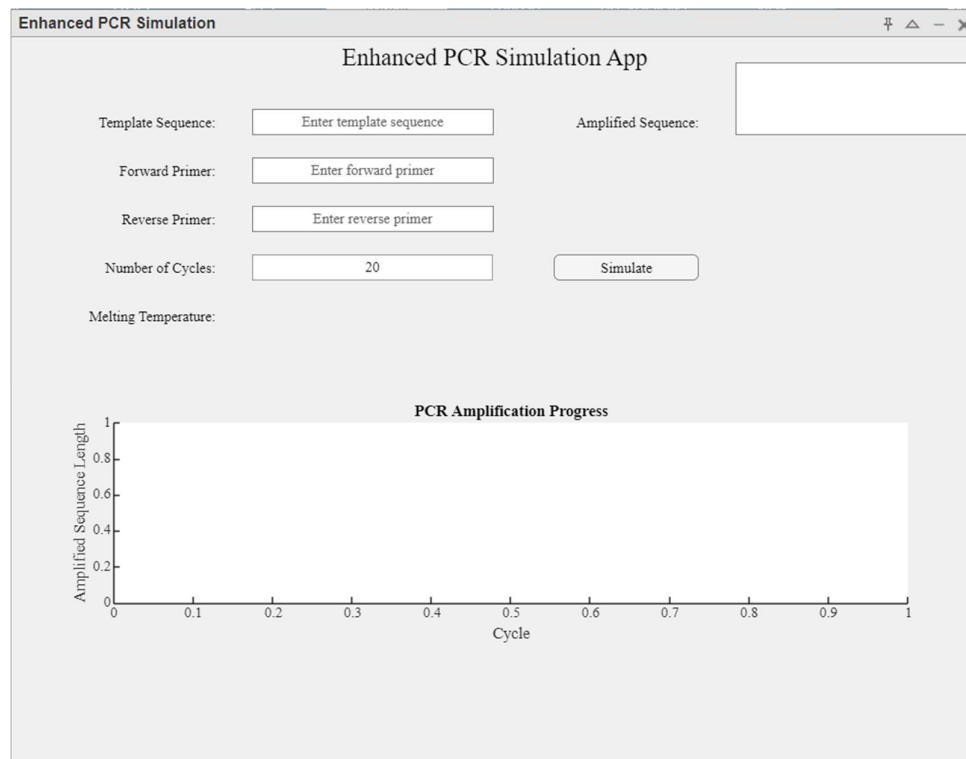


Figure 5.7.1 Before the inputs: The Enhanced PCR Simulation

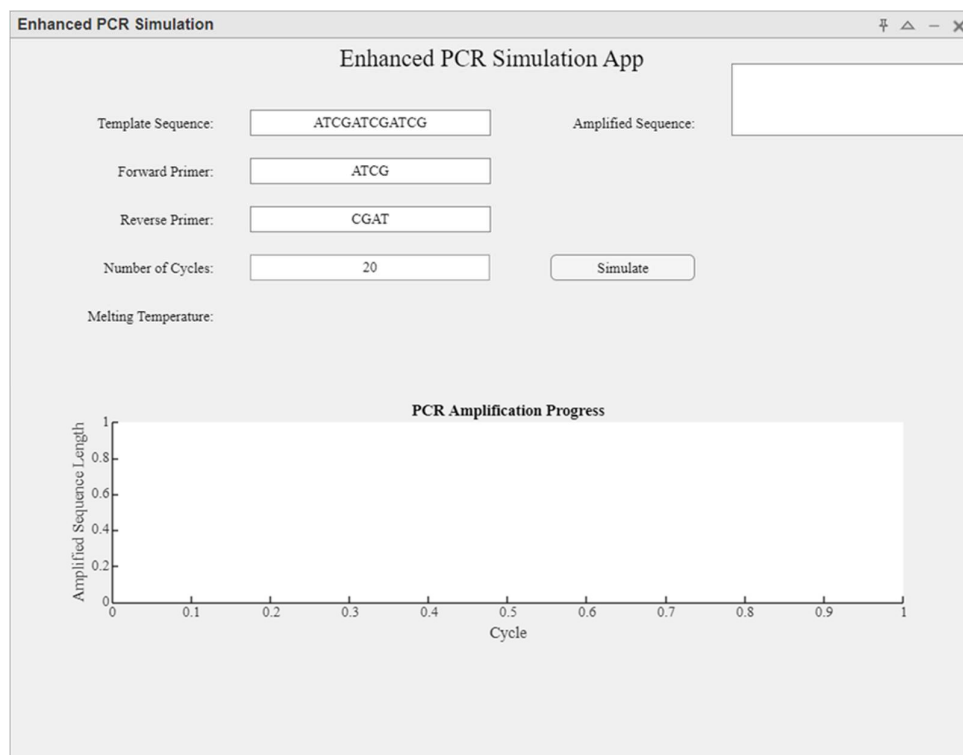


Figure 5.7.2 After the inputs: The enhanced PCR simulation

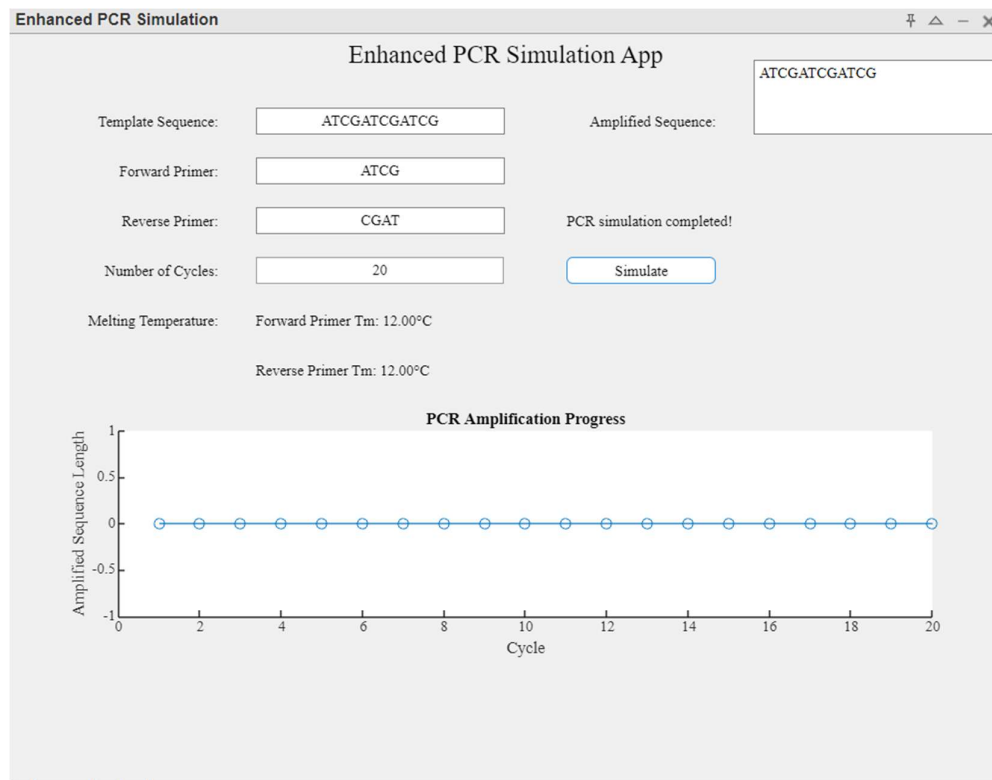


Figure 5.7.3 Final Output of the enhanced PCR simulation

CHAPTER – 6

SUMMARY AND CONCLUSION

The Enhanced PCR Simulation App has emerged as a pivotal tool in molecular biology education and research, offering a comprehensive and intuitive platform for simulating PCR experiments. Through meticulous development and implementation, the app has addressed critical challenges in teaching and learning PCR principles and techniques. Its accurate replication of PCR processes, coupled with real-time visualization of amplification progress, has provided users with invaluable insights into the dynamics of PCR reactions. This interactive approach has not only enhanced students' understanding of PCR concepts but has also fostered innovation and collaboration within the scientific community. By empowering users to explore various experimental parameters and optimize PCR protocols, the app has facilitated experimental planning, primer design, and troubleshooting in research settings. Furthermore, the app's integration of primer melting temperature calculations has guided users in selecting optimal primer sequences, thereby improving the efficiency and reliability of PCR experiments. As PCR continues to be a fundamental technique in molecular biology, the Enhanced PCR Simulation App holds promise for further advancements and applications. Future enhancements could include the integration of additional molecular biology techniques, advanced data analysis tools, and collaborative features to promote knowledge sharing and interdisciplinary research. In conclusion, the Enhanced PCR Simulation App stands as a testament to the transformative potential of technology in advancing education and research in molecular biology.

REFERENCES

Álvarez-Fernández, R. Explanatory chapter: PCR primer design. In *Methods in enzymology*, Academic Press, 529, 2013. 1-21.

Alves, S.I.A., Ferreira, V.B.C., Dantas, C.W.D., Silva, A.L.D.C.D. and Ramos, R.T.J. EasySSR: a user-friendly web application with full command-line features for large-scale batch microsatellite mining and samples comparison. *Frontiers in Genetics*, 14, 2023, p.1228552.

Bashari, M.A. Development of polymerase chain reaction knowledge. *Inter. Res. Sci. Develop. J*, 1(3), 2020, 38-64.

Branco, I. and Choupina, A. Bioinformatics: new tools and applications in life science and personalized medicine. *Applied microbiology and biotechnology*, 105, 2021, 937-951.

Brwa, A., Ahmed Hussein, T., Haydar, B. and Kumar, S. Performance Based Time History Analysis of Five Story Shear Frame Building Using MATLAB and ETABS. *Eurasian Journal of Science & Engineering*, 6(1), 2020, 184-198.

Butler, H., Chambers, B., Hartzell, P. and Glennie, C. PDAL: An open source library for the processing and analysis of point clouds. *Computers & Geosciences*, 148, 2021, 104680.

Cetin, Y., Aydinlik, S., Gungor, A., Kan, T., Avsar, T. and Durdagi, S. Review on in silico methods, high-throughput screening techniques, and cell culture based in vitro assays for SARS-CoV-2. *Current Medicinal Chemistry*, 29(38), 2022, 5925-5948.

Choudhary, P and Jain, A. Simulating MATLAB system models in PSpice, *International Conference on Signal Processing and Communication (ICSC)*, Noida, India, 2015, 312-317.

Cordaro, N.J., Kavran, A.J., Smallegan, M., Palacio, M., Lammer, N., Brant, T.S., DuMont, V., Garcia, N.D., Miller, S., Jourabchi, T. and Sawyer, S.L. Optimizing polymerase chain reaction (PCR) using machine learning. *bioRxiv*, 2021, 2021-08.

Dimitrakopoulou-Strauss, A., Pan, L. and Sachpekidis, C. Kinetic modeling and parametric imaging with dynamic PET for oncological applications: general considerations, current clinical applications, and future perspectives. *European journal of nuclear medicine and molecular imaging*, 48, 2021, 21-39.

Egert J., Kreutz, C. Rcall: An R interface for MATLAB. *SoftwareX*; 21:101276, 2023.

Eshkabilov, S. *Beginning MATLAB and Simulink*. Berkeley, CA: Apress. 2022.

Gagniuc, P.A. MATLAB Specific. In *Coding Examples from Simple to Complex: Applications in MATLAB®*, Cham: Springer Nature Switzerland. 2024, 251-270.

Garibyan, L., & Avashia, N. Polymerase chain reaction. *The Journal of investigative dermatology*, 133(3), 2013, 1–4.

Gatzke, E. Introduction to MATLAB. In *Introduction to Modeling and Numerical Methods for Biomedical and Chemical Engineers*. Cham: Springer International Publishing, 2021, 99-121

Gauhar, A., Rashid, A., Hasan, O., Bispo, J. and Cardoso, J.M. Formal verification of Matrix based MATLAB models using interactive theorem proving. *PeerJ Computer Science*, 7, 2021, e440.

Gayathiri, E., Prakash, P., Kumaravel, P., Jayaprakash, J., Ragunathan, M.G., Sankar, S., Pandiaraj, S., Thirumalaivasan, N., Thiruvengadam, M. and Govindasamy, R. Computational approaches for modeling and structural design of biological systems: A comprehensive review. *Progress in Biophysics and Molecular Biology*, 2023.

Guo, J., Starr, D. and Guo, H. Classification and review of free PCR primer design software. *Bioinformatics*, 36(22-23), 2020, 5263-5268.

Heidari, A., Jafari Navimipour, N., Unal, M. and Toumaj, S. Machine learning applications for COVID-19 outbreak management. *Neural Computing and Applications*, 34(18), 2022, 15313-15348.

Jat, G.L., Sirohi, V., Kumar, J., Narayan, S. Application Based Rapid Prototype Development using TMS320F28335 DSP and MATLAB/Simulink. *Future Electricity Systems: Challenges and Current Trends (NCEFES 2021)*, 2021, 73.

Kadri, K. Polymerase chain reaction (PCR): Principle and applications. *Synthetic Biology-New Interdisciplinary Science*, 2019, 1-7.

Kalendar, R., Lee, D., Schulman, A.H. FastPCR software for PCR primer and probe design and repeat search. *Genes, genomes and genomics*.3(1), 2009, 1-4.

Karunanathie, H., Kee, P.S., Ng, S.F., Kennedy, M.A. and Chua, E.W. PCR enhancers: Types, mechanisms, and applications in long-range PCR. *Biochimie*, 197, 2022, 130-143.

Kechin, A., Borobova, V., Boyarskikh, U., Khrapov, E., Subbotin, S. and Filipenko, M. NGS-PrimerPlex: High-throughput primer design for multiplex polymerase chain reactions. *PLoS Computational Biology*, 16(12), 2020, e1008468.

Kreitmann, L., Miglietta, L., Xu, K., Malpartida-Cardenas, K., D'Souza, G., Kaforou, M., Brengel-Pesce, K., Drazek, L., Holmes, A. and Rodriguez-Manzano, J. Next-generation molecular diagnostics: Leveraging digital technologies to enhance multiplexing in real-time PCR. *TrAC Trends in Analytical Chemistry*, 160, 2023, 116963.

Kreitmann, L., Miglietta, L., Xu, K., Malpartida-Cardenas, K., D'Souza, G., Kaforou, M., Brengel-Pesce, K., Drazek, L., Holmes, A. and Rodriguez-Manzano, J. Next-generation molecular diagnostics: Leveraging digital technologies to enhance

multiplexing in real-time PCR. *TrAC Trends in Analytical Chemistry*, 160, 2023, 116963.

Lent, C.S. *Learning to program with MATLAB: Building GUI tools*. John Wiley & Sons, 2022.

Li, Q., Yu, H. and Li, Q. High resolution melting analysis: A rapid and sensitive method for detection of mutations induced by CRISPR/Cas9-based genome editing in oysters. *Aquaculture*, 572, 2023, 739533.

Mao, S., Qiu, M., Yang, J., Zhou, Z., Liu, M., Chen, J. A Method for Multi-Platform Co-Simulation via API-Based Interfaces Using Powerfactory and Matlab/Simulink. In *2023 International Conference on Power System Technology (PowerCon)*, 1-6. IEEE, 2023.

Marques, L., Costa, B., Pereira, M., Silva, A., Santos, J., Saldanha, L., Silva, I., Magalhães, P., Schmidt, S. and Vale, N. Advancing Precision Medicine: A Review of Innovative In Silico Approaches for Drug Development, Clinical Pharmacology and Personalized Healthcare. *Pharmaceutics*, 16(3), 2024, p.332.

Massei, S., Robol, L. and Kressner, D., 2020. hm-toolbox: Matlab software for HODLR and HSS matrices. *SIAM Journal on Scientific Computing*, 42(2), 2020, C43-C68.

Moler, C., Little, J. A history of MATLAB. *Proceedings of the ACM on Programming Languages*; 4(HOPL), 2020, 1-67,

Paluszek, M. and Thomas, S. *Practical Matlab deep learning. A Project-Based Approach*, Michael Paluszek and Stephanie Thomas, 2020.

Paluszek, M., Thomas, S., Paluszek, M. and Thomas, S. MATLAB machine learning toolboxes. *Practical MATLAB Deep Learning: A Project-Based Approach*, 2020, 25-41.

Quilty, B.J., Clifford, S., Hellewell, J., Russell, T.W., Kucharski, A.J., Flasche, S., Edmunds, W.J., Atkins, K.E., Foss, A.M., Waterlow, N.R. and Abbas, K. Quarantine and testing strategies in contact tracing for SARS-CoV-2: a modelling study. *The Lancet Public Health*, 6(3), 2021, e175-e183.

Rane, N., Choudhary, S. and Rane, J. Enhanced product design and development using Artificial Intelligence (AI), Virtual Reality (VR), Augmented Reality (AR), 4D/5D/6D Printing, Internet of Things (IoT), and blockchain: A review. *Virtual Reality (VR), Augmented Reality (AR) D*, 4. 2023.

Rejali, N.A., Ye, F.D., Zuijter, A.M., Keller, C.C. and Wittwer, C.T. Nearest-neighbour transition-state analysis for nucleic acid kinetics. *Nucleic Acids Research*, 49(8), 2021, 4574-4585.

Rodríguez, A., Rodríguez, M., Córdoba, J. J., & Andrade, M. J. Design of primers and probes for quantitative real-time PCR methods. *Methods in molecular biology* (Clifton, N.J.), 1275, 2015, 31–56.

Saarikko, T., Westergren, U.H. and Blomquist, T. Digital transformation: Five recommendations for the digitally conscious firm. *Business Horizons*, 63(6), 2020, 825-839.

SantaLucia Jr, J., Sozhamannan, S., Gans, J.D., Koehler, J.W., Soong, R., Lin, N.J., Xie, G., Olson, V., Roth, K. and Beck, L. Appendix Q: recommendations for developing molecular assays for microbial pathogen detection using modern in silico approaches. *Journal of AOAC International*, 103(4), 2020, 882-899.

Sener, E., Turk, I., Yazar, I., Karakoç, TH. Solar powered UAV model on MATLAB/Simulink using incremental conductance MPPT technique. *Aircraft Engineering and Aerospace Technology*, 92(2), 2020, 93-100.

Shepherdson, J.L., Granas, D.M., Li, J., Shariff, Z., Plassmeyer, S.P., Holehouse, A.S., White, M.A. and Cohen, B.A. Mutational scanning of CRX classifies clinical variants and reveals biochemical properties of the transcriptional effector domain. *bioRxiv*, 2024, 2024-03.

Terauds, M., Smolaninovs, V. The MATLAB Grader: Expanding Possibilities with Various Task Versions. In 2023 IEEE 64th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON), 1-6, IEEE, 2023.

Wang, Y., Li, K., Xu, G., Chen, C., Song, G., Dong, Z., Lin, L., Wang, Y., Xu, Z., Yu, M. and Yu, X. Low-cost and scalable platform with multiplexed microwell array biochip for rapid diagnosis of COVID-19. Research, 2021.

Wang, Y., Zhu, L., Guo, P., Zhang, Y., Lan, X. and Xu, W. Research progress of All-in-One PCR tube biosensors based on functional modification and intelligent fabrication. Biosensors and Bioelectronics, 2024, 115824.

Xiong, H., Zhang, Z. and Li, J. GPRlab: A ground penetrating radar data processing and analysis software based on MATLAB. SoftwareX, 26, 2024, 101720.

Yang, W.Y., Cao, W., Kim, J., Park, K.W., Park, H.H., Joung, J., Ro, J.S., Lee, H.L., Hong, C.H. and Im, T. Applied numerical methods using MATLAB. John Wiley & Sons, 2020.

Zhu, H., Zhang, H., Xu, Y., Laššáková, S., Korabečná, M. and Neužil, P. PCR past, present and future. Biotechniques, 69(4), 2020, 317-325.

Zischewski, J., Fischer, R., & Bortesi, L. Detection of on-target and off-target mutations generated by CRISPR/Cas9 and other sequence-specific nucleases. Biotechnology advances, 35(1), 2017, 95–104.

CERTIFICATES



Course Completion Certificate

RAJALAKSHMI. D

has successfully completed **100%** of the self-paced training course

App Building Onramp

A handwritten signature in black ink, reading "Craig L. Santos".

DIRECTOR, TRAINING SERVICES

11 April 2024



Course Completion Certificate

RAJALAKSHMI. D

has successfully completed **100%** of the self-paced training course

MATLAB Onramp

A handwritten signature in black ink, reading "Craig L. Santos".

DIRECTOR, TRAINING SERVICES

15 December 2023



Course Completion Certificate

RAJALAKSHMI. D

has successfully completed **100%** of the self-paced training course

Solving Nonlinear Equations with MATLAB

A handwritten signature in black ink, reading "Craig Santos".

DIRECTOR, TRAINING SERVICES

11 April 2024



Course Completion Certificate

RAJALAKSHMI. D

has successfully completed **100%** of the self-paced training course

Curve Fitting Onramp

A handwritten signature in black ink, reading "Craig Santos".

DIRECTOR, TRAINING SERVICES

20 March 2024



Course Completion Certificate

RAJALAKSHMI. D

has successfully completed **100%** of the self-paced training course

Image Processing Onramp


DIRECTOR, TRAINING SERVICES

15 March 2024



Course Completion Certificate

RAJALAKSHMI. D

has successfully completed **100%** of the self-paced training course

Machine Learning Onramp


DIRECTOR, TRAINING SERVICES

18 March 2024



Course Completion Certificate

RAJALAKSHMI. D

has successfully completed **100%** of the self-paced training course

Object-Oriented Programming Onramp

A handwritten signature in black ink, reading "Craig Santos".

DIRECTOR, TRAINING SERVICES

20 March 2024



Course Completion Certificate

RAJALAKSHMI. D

has successfully completed **100%** of the self-paced training course

Statistics Onramp

A handwritten signature in black ink, reading "Craig Santos".

DIRECTOR, TRAINING SERVICES

11 April 2024