University of Glasgow | School of Computing Science

A Hands-on Approach to Learning Molecular Biology Techniques

Ross Eric Barnie
Dmitrijs Jonins
Daniel McElroy
Murray Ross
Ross Taylor

Level 3 Project — 18th March 2013

Abstract

PCR and Primer design is a concept taught to all students in the School of Life Sciences at Glasgow University. This application has been designed to allow the students to reinforce the concepts they have been taught and improve their skills in primer design. A user enters a DNA sequence into the application, before being asked to select a correct forward and reverse primer. The application gives feedback on these selections. By using this as a teaching tool, we hope the School of Life Sciences can enhance students understanding of these concepts.

Acknowledgements

First of all, we would like to thank our supervisor, Dr Gethin Norman, for his guidance, organisational skills and support throughout this project. We would also like to thank Dr Nicola Veitch and Dr Pamela Scott for their assistance with all matters relating to PCR and Primer design.

Education Use Consent

We hereby give our permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name:		Signature:	
Name:		Signature:	
Name:		Signature:	
Name:		Signature:	
Name:		Signature:	
Name:		Signature:	

Contents

Chapter 1

Introduction

1.1 Preliminaries

There are some terms that will be used later in the report that should be clarified here, so as to avoid confusion. The aim of the project is to creating a teaching tool for PCR, or Polymerase Chain Reactions. This is the process of amplifying a specified sequence of DNA thousands to millions of times. It should be explained that DNA sequences are made up of two strands, comprised of bases of the nucleotides Adenine, Thymine, Guanine and Cytosine, represented by the letters `a`, `t`, `g`, and `c` respectively. Base pairing is when one base bonds with its complement on the other strand. The bases `a` and `t` complement each other, and bases `g` and `c` complement each other.

Primers, used to select the sequence for PCR in a given selection of DNA, are shorter fragments of DNA, usually between 20 and 30 bases in length. For use in PCR, a primer must be chosen from the "left" of one strand (this is the forward primer) and the "right" of the other (this is the reverse primer), and these must obey a number of rules, which are the focus of the teaching tool:

- Neither primer should self-anneal. This means that if the primer were to fold over in such a way that that more than 3 bases in a row on one side complemented the bases on the opposite side, this primer would pair with itself and become useless for the purposes of PCR.
- The melting temperature of each primer, calculated in degrees Celsius using a simple mathematical formula involving the frequency of `a`s, `t`s, `g`s and `c`s, should be between 50 and 65°C, and within 2-3°C of each other.
- The forward primer should be unique within the first strand, and not appear in the second, complementary strand. Likewise, the reverse primer should be unique within the second strand and should not appear in the first.
- The percentage of `g`s and `c`s within each primer should be between 40% and 60%.
- The length of each primer should be between 20 and 30 bases.
- The same base should not be repeated several times in a row in either primer.
- The last base of each primer should be either a `g` or a `c`.

- The primers should not anneal to each other. This means that if the primers were put side by side, at no point of overlap should more than 3 bases in a row complement the overlapping primers base.

It should be noted that many of these rules are not precise, and do not involve rigid limits for success or failure. In fact, these "rules" more closely resemble rough guides. Obviously, the nature of programming does not lend itself to "rough guides", and so we followed advice from Pamela Scott and Nicola Veitch on how best to structure these tests to effectively represent the imprecision of their boundaries.

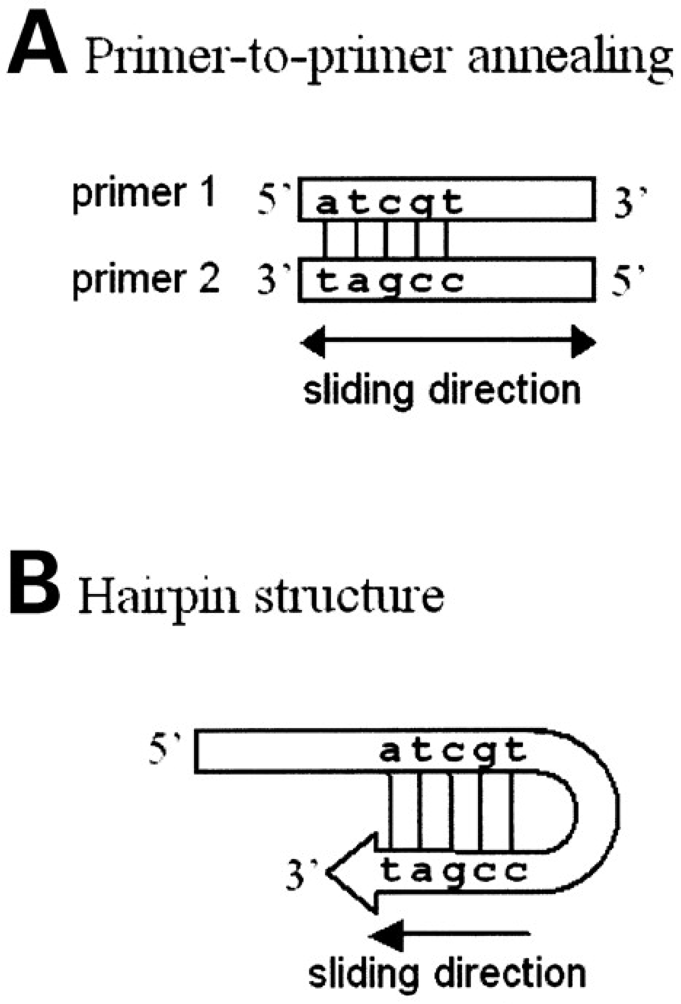


Figure 1.1: Annealing

The ends of strands of DNA are often referred to as the "5'-end" and the "3'end" which are named as such due to the positioning of carbon atoms near to the ends of the strand. This naming convention allows DNA strand orientation to be described as either "5'-3"' or "3'-5"'. This is relevant to PCR and primer design because strands anneal to each other by the 5'-ends of the strands joining to the 3'-ends of the other strand. This is shown in section A of Figure 1.1.

An enzyme is a biological which is used to synthesise new DNA strands. Taq Polymerase is the enzyme which synthesises a strand complementary to the target sequence.

Two external services that are integral to the successful use of the system are the National Center for Biotechnology Information (NCBI) [10], and the use of a Basic Local Alignment Search Tool. NCBI is, for the purposes of this exercise, a vast database of DNA sequences and related information, from which the user can obtain a sequence on which to perform PCR (though this can also be obtained from other sources). At the end of the exercise, the user is encouraged to use the BLAST functionality on the NCBI website to ensure that their primer is specific to their intended PCR target.

It should also be explained that we developed the teaching tool in Java, using Netbeans, a free IDE primarily designed to be used with Java, and developing the user interface with Swing, the primary Java 6 GUI framework and that we stored all project-related files in a version-controlled repository hosted by GitHub.

1.2 Aims

The overall aim of this project is to produce a piece of software to help Level 3 Life Sciences students taking the Molecular Methods course (taught by Drs. Pamela Scott and Nicola Veitch) learn about PCR and Primer Design Techniques and to allow them to test their knowledge of these subjects. At the outset of the project, Scott and Veitch helped us to separate this aim into key tasks to be completed and important aspects of the interface design to be implemented:

1. The software should work as an interactive tutorial which users can work through. This requires:

    - A number of areas for users to enter their own choice of data, such as a choice of DNA sequence to work with and the primers with which to operate on the selected strand. For this feature to be useful as an educational tool feedback must be provided upon data entry.
    - Users should be able to experiment with different data e.g. examining the different melting temperatures of different primers. This requires the ability to easily move forwards and backwards between the different stages of the tutorial.
    - To help newer users and students who are unfamiliar with PCR there should be simple instructions to guide users through the process and explain PCR throughout the application. The user should also be able to access the rules of PCR and primer design at any time.

2. The software should be accessible to all users with a basic understanding of molecular biology, regardless of their different levels of knowledge, ability etc.:

    - To achieve this, the interface should be uncomplicated and intuitive without compromising the required functionality. This will be aided by the instructions and help section mentioned above, as well as labels placed next to any areas users can interact with.
    - Any section which makes use of colour should be designed with colour blind users in mind.

3. The software should improve upon the tools currently available for learning primer design. The main issues with these systems are:

    - The low level of interactivity offered by the systems, such as the numerous YouTube videos available on the subject [21]. Users who are not actively working through a tutorial or a demonstration are likely to lose interest faster so it is important to make them involved with every step of the tutorial by having them design their own primers etc.
    - The available tools rarely go into detail about primer design specifically. One example of an interactive, well designed application that fails to convey the process of designing primers to a satisfactory degree is University of Utah's "PCR Virtual Lab" [22]. Therefore, an important aim for the project is that primer design must be explained in detail and provide enough information to be informative, whilst remaining interesting to students using the system.

4. Another aim related to accessibility is that the users should be able to download and use the software from home. This means that the program must be able to run on a variety of different operating systems and computers with varying performance levels. With this in mind it was decided that the program should be written in Java due to it being highly portable.

1.3 Background

PCR, in the simplest terms, is used to amplify (reproduce) a specific region of a DNA strand. Initially developed in 1983 by Dr. Kary Mullis [17], PCR has since found widespread use in a number of areas of genetic analysis such as detection of infectious disease organisms [23] and DNA profiling of suspects by forensic scientists as well as numerous research and medical applications. Given this wide range of applications, it is clear to see why PCR is an essential technique for students to learn and fully understand.

The idea for our system was first put forward by Dr Pamela Scott and Dr Nicola Veitch, the clients for the project. Their involvement stems from their role in teaching the Molecular Methods course, compulsory for all Level 3 Life Sciences students.

In their experience, they had found that a number of students struggle with the PCR element of the course, particularly the task of designing primers for the process (described in more detail in Section 1.1), and decided that a teaching tool for this element of the course would be helpful in engaging the students toward the subject.

They had seen a number of videos and an interactive web application on PCR, but these either did not focus on the actual design of primers or were severely limited by the lack of interactivity. It was with these missing elements in mind that they created the detailed problem specification that we were presented with at the beginning of the project.

1.4 Motivation

When selecting a project at the outset of the course, we identified several factors associated with this project that motivated us to take it on.

Chief among these factors was the aim of building an interactive teaching tool. Some members of the team expressed an interest in going on to create educational software after completion of their degree, and this project would serve as ideal experience in developing such software.

Another part of the project that excited us was the opportunity of working within the university to potentially improve the education of our peers. An important aspect of the involvement of Drs. Scott and Veitch was in gaining valuable experience in client relations. Several of the projects on offer, while interesting, did not involve any stakeholders other than their supervisors, and so we felt this project would be a unique opportunity to put into practice the lessons we had learned from other courses about requirements gathering, without the risk associated with the involvement of an external business entity, for whom the consequences of failure might be more severe.

Lastly, the element of the project that excited us the most was the chance to do work related to a field that we had absolutely minimal experience with. The sum total of biology-related experience on the team was Ross Taylor's Higher qualification in Biology in secondary school, and that placed us in an advantageous position to learn about certain elements of molecular biology from an outsider's perspective.

As previously explained in section 1.3, the idea for project came from a dissatisfaction from the teaching staff of Molecular Methods with the current method of teaching PCR, but in order to better understand what it is the lecturers sought after, research into PCR education systems currently in place became a necessity.

Current Systems

In order to understand the motivation for the development of the system, Doctors Scott and Veitch provided us with links to several systems currently in place which attempt to make learning this process more interactive and/or visual. However, videos and multimedia in general have been questioned as teaching aids in the past [19]. As expressed in this paper, simply because the information is in video or multimedia format does not necessarily mean that it is benefiting the learning of its viewers, or creating the correct environment to encourage learning. Interactivity, along with other factors, are key to engaging people to learn.

The first was a video hosted on YouTube [21], made by demonstrators within the School of Life Sciences. During its eighteen second duration, the video shows various elements of the PCR process including change in temperature and the role of the primer. However, it was agreed on by the team and by the clients that it was insubstantial in terms of information delivery, as several of the stages of PCR are omitted with no mention of primer design, and in terms of interactivity.

Another video hosted on YouTube [20], currently referred to on School of Life Sciences' website, is similar in style to a lecture with slides and a voice-over which repeats the textual information on each slide. While this video is far more informative than the previous one, with each stage of PCR clearly described, and with visually pleasing animations, it lacks in explicit primer design and again in interactivity.

Finally, an animation from the University of Utah, titled “PCR Virtual Lab” [22]. This is a much more interactive experience and allows the user to use virtual pipettes in order to simulate what you would do in a lab situation when performing PCR. Additionally, the information it provides, while slightly basic in the beginning for our target users, is extensive and very informative to the novice user, such as Biology-illiterate Computing Scientists. While this is a much more interactive and, compared to the alternatives described above, much more informative experience, it fails to provide the user with the theoretical background information, particularly on primer design (required to fully understand the process and why the reaction occurs), and does not allow the user to test their ability to select good primers, arguably the most intellectually challenging aspect of PCR.

Chapter 2

Design

2.1 Requirements

Inital Requirements Gathering

The requirements gathering process for the application began immediately. At the first meeting, our clients presented us with a document outlining what it was that they wanted from the end product, including a very early step-by-step walkthrough of the application they envisioned. This proved to be a key tool in bringing us up to speed with what the system should accomplish, and really sped up the initial requirements gathering phase. This design was altered and adapted throughout the project, but the steps served to provide a rough guidline that we followed throughout development. The aim of the project, as described in the aforementioned document, are as follows:

> To design a PCR-primer design exercise to complement teaching of a Molecular Methods course to Level 3 Life Sciences Undergraduates. This exercise will be integrated into a new part of the lab which we are designing based around diagnosis of HIV using PCR. You will need to understand the theory behind PCR and primer design in order to achieve this.

This statement alone is helpful as it tells us about our userbase, where and how the application will be used, and what background knowledge is required in order to thoroughly understand the premise of the project.

On the subject of background knowledge, along with this document, we were given the Molecular Methods lab book, in order to see how Primer Design is currently taught in the course and get a better idea of how it worked ourselves.

Finally, within the first two or three meetings we were sent links to various multimedia teaching tools for Primer Design, as described in Section 1.4. Along with our own research, this gave us an informed view of the current climate of PCR teaching tools, the positives and negatives of these current approaches, and what we could improve upon in our own product.

From these first few weeks of meetings with the clients, we drafted a requirements document, which was presented to the clients and agreed upon, and presented the following requirements:

System Scope

- The main aim of this system is to act as a teaching tool to aid students in learning how to design primers for PCR experiments and should be usable in a teaching environment or by people on their home computers.
- It should function as an interactive, step-by-step guide through the process of PCR on a DNA strand of the user's choice. The user is required to access the NCBI website [10] and copy and paste their choice of DNA sequence into the system. The system should provide feedback if the user enters incorrect primers. The system should then check if the melting temperatures of the primers are in the required range, using the equation provided to us by the clients. This equation should be visible to the user. The user is then given a link to perform primer BLAST (described in Section 1.1) to check if the primers they have chosen are unique.
- The system should also provide the user help with completing each task by providing relevant rules for each task and giving the user instructions about how to use websites and resources outwith the system (NCBI [10], primer BLAST [16] etc.).
- When the user has provided an appropriate pair of primers the system will then show an animation of the PCR reaction taking place.

Non-Functional Requirements

- The system is expected to be used at students homes or in the Biology lab computers, so portability is essential for the system to work to the clients expectations.

We found that while several aspects of the system deviated from the document we were handed in the first meeting, the requirements largely stayed the same throughout.

Design Feedback

Throughout the project, we maintained a weekly meeting schedule with our supervisor and clients, and despite scheduling difficulties at least one of the clients was present at every one of these meetings. This allowed us the opportunity to improve our design iteratively through multiple pitches, internalising the feedback given over the following week to produce a design more in line with their requirements.

When the team had formed a solid idea of the layout and flow of the system, we drew up some early mockups of the system's user interface (discussed in further detail in Section 2.2) and presented them to the clients during one of the weekly meetings to get their feedback on the design and how they felt it met the requirements.

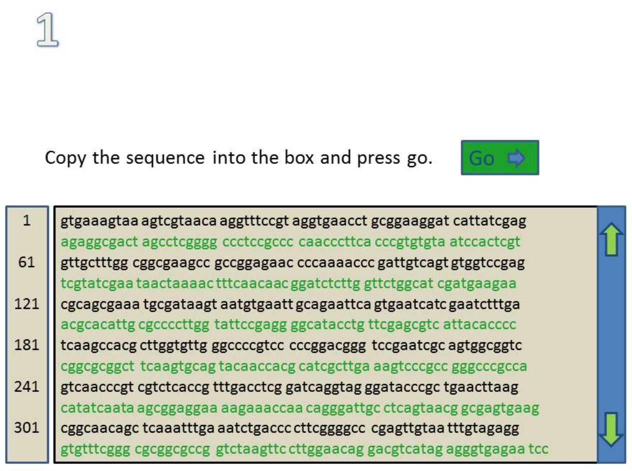
We asked a number of questions related to both the user interface and our understanding of PCR, and found that they were largely satisfied with prototype design. This prototype did not feature the animation described in the requirements, but we received valuable feedback on the validity of the system described by the prototype as a teaching tool.

Implementation Feedback

Once implementation was fully underway and the application had reached a demonstrable state, we began presenting our progress at the weekly client meetings. We received substantial feedback on how to change the UI and how the primer checks were handled in order to best improve the product as a teaching tool, and to be as accurate to the primer design process as possible.

2.2 UI

The User Interface design was developed based on the feedback and guidelines set by the clients. As it was an abstraction, we decided a rough mock-up was adequate for flexibility of the design document. Four main stages of the UI were separated into five slides which are provided and explained below.



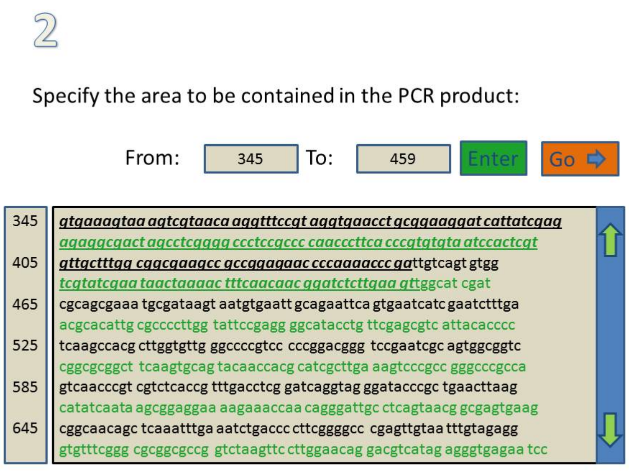
Copy the sequence into the box and press go.

```
1    gtgaaagtaa agtcgtaaca aggtttccgt aggtgaacct gcggaaggat cattatcgag
     agaggcgact agcctcgggg ccctccgccc caacccttca cccgtgtgta atccactcgt
61   gttgctttgg cggcgaagcc gccggagaac cccaaaaccc gattgtcagt gtggtccgag
     tcgtatcgaa taactaaaac tttcaacaac ggatctcttg gttctggcat cgatgaagaa
121  cgcagcgaaa tgcgataagt aatgtgaatt gcagaattca gtgaatcatc gaatctttga
     acgcacattg cgccccttgg tattccgagg ggcatacctg ttcgagcgtc attacacccc
181  tcaagccacg cttggtgttg ggccccgtcc cccggacggg tccgaatcgc agtggcggtc
     cggcgcggct tcaagtgcag tacaaccacg catcgcttga aagtcccgcc gggcccgcca
241  gtcaacccgt cgtctcaccg tttgacctcg gatcaggtag ggatacccgc tgaacttaag
     catatcaata agcggaggaa aagaaaccaa cagggattgc ctcagtaacg gcgagtgaag
301  cggcaacagc tcaaatttga aatctgaccc cttcggggcc cgagttgtaa tttgtagagg
     gtgtttcggg cgcggcgccg gtctaagttc cttggaacag gacgtcatag agggtgagaa tcc
```

Figure 2.1: Initial design, Sequence entry

Sequence entry (figure 2.1)

The first slide shows the sequence entry screen, where the user copies the DNA sequence on which to perform PCR into the provided box. The complementary strand is generated automatically and shown in different colour. The numbers on the left show the index number at the start of the corresponding line, as a way for the user to quickly find the part of the sequence to be amplified in the next stage and to provide better orientation in the sequence altogether. Scrolling through the sequence is performed either by pressing the arrow buttons on the right or using scrolling with the mouse. Once the user is content with the sequence entered, he can press the ”Go” button to advance to the next stage.



Specify the area to be contained in the PCR product:

From: 345 To: 459

```
345  gtgaaagtaa agtcgtaaca aggtttccgt aggtgaacct gcggaaggat cattatcgag
     agaggcgact agcctcgggg ccctccgccc caacccttca cccgtgtgta atccactcgt
405  gttgctttgg cggcgaagcc gccggagaac cccaaaaccc gattgtcagt gtgg
     tcgtatcgaa taactaaaac tttcaacaac ggatctcttgaa gttggcat cgat
465  cgcagcgaaa tgcgataagt aatgtgaatt gcagaattca gtgaatcatc gaatctttga
     acgcacattg cgccccttgg tattccgagg ggcatacctg ttcgagcgtc attacacccc
525  tcaagccacg cttggtgttg ggccccgtcc cccggacggg tccgaatcgc agtggcggtc
     cggcgcggct tcaagtgcag tacaaccacg catcgcttga aagtcccgcc gggcccgcca
585  gtcaacccgt cgtctcaccg tttgacctcg gatcaggtag ggatacccgc tgaacttaag
     catatcaata agcggaggaa aagaaaccaa cagggattgc ctcagtaacg gcgagtgaag
645  cggcaacagc tcaaatttga aatctgaccc cttcggggcc cgagttgtaa tttgtagagg
     gtgtttcggg cgcggcgccg gtctaagttc cttggaacag gacgtcatag agggtgagaa tcc
```

Figure 2.2: Initial design,Specification of target area

Specification of target area (figure 2.2)

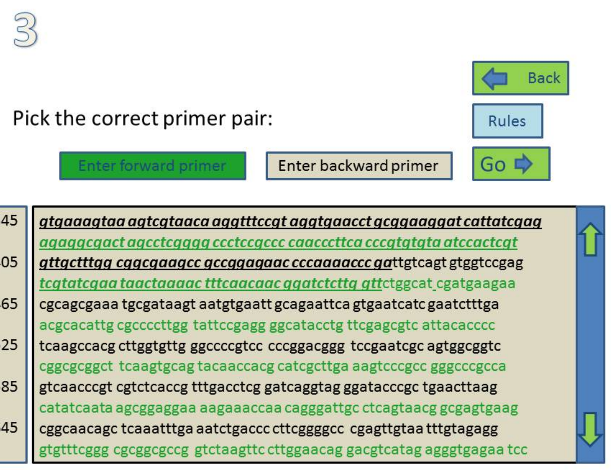
The second slide shows the selection of the DNA sequence part to be amplified by PCR. The user enters the first and the last base indices in the “From” and “To” text fields respectively, and the chosen area is highlighted after the user presses the “Enter” button. All of the previously introduced functions of the UI, including DNA sequence editing, are present at this stage also. When the user is content with the area specified, they can press the “Go” button to advance to the next stage. The “Go” button color is changed to clearly differentiate from the “Enter” button.

Primer selection (figure 2.3)

The user is prompted to copy/paste or type the forward and the reverse primers into their respective text fields located above the sequence. At the time of the interface being designed the team was still becoming accustomed to the terminology involved so the design mistakenly shows “Backward” primer instead of the correct “Reverse”. The rules of primer design can be viewed by pressing the “Rules” button. If the user decides to change the target area or the sequence itself, they can go back to the previous stage by pressing the “Back” button.

When the user has entered the primers into their respective boxes, as shown in figure 2.3, the user can press the “Go” button to go to the next stage if the primers are correct, or be shown an error message if they are not.

Each primer is checked for correctness according to the primer design rules and the user is shown a message explaining why their chosen primer is incorrect for the target sequence. In the case that



Pick the correct primer pair:

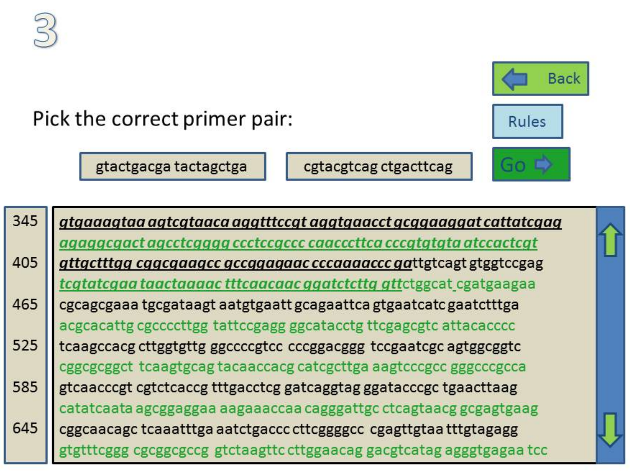
```
345 gtgaaagtaa agtcgtaaca aggtttccgt aggtgaacct gcggaaggat cattatcgag
    agaggcgact agcctcgggg ccctccgccc caacccttca cccgtgtgta atccactcgt
405 gttgctttgg cggcgaagcc gccggagaac cccaaaaccc gattgtcagt gtggtccgag
    tcgtatcgaa taactaaaac tttcaacaac ggatcttg gttctggcat cgatgaagaa
465 cgcagcgaaa tgcgataagt aatgtgaatt gcagaattca gtgaatcatc gaatctttga
    acgcacattg cgccccttgg tattccgagg ggcatacctg ttcgagcgtc attacacccc
525 tcaagccacg cttggtgttg ggccccgtcc cccggacggg tccgaatcgc agtggcggtc
    cggcgcggct tcaagtgcag tacaaccacg catcgcttga aagtcccgcc gggcccgcca
585 gtcaacccgt cgtctcaccg tttgacctcg gatcaggtag ggatacccgc tgaacttaag
    catatcaata agcggaggaa aagaaaccaa cagggattgc ctcagtaacg gcgagtgaag
645 cggcaacagc tcaaatttga aatctgaccc cttcggggcc cgagttgtaa tttgtagagg
    gtgtttcggg cgcggcgccg gtctaagttc cttggaacag gacgtcatag agggtgagaa tcc
```

Figure 2.3: Initial design, Primer selection - initial screen

both primers are incorrect, a separate list of errors is shown for each one. The user can then press the white arrow button to go back to selecting the primers again.

Melting temperature check (figure 2.6)

The last slide shows the primers chosen by the user still in their respective boxes and their melting temperatures just below. Both temperatures must be in the range of 50 to 60°C for PCR to work, so if they are not the user is suggested going back to the previous screen and selecting a different primer pair. As before, pressing “Back” will take the user to a previous stage. The user is also advised to visit the NCBI website [10] and performing primer blast on his selected primers to check them for specificity. Lastly, pressing the ”Go” button will open a new window with an animation of PCR in action.



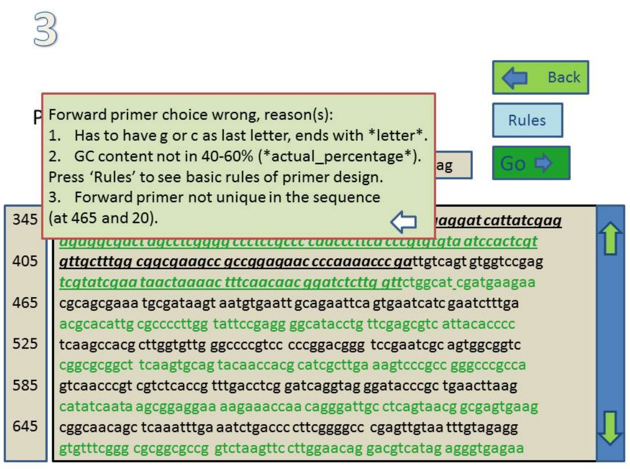
Pick the correct primer pair:

gtactgacga tactagctga

cgtacgtcag ctgacttcag

```
345  gtgaaagtaa agtcgtaaca aggtttccgt aggtgaacct gcggaaggat cattatcgag
     agaggcgact agcctcgggg ccctccgccc caacccttca cccgtgtgta atccactcgt
405  gttgctttgg cggcgaagcc gccggagaac cccaaaaccc gattgtcagt gtggtccgag
     tcgtatcgaa taactaaaac tttcaacaac ggatctcttg gttctggcat cgatgaagaa
465  cgcagcgaaa tgcgataagt aatgtgaatt gcagaattca gtgaatcatc gaatctttga
     acgcacattg cgccccttgg tattccgagg ggcatacctg ttcgagcgtc attacacccc
525  tcaagccacg cttggtgttg ggccccgtcc cccggacggg tccgaatcgc agtggcggtc
     cggcgcggct tcaagtgcag tacaaccacg catcgcttga aagtcccgcc gggcccgcca
585  gtcaacccgt cgtctcaccg tttgacctcg gatcaggtag ggatacccgc tgaacttaag
     catatcaata agcggaggaa aagaaaccaa cagggattgc ctcagtaacg gcgagtgaag
645  cggcaacagc tcaaatttga aatctgaccc cttcggggcc cgagttgtaa tttgtagagg
     gtgtttcggg cgcggcgccg gtctaagttc cttggaacag gacgtcatag agggtgagaa tcc
```

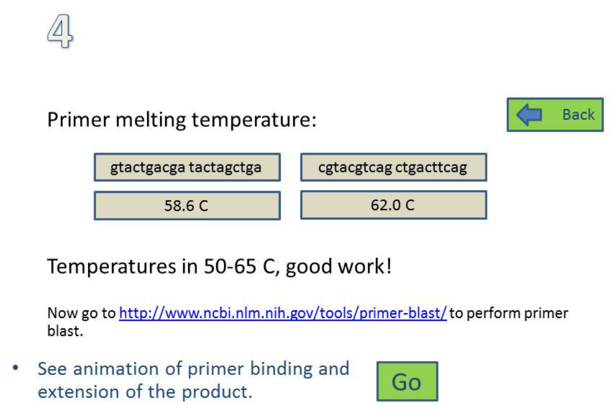
Figure 2.4: Initial design, Primer selection - primers entered



Forward primer choice wrong, reason(s):
1. Has to have g or c as last letter, ends with *letter*.
2. GC content not in 40-60% (*actual_percentage*).
Press 'Rules' to see basic rules of primer design.
3. Forward primer not unique in the sequence
(at 465 and 20).

```
345                                             aggat cattatcgag
                                                gta atccactcgt
405  gttgctttgg cggcgaagcc gccggagaac cccaaaaccc gattgtcagt gtggtccgag
     tcgtatcgaa taactaaaac tttcaacaac ggatctcttg gttctggcat cgatgaagaa
465  cgcagcgaaa tgcgataagt aatgtgaatt gcagaattca gtgaatcatc gaatctttga
     acgcacattg cgccccttgg tattccgagg ggcatacctg ttcgagcgtc attacacccc
525  tcaagccacg cttggtgttg ggccccgtcc cccggacggg tccgaatcgc agtggcggtc
     cggcgcggct tcaagtgcag tacaaccacg catcgcttga aagtcccgcc gggcccgcca
585  gtcaacccgt cgtctcaccg tttgacctcg gatcaggtag ggatacccgc tgaacttaag
     catatcaata agcggaggaa aagaaaccaa cagggattgc ctcagtaacg gcgagtgaag
645  cggcaacagc tcaaatttga aatctgaccc cttcggggcc cgagttgtaa tttgtagagg
     gtgtttcggg cgcggcgccg gtctaagttc cttggaacag gacgtcatag agggtgagaa
```

Figure 2.5: Initial design, Primer selection - error message



4

Primer melting temperature:

gtactgacga tactagctga	cgtacgtcag ctgacttcag
58.6 C	62.0 C

Temperatures in 50-65 C, good work!

Now go to http://www.ncbi.nlm.nih.gov/tools/primer-blast/ to perform primer blast.

- See animation of primer binding and extension of the product.

Figure 2.6: Initial design, Primer melting temperatures

Chapter 3

Implementation

3.1 Team Distribution

Before the team began implementing the application, we decided to split the team into three smaller sub-teams, in order to maximise the use of everyone's time. These groups were to:

- Design and implement data models and associated custom methods;
- Implement the graphical user interface;
- Design and implement an animation to show the process of PCR.

The team's lead programmer, Daniel McElroy, took the lead on this decision and, while noting each member's particular preferences, decided to split the team in the following way:

GUI Ross Eric Barnie, Murray Ross

Data Models and Custom Methods Daniel McElroy, Ross Taylor

Animation Dmitrijs Jonins

While it may have been unnecessary to assign a team-member entirely to the animation, the team felt that because at the start of the implementation stage Dmitrijs was out of the country for a prolonged period of time, that it would serve the team best to give him work he could complete in his own time.

3.2 Programming Language

When discussing implementation, the group briefly considered a small number of languages before quickly settling on Java as our implementation language. This decision was made in light of our collective experience with the language as a consequence of the Java Programming course taken in the previous academic year [7], and our knowledge of existing GUI frameworks that would suit our purposes, described below in Section 3.3.

Additionally, one of the requirements stated, in section 2.1, that portability was essential and Java is extremely portable with 1.1 billion desktop PCs [1], meaning that no matter what platform the students use, the application should run.

3.3 User Interface

The implementation of the graphical user interface (GUI) required a number of decisions to be made before writing it could begin.

3.3.1 GUI Framework

From a brief research period at the start of the implementation process, we settled on two possible options for a GUI framework to use for the application. It is important to note that other GUI options are available, but based on the team's experience, it became clear that Swing[8] or JavaFX[9] would be the most suitable.

Swing Each member of the group had some limited experience with the Swing framework, though not all of it had been positive. Each member's experience of Swing varied, and although every member had agreed that their experience had not been entirely problem-free, we conceded that its integration with the NetBeans Integrated Development Environment (IDE), discussed in section 3.3.2, was extremely useful.

However, on investigating the framework more closely it was clear that Swing was extremely well documented with full API specification [8], and in-depth tutorials [14]. This was a key element of our decision as we felt that the documentation provided would be more than adequate to enable us to use the framework with relative ease.

JavaFX Another framework considered was JavaFX, with which no member of the team had any experience. Some members felt that this was a risk worth taking, given the challenges faced in previous experiences with Swing. In reality, JavaFX was only briefly considered and totally disregarded when, upon investigation, we discovered that JavaFX was still a relatively new framework, and consequently, comprehensive documentation was not as readily available for JavaFX as with Swing, particularly when it came to troubleshooting on online forums.

In addition, JavaFX required Java 7, which, again, no member of the group had used before and which at the time was not available in the Level 3 Laboratory where we would be working for the majority of the year. It seemed like too much of a risk to try to learn two different technologies at the same time, while having to provide our own development platforms, which, with various members of the team never having used the Linux OS before, could potentially cause a number of problems.

Decision

The investigation was carried out by the group's Toolsmith, Ross Barnie, who presented the evidence discussed in the sections above regarding the two frameworks to the rest of the team. With this evidence the team voted in favor of using the Swing framework with Java 6.

Retrospectively, Swing and Java 6 are out-of-date technologies and JavaFX is now packaged with Java 7 [9], so the application would have been more up-to-date or future-proof had we used JavaFX. Additionally, many of the computers in the level 3 lab now do have Java 7 installed upon another project team requesting it, so our fears over development platform problems were nullified, though this was only after we had started development and could not have been foreseen.

It was an unfortunate shortcoming of the research into JavaFX that the group did not know about JavaFX's integration with the NetBeans IDE which was seen as one of the key differences between the two frameworks at the time of making the decision.

3.3.2 Integrated Development Environment (IDE)

One concern was that, in some members' experience, using two separate IDEs was extremely time consuming, particularly while using version control. This was mostly due to various metadata that IDEs keep track of in various files, however this meant that any small change to the source code would change the metadata and therefore each commit would have to involve adding it, which would be very time-consuming.

It is because of this experience that the group decided to work from a single IDE, researched again by Ross Barnie.

NetBeans [11] NetBeans is an IDE which the team had had little experience with and had only used in the context of building applications with GUIs created using the Swing framework. There was some trepidation to using NetBeans since most of the team had associated their problems with Swing with NetBeans itself. Upon further research, which involved using the IDE to build small applications, NetBeans started much faster than Eclipse, discussed below. The design interface was also very simple and easy to use, with each element being laid out the way you wish and the associated source code being generated for you. This meant that the design layout could be finished very quickly, rather than spending our time writing hundreds of lines of source code just for the interface.

NetBeans' metadata was minimal so would not clutter the version control repository to an unacceptable degree.

Eclipse [2]

The team had substantial knowledge of Eclipse from its mandated use in Java Programming 2 [7]. Again, our experience of Eclipse is somewhat tainted by associations with problems we faced at the time, such as a bug on the version for Windows which meant that Eclipse would freeze if you tried to copy or paste anything.

In our experience, we found Eclipse to be very slow, both during start-up and normal operation. Editing-wise, Eclipse was rather cumbersome and had few benefits over a text editor. Also, the requirement to bind the "Workspace" was seen as a potential point for confusion and errors.

In addition, the team felt that the missing design interface seen on NetBeans, discussed above, was a huge disadvantage and would cause a significant loss of time, simply due to the volume of code we would have to write instead of being auto-generated.

Members of the team also pointed out that Eclipse has a tendency to create a large amount of metadata which would clutter the version control repository.

No IDE

It was briefly considered to have no IDE at all and simply use text editors. This would allow for extremely fast editing in a very comfortable environment, since most text editors, such as Vim [15] or Emacs [3], are highly customisable and can launch in a matter of seconds. Text editors would also not require metadata, keeping our version controlled directories clean.

However, the obvious problem with no IDE is that troubleshooting source code problems without any real-time error-checking like in IDEs is more difficult and, unlike with IDEs, you cannot automatically import a missing package or method, nor can there be any auto-generated code at all for that matter.

Decision

When the evidence above was given to the team, we were also discussing which GUI Framework to use (as discussed in section 3.3.1) and it became obvious that integration with the framework would be key to helping us develop the GUI.

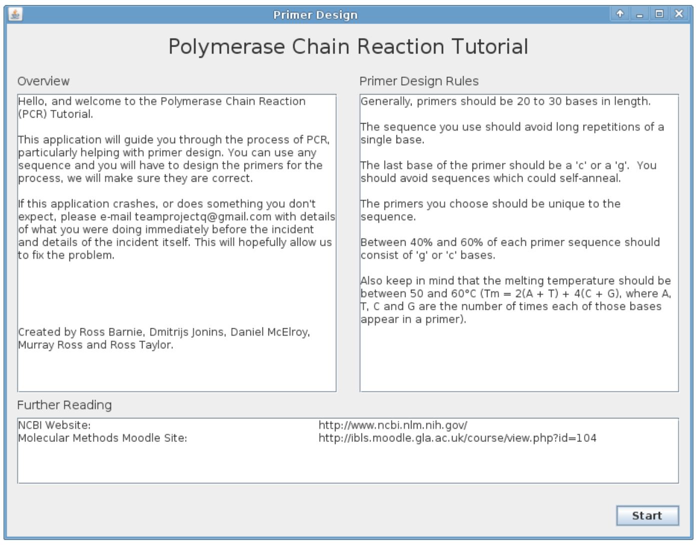
We therefore decided to work with the NetBeans IDE because of the design interface, minimal metadata, and lack of (known) bugs that would affect us in any meaningful way.

Retrospectively, this was the correct decision. Even if we had chosen a different GUI framework, the advantages of the easy-to-edit design interface far outweigh any problems we had with it.

3.3.3 Builds

To demonstrate the GUI and the changes we made to it over time, we will discuss two builds of the system at two crucial points in time.

The first is what the team refer to as the "demo build", which was the first build of the system in general to be used by anyone outwith the project. It was, at the time, the most up-to-date version of the program and is called the demo build since it was used in a demonstration (discussed in section 4.2) on 8th February 2012.



Polymerase Chain Reaction Tutorial

Overview

Hello, and welcome to the Polymerase Chain Reaction (PCR) Tutorial.

This application will guide you through the process of PCR, particularly helping with primer design. You can use any sequence and you will have to design the primers for the process, we will make sure they are correct.

If this application crashes, or does something you don't expect, please e-mail teamprojectq@gmail.com with details of what you were doing immediately before the incident and details of the incident itself. This will hopefully allow us to fix the problem.

Created by Ross Barnie, Dmitrijs Jonins, Daniel McElroy, Murray Ross and Ross Taylor.

Primer Design Rules

Generally, primers should be 20 to 30 bases in length.

The sequence you use should avoid long repetitions of a single base.

The last base of the primer should be a 'c' or a 'g'. You should avoid sequences which could self-anneal.

The primers you choose should be unique to the sequence.

Between 40% and 60% of each primer sequence should consist of 'g' or 'c' bases.

Also keep in mind that the melting temperature should be between 50 and 60°C (Tm = 2(A + T) + 4(C + G), where A, T, C and G are the number of times each of those bases appear in a primer).

Further Reading

NCBI Website:	http://www.ncbi.nlm.nih.gov/
Molecular Methods Moodle Site:	http://ibls.moodle.gla.ac.uk/course/view.php?id=104

Figure 3.1: Demo Build, Overview Panel

The second is the current build of the system, which is currently linked to on the Molecular Methods moodle site to be used by any of its 160 students. This build by nature has developed from the demo build in that most of the changes made were based on the evaluation and feedback we received from the demonstration itself (discussed in section 4.2).

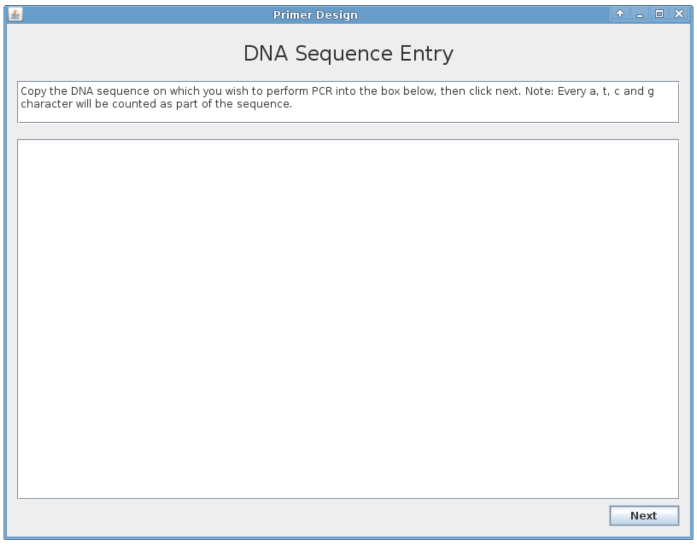
Demo Build

Overview Before the demonstration, the team were asked to include an "overview" screen to tell the user what they can expect from the application, as well as show the primer design rules to remind the user about them. This can be seen in figure 3.1.

Its design is to maximise the separation of concerns, so the Overview section is to the left, which due to the way English is read, is the more likely of the three sections to be read first, at least by fluent English readers. In this overview section it was decided to include contact details of the team in case the user found technical problems with the program since, as discussed in section 4.4, the team plan on maintaining the system for future use.

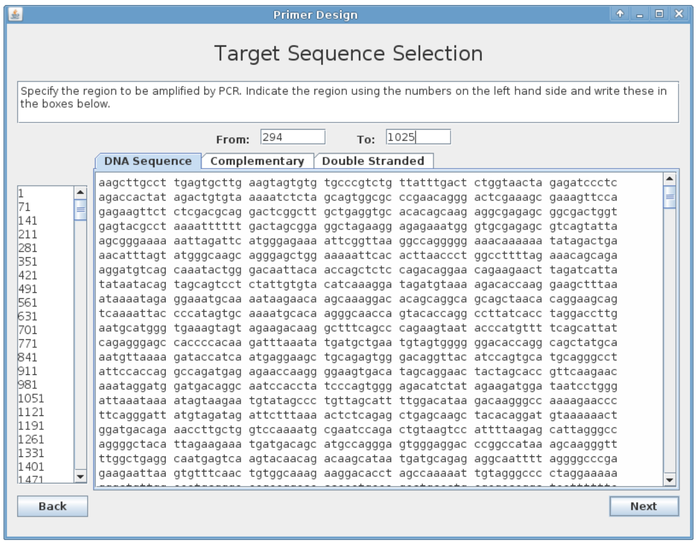
Sequence Entry Figure 3.2 shows the next panel, referred to by the team as the "sequence entry" panel. While based on the design discussed in Section 2.2 it has been altered slightly to maximise the amount of space to be used for entering in the sequence, as this is the primary purpose of this panel.

It was expected of the user to go to the National Center for Biotechnology Information ([10]) website and obtain a DNA sequence by copying it to their clipboard and then pasting this into the sequence entry panel and this was explained in the accompanying user guide (see documentation). Although this relied heavily on the users' ability to use keyboard shortcuts, it was assumed that all students at university level would at least have an awareness of these shortcuts. We also assumed



Copy the DNA sequence on which you wish to perform PCR into the box below, then click next. Note: Every a, t, c and g character will be counted as part of the sequence.

Figure 3.2: Demo Build, sequence entry panel



Specify the region to be amplified by PCR. Indicate the region using the numbers on the left hand side and write these in the boxes below.

From: 294 To: 1025

```
1
71
141
211
281
351
421
491
561
631
701
771
841
911
981
1051
1121
1191
1261
1331
1401
1471
```

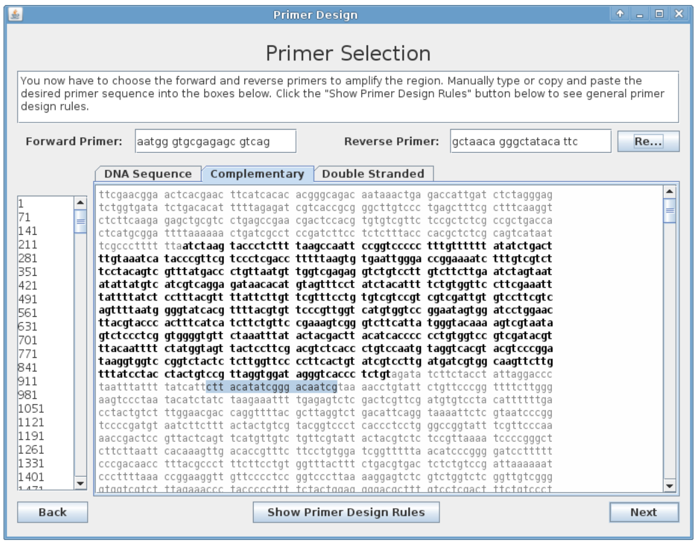
```
aagcttgcct tgagtgcttg aagtagtgtg tgcccgtctg ttatttgact ctggtaacta gagatccctc
agaccactat agactgtgta aaaatctcta gcagtggcgc ccgaacaggg actcgaaagc gaaagttcca
gagaagttct ctcgacgcag gactcggctt gctgaggtgc acacagcaag aggcgagagc ggcgactggt
gagtacgcct aaaatttttt gactagcgga ggctagaagg agagaaatgg gtgcgagagc gtcagtatta
agcgggaaaa aattagattc atgggagaaa attcggttaa ggccaggggg aaacaaaaaa tatagactga
aacatttagt atgggcaagc agggagctgg aaaaattcac acttaaccct ggccttttag aaacagcaga
aggatgtcag caaatactgg gacaattaca accagctctc cagacaggaa cagaagaact tagatcatta
tataatacag tagcagtcct ctattgtgta catcaaagga tagatgtaaa agacaccaag gaagctttaa
ataaaataga ggaaatgcaa aataagaaca agcaaaggac acagcaggca gcagctaaca caggaagcag
tcaaaattac cccatagtgc aaaatgcaca agggcaacca gtacaccagg ccttatcacc taggaccttg
aatgcatggg tgaaagtagt agaagacaag gctttcagcc cagaagtaat acccatgttt tcagcattat
cagagggagc caccccacaa gatttaaata tgatgctgaa tgtagtgggg ggacaccagg cagctatgca
aatgttaaaa gataccatca atgaggaagc tgcagagtgg gacaggttac atccagtgca tgcagggcct
attccaccag gccagatgag agaaccaagg ggaagtgaca tagcaggaac tactagcacc gttcaagaac
aaataggatg gatgacaggc aatccaccta tcccagtggg agacatctat agaagatgga taatcctggg
attaaataaa atagtaagaa tgtatagccc tgttagcatt ttggacataa gacaagggcc aaaagaaccc
ttcagggatt atgtagatag attctttaaa actctcagag ctgagcaagc tacacaggat gtaaaaaact
ggatgacaga aaccttgctg gtccaaaatg cgaatccaga ctgtaagtcc attttaagag cattagggcc
aggggctaca ttagaagaaa tgatgacagc atgccaggga gtgggaggac ccggccataa agcaagggtt
ttggctgagg caatgagtca agtacaacag acaagcataa tgatgcagag aggcaatttt aggggcccga
gaagaattaa gtgtttcaac tgtggcaaag aaggacacct agccaaaaat tgtagggccc ctaggaaaaa
```

Figure 3.3: Demo Build, Target Selection Panel

that once students were told of these shortcuts, as they were in the user guide, that they would be comfortable using them.

Target Selection Following the Sequence Entry panel is the “Target Selection” panel, shown in figure 3.3, which requires the user to specify the sequence that they wish to amplify, or the “target” sequence. In order to select the target sequence, the user must enter the index of the first base of the target into the “From” text field, and the index of the last base of the target into the “To” text field.

Finding the indices would, ideally, be aided by the text pane at the left of the screen which shows the base number of the first base on its line. Unfortunately, for unknown reasons, the text panes became misaligned when viewed from any platform other than the current development platform (the level



Primer Selection

You now have to choose the forward and reverse primers to amplify the region. Manually type or copy and paste the desired primer sequence into the boxes below. Click the "Show Primer Design Rules" button below to see general primer design rules.

Forward Primer: aatgg gtgcgagagc gtcag

Reverse Primer: gctaaca gggctataca ttc

```
1     ttcgaacgga actcacgaac ttcatcacac acgggcagac aataaactga gaccattgat ctctagggag
71    tctggtgata tctgacacat ttttagagat cgtcaccgcg ggcttgtccc tgagctttcg ctttcaaggt
141   ctcttcaaga gagctgcgtc ctgagccgaa cgactccacg tgtgtcgttc tccgctctcg ccgctgacca
211   ctcatgcgga ttttaaaaaa ctgatcgcct ccgatcttcc tctctttacc cacgctctcg cagtcataat
281   tcgccctttt ttaatctaag taccctcttt taagccaatt ccggtccccc tttgtttttt atatctgact
351   ttgtaaatca tacccgttcg tccctcgacc tttttaagtg tgaattggga ccggaaaatc tttgtcgtct
421   tcctacagtc gtttatgacc ctgttaatgt tggtcgagag gtctgtcctt gtcttcttga atctagtaat
491   atattatgtc atcgtcagga gataacacat gtagtttcct atctacattt tctgtggttc cttcgaaatt
561   tattttatct cctttacgtt ttattcttgt tcgtttcctg tgtcgtccgt cgtcgattgt gtccttcgtc
631   agttttaatg gggtatcacg ttttacgtgt tcccgttggt catgtggtcc ggaatagtgg atcctggaac
701   ttacgtaccc actttcatca tcttctgttc cgaaagtcgg gtcttcatta tgggtacaaa agtcgtaata
771   gtctccctcg gtggggtgtt ctaaatttat actacgactt acatcacccc cctgtggtcc gtcgatacgt
841   ttacaatttt ctatggtagt tactccttcg acgtctcacc ctgtccaatg taggtcacgt acgtcccgga
911   taaggtggtc cggtctactc tcttggttcc ccttcactgt atcgtccttg atgatcgtgg caagttcttg
981   tttatcctac ctactgtccg ttaggtggat agggtcaccc tctgtagata tcttctacct attaggaccc
1051  taatttattt tatcattctt acatatcggg acaatcgtaa aacctgtatt ctgttcccgg ttttcttggg
1121  aagtccctaa tacatctatc taagaaattt tgagagtctc gactcgttcg atgtgtccta catttttga
1191  cctactgtct ttggaacgac caggttttac gcttaggtct gacattcagg taaaattctc gtaatcccgg
1261  tccccgatgt aatcttcttt actactgtcg tacggtccct caccctcctg ggccggtatt tcgttcccaa
1331  aaccgactcc gttactcagt tcatgttgtc tgttcgtatt actacgtctc tccgttaaaa tccccgggct
1401  cttcttaatt cacaaagttg acaccgtttc ttcctgtgga tcggttttta acatcccggg gatccttttt
      cccgacaacc tttacgccct ttcttcctgt ggtttacttt ctgacgtgac tctctgtccg attaaaaaat
      ccctttttaaa ccggaaggtt gttcccctcc ggtcccttaa aaggagtctc gtctggtctc ggttgtcggg
```

Figure 3.4: Demo Build, Primer Design Panel

3 Computing Science Laboratory computers running Scientific Linux [12]) and this misalignment can be seen in figure 3.3.

An addition made to the original design discussed in Section 2.2 is the tabs above the main text area, which allow the user to switch between the sequence they entered, and its complementary equivalent, generated by the program. It was suggested by the clients to have this feature as it would greatly increase the speed at which the user could design the reverse primer since this is based on the complementary strand. Without the complementary tab, not only would the user have to manually convert the primer to its complementary equivalent, but also reverse its order, which neither the team or the clients felt was a useful way for students to spend their time.

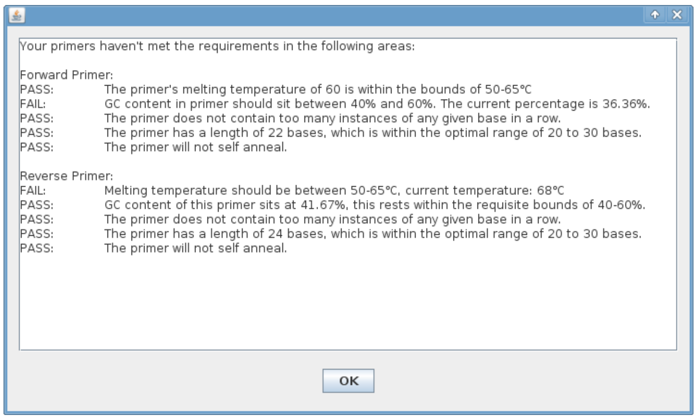
Primer Design Following the Area Selection panel is the "Primer Design" panel, shown in figure 3.4, which allows the user to enter forward and reverse primers.

One of the design features we had intended to provide was "dynamic highlighting", as it was referred to by the team. This would highlight the user-entered primers (from the "Forward Primer" and "Reverse Primer" text fields) within the sequence in the center of the panel, showing the user where their primer lies within the sequence. This highlighting was unfortunately missing in the demo build due to time constraints.

However, we had always intended to give feedback to the user should they break rules of primer design and the demo build version of this can be seen in figure 3.5 and appears when the user clicks the "Next" button. Again based on the design (section 2.2), this dialogue window shows the user any rules which they have broken, and which primer the feedback is referring to.

Again, due to time constraints this was not as fully featured as we had hoped for in the demonstration, however it did display enough information to give an idea to our clients of what the feedback might look like in the future (see further discussion in section 4.2).

In order to design a reverse primer outwith the system, the complementary strand would have to

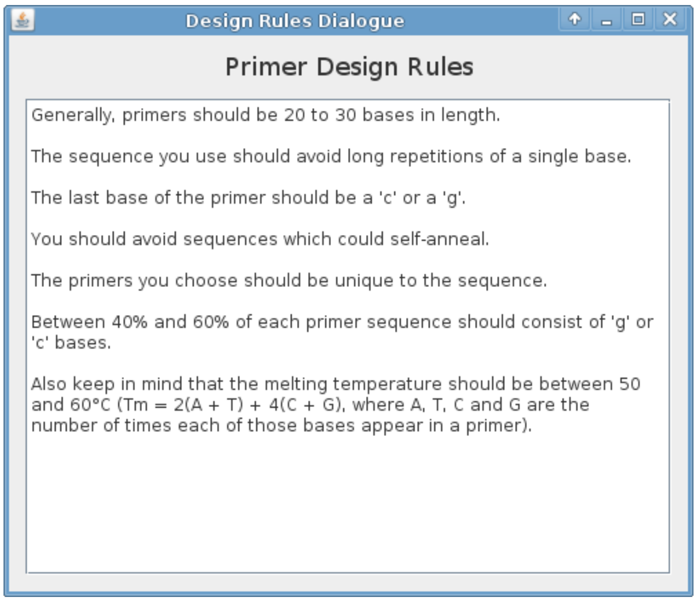


```
Your primers haven't met the requirements in the following areas:

Forward Primer:
PASS:		The primer's melting temperature of 60 is within the bounds of 50-65℃
FAIL:		GC content in primer should sit between 40% and 60%. The current percentage is 36.36%.
PASS:		The primer does not contain too many instances of any given base in a row.
PASS:		The primer has a length of 22 bases, which is within the optimal range of 20 to 30 bases.
PASS:		The primer will not self anneal.

Reverse Primer:
FAIL:		Melting temperature should be between 50-65℃, current temperature: 68℃
PASS:		GC content of this primer sits at 41.67%, this rests within the requisite bounds of 40-60%.
PASS:		The primer does not contain too many instances of any given base in a row.
PASS:		The primer has a length of 24 bases, which is within the optimal range of 20 to 30 bases.
PASS:		The primer will not self anneal.
```

Figure 3.5: Demo Build, Feedback on User-entered Primer



Primer Design Rules

Generally, primers should be 20 to 30 bases in length.

The sequence you use should avoid long repetitions of a single base.

The last base of the primer should be a 'c' or a 'g'.

You should avoid sequences which could self-anneal.

The primers you choose should be unique to the sequence.

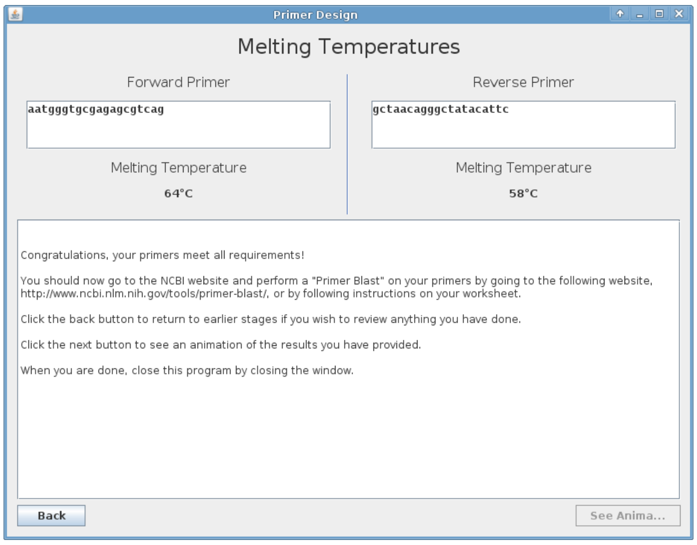
Between 40% and 60% of each primer sequence should consist of 'g' or 'c' bases.

Also keep in mind that the melting temperature should be between 50 and 60°C (Tm = 2(A + T) + 4(C + G), where A, T, C and G are the number of times each of those bases appear in a primer).

Figure 3.6: Demo Build, Primer Design Rules Dialogue

be calculated (which would be in the 3'—5' direction) and reversed to be in the correct direction (5'—3'). In the application, the complementary strand is already calculated, so the user can simply copy and paste a primer of their choosing from the sequence and put it in the reverse primer text field. However, this does not solve the problem of reversing it, so the "Reverse" button was put next to the reverse primer text field, which intuitively reverses the order of the primer in the text field.

At the bottom of the panel in the middle of the "Back" and "Next" buttons is the "Primer Design Rules" button, which shows, intuitively enough, the primer design rules set out at the beginning of the program. This was part of the initial design, the button for which can be seen in figure 2.3 and the implementation of which can be seen in figure 3.6.



Melting Temperatures

Forward Primer

aatgggtgcgagagcgtcag

Melting Temperature

64°C

Reverse Primer

gctaacagggctatacattc

Melting Temperature

58°C

Congratulations, your primers meet all requirements!

You should now go to the NCBI website and perform a "Primer Blast" on your primers by going to the following website, http://www.ncbi.nlm.nih.gov/tools/primer-blast/, or by following instructions on your worksheet.

Click the back button to return to earlier stages if you wish to review anything you have done.

Click the next button to see an animation of the results you have provided.

When you are done, close this program by closing the window.

Figure 3.7: Demo Build, Melting Temperatures of User's Primers.

Melting Temperature After designing a primer, the user is presented with the "Melting Temperature" panel as seen in figure 3.7 for the user to evaluate their primers.

In the case of the demonstration, this panel was blocked from the user unless they had a correct primer.

While the initial design 2.2 was sufficient as a mock-up, it lacked consistency with the rest of the program. It also proved difficult to reproduce in Swing with the NetBeans Design view while maintaining the professional appearance of the rest of the program. Initial attempts had JLabel objects, all misaligned, spread very sparsely around the panel and looked very rushed and unprofessional.

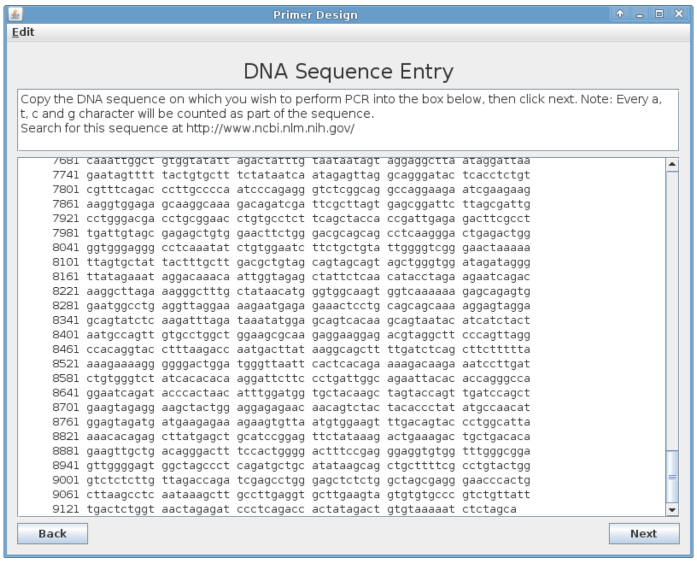
The demo build version addresses the problems we faced in these initial attempts by using a single large text box to display all the necessary information to the user about possible future work and the upper section of the panel to provide the user with the ability to review their primers. Thus reducing the amount of unused space on the panel, while also keeping it consistent (with the traversal buttons on the bottom of the panel) with the rest of the program.

The emphasis on the primers and the melting temperatures in bold means that the user can easily see their primers and the associated melting temperatures, while the separator down the center visually separates the forward from the reverse primer.

Unfortunately the animation was not available in time for the demonstration and although the button for it was included in this panel it was disabled.

Current Build

Addressing feedback from the demonstration (see section 4.2), and adding new elements to provide extra functionality, the current build is several iterations ahead of the demo build discussed above and is currently available to all Molecular Methods students.



DNA Sequence Entry

Copy the DNA sequence on which you wish to perform PCR into the box below, then click next. Note: Every a, t, c and g character will be counted as part of the sequence.
Search for this sequence at http://www.ncbi.nlm.nih.gov/

```
     7681 caaattggct gtggtatatt agactatttg taataatagt aggaggctta ataggattaa
     7741 gaatagtttt tactgtgctt tctataatca atagagttag gcagggatac tcacctctgt
     7801 cgtttcagac ccttgcccca atcccagagg gtctcggcag gccaggaaga atcgaagaag
     7861 aaggtggaga gcaaggcaaa gacagatcga ttcgcttagt gagcggattc ttagcgattg
     7921 cctgggacga cctgcggaac ctgtgcctct tcagctacca ccgattgaga gacttcgcct
     7981 tgattgtagc gagagctgtg gaacttctgg gacgcagcag cctcaaggga ctgagactgg
     8041 ggtgggaggg cctcaaatat ctgtggaatc ttctgctgta ttggggtcgg gaactaaaaa
     8101 ttagtgctat tactttgctt gacgctgtag cagtagcagt agctgggtgg atagataggg
     8161 ttatagaaat aggacaaaca attggtagag ctattctcaa catacctaga agaatcagac
     8221 aaggcttaga aagggctttg ctataacatg ggtggcaagt ggtcaaaaaa gagcagagtg
     8281 gaatggcctg aggttaggaa aagaatgaga gaaactcctg cagcagcaaa aggagtagga
     8341 gcagtatctc aagatttaga taaatatgga gcagtcacaa gcagtaatac atcatctact
     8401 aatgccagtt gtgcctggct ggaagcgcaa gaggaaggag acgtaggctt cccagttagg
     8461 ccacaggtac ctttaagacc aatgacttat aaggcagctt ttgatctcag cttcttttta
     8521 aaagaaaagg ggggactgga tgggttaatt cactcacaga aaagacaaga aatccttgat
     8581 ctgtgggtct atcacacaca aggattcttc cctgattggc agaattacac accagggcca
     8641 ggaatcagat acccactaac atttggatgg tgctacaagc tagtaccagt tgatccagct
     8701 gaagtagagg aagctactgg aggagagaac aacagtctac tacaccctat atgccaacat
     8761 ggagtagatg atgaagagaa agaagtgtta atgtggaagt ttgacagtac cctggcatta
     8821 aaacacagag cttatgagct gcatccggag ttctataaag actgaaagac tgctgacaca
     8881 gaagttgctg acagggactt tccactgggg actttccgag ggaggtgtgg tttgggcgga
     8941 gttggggagt ggctagccct cagatgctgc atataagcag ctgcttttcg cctgtactgg
     9001 gtctctcttg ttagaccaga tcgagcctgg gagctctctg gctagcgagg gaacccactg
     9061 cttaagcctc aataaagctt gccttgaggt gcttgaagta gtgtgtgccc gtctgttatt
     9121 tgactctggt aactagagat ccctcagacc actatagact gtgtaaaaat ctctagca
```

Figure 3.8: Current Build, Sequence Entry Panel

While there have been many changes to the build in this time, there have been few alterations to the UI as most feedback about it at the demonstration was positive. Most changes stem from evolving requirements or from the demonstration feedback.

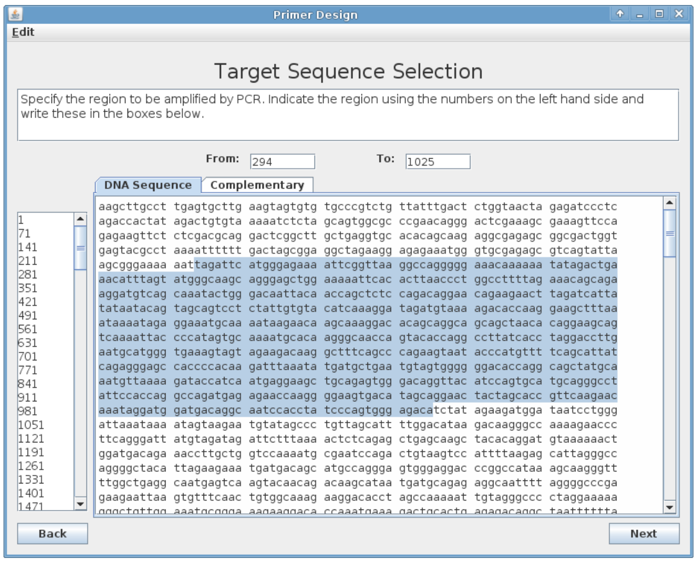
Overview The overview screen (seen in figure 3.1) changed only because of the addition of the menu bar, discussed below. It was felt that this needed very little change, if any, from the demo build and is therefore identical to the demo build version in almost every respect.

Sequence Entry In terms of changes made after the demo build, this panel is similar to the Overview panel above in that neither have been changed to any significant degree.

However, as can be seen in figure 3.8 there have been some minor changes.

- The menu bar at the top of the panel, which was added to address the issue of people not being familiar with keyboard shortcuts. The menu itself was named “Edit” to conform to standards set by most applications with menu bars in that the “Copy” and “Paste” features are usually found under the “Edit” menu.
- The “Back” button on this panel which was not present for the demo build. For the demo build, the inclusion of a back button felt unnecessary as any information that users would require in order to use the program is available throughout the application (for example, Primer Design Rules is available on Primer Design panel, and the NCBI website [10] is given to user in the instructions at top of panel). However, the clients felt the “Back” button was a necessary inclusion, and we were more than happy to defer to their insight.

Target Selection While the interface of the Target Selection panel may not have changed drastically from the demo build, the way a user selects their desired area of the sequence has changed to



Target Sequence Selection

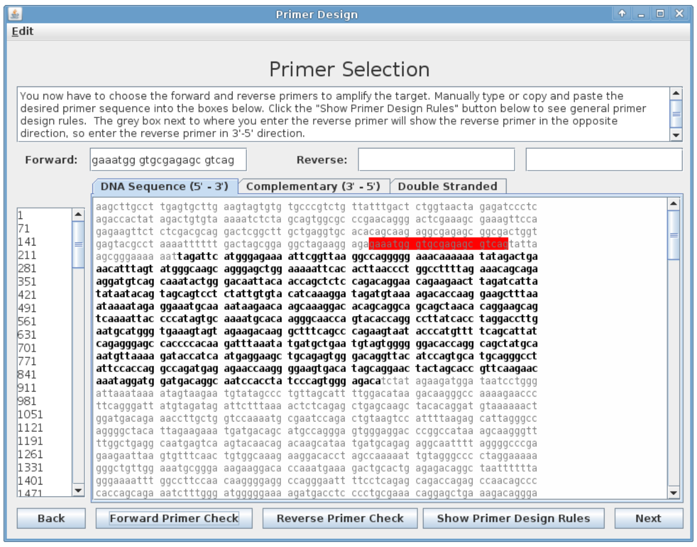
Specify the region to be amplified by PCR. Indicate the region using the numbers on the left hand side and write these in the boxes below.

From: 294 To: 1025

```
1
71
141
211
281
351
421
491
561
631
701
771
841
911
981
1051
1121
1191
1261
1331
1401
1471
```

```
aagcttgcct tgagtgcttg aagtagtgtg tgcccgtctg ttatttgact ctggtaacta gagatccctc
agaccactat agactgtgta aaaatctcta gcagtggcgc ccgaacaggg actcgaaagc gaaagttcca
gagaagttct ctcgacgcag gactcggctt gctgaggtgc acacagcaag aggcgagagc ggcgactggt
gagtacgcct aaaatttttt gactagcgga ggctagaagg agagaaatgg gtgcgagagc gtcagtatta
agcgggaaaa aattagattc atgggagaaa attcggttaa ggccaggggg aaacaaaaaa tatagactga
aacatttagt atgggcaagc agggagctgg aaaaattcac acttaaccct ggccttttag aaacagcaga
aggatgtcag caaatactgg gacaattaca accagctctc cagacaggaa cagaagaact tagatcatta
tataatacag tagcagtcct ctattgtgta catcaaagga tagatgtaaa agacaccaag gaagctttaa
ataaaataga ggaaatgcaa aataagaaca agcaaaggac acagcaggca gcagctaaca caggaagcag
tcaaaattac cccatagtgc aaaatgcaca agggcaacca gtacaccagg ccttatcacc taggaccttg
aatgcatggg tgaaagtagt agaagacaag gctttcagcc cagaagtaat acccatgttt tcagcattat
cagagggagc caccccacaa gatttaaata tgatgctgaa tgtagtgggg ggacaccagg cagctatgca
aatgttaaaa gataccatca atgaggaagc tgcagagtgg gacaggttac atccagtgca tgcagggcct
attccaccag gccagatgag agaaccaagg ggaagtgaca tagcaggaac tactagcacc gttcaagaac
aaataggatg gatgacaggc aatccaccta tcccagtggg agacatctat agaagatgga taatcctggg
attaaataaa atagtaagaa tgtatagccc tgttagcatt ttggacataa gacaagggcc aaaagaaccc
ttcagggatt atgtagatag attctttaaa actctcagag ctgagcaagc tacacaggat gtaaaaaact
ggatgacaga aaccttgctg gtccaaaatg cgaatccaga ctgtaagtcc attttaagag cattagggcc
aggggctaca ttagaagaaa tgatgacagc atgccaggga gtgggaggac ccggccataa agcaagggtt
ttggctgagg caatgagtca agtacaacag acaagcataa tgatgcagag aggcaatttt aggggcccga
gaagaattaa gtgtttcaac tgtggcaaag aaggacacct agccaaaaat tgtagggccc ctaggaaaaa
gggctgttgg aaatgcggga aagaaggaca ccaaatgaaa gactgcactg agagacaggc taattttta
```

Figure 3.9: Current Build, Target Selection Panel



Primer Selection

You now have to choose the forward and reverse primers to amplify the target. Manually type or copy and paste the desired primer sequence into the boxes below. Click the "Show Primer Design Rules" button below to see general primer design rules. The grey box next to where you enter the reverse primer will show the reverse primer in the opposite direction, so enter the reverse primer in 3'-5' direction.

Forward: gaaatgg gtgcgagagc gtcag Reverse:

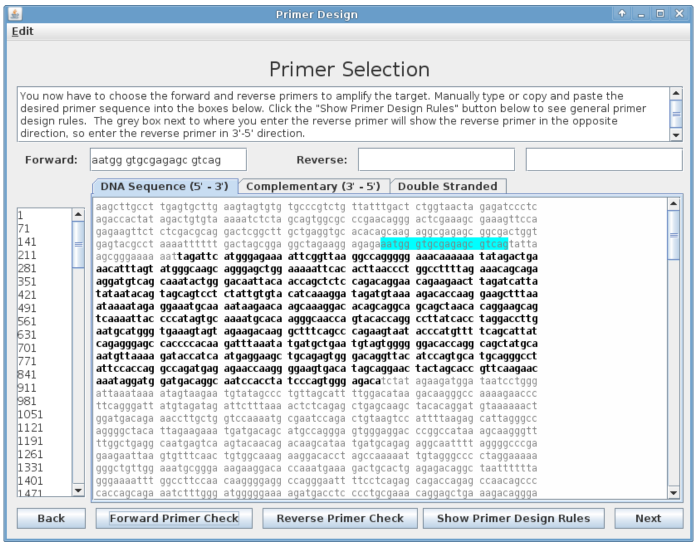
```
1
71
141
211
281
351
421
491
561
631
701
771
841
911
981
1051
1121
1191
1261
1331
1401
1471
```

```
aagcttgcct tgagtgcttg aagtagtgtg tgcccgtctg ttatttgact ctggtaacta gagatccctc
agaccactat agactgtgta aaaatctcta gcagtggcgc ccgaacaggg actcgaaagc gaaagttcca
gagaagttct ctcgacgcag gactcggctt gctgaggtgc acacagcaag aggcgagagc ggcgactggt
gagtacgcct aaaatttttt gactagcgga ggctagaagg agagaaatgg gtgcgagagc gtcagtatta
agcgggaaaa aattagattc atgggagaaa attcggttaa ggccaggggg aaacaaaaaa tatagactga
aacatttagt atgggcaagc agggagctgg aaaaattcac acttaaccct ggccttttag aaacagcaga
aggatgtcag caaatactgg gacaattaca accagctctc cagacaggaa cagaagaact tagatcatta
tataatacag tagcagtcct ctattgtgta catcaaagga tagatgtaaa agacaccaag gaagctttaa
ataaaataga ggaaatgcaa aataagaaca agcaaaggac acagcaggca gcagctaaca caggaagcag
tcaaaattac cccatagtgc aaaatgcaca agggcaacca gtacaccagg ccttatcacc taggaccttg
aatgcatggg tgaaagtagt agaagacaag gctttcagcc cagaagtaat acccatgttt tcagcattat
cagagggagc caccccacaa gatttaaata tgatgctgaa tgtagtgggg ggacaccagg cagctatgca
aatgttaaaa gataccatca atgaggaagc tgcagagtgg gacaggttac atccagtgca tgcagggcct
attccaccag gccagatgag agaaccaagg ggaagtgaca tagcaggaac tactagcacc gttcaagaac
aaataggatg gatgacaggc aatccaccta tcccagtggg agacatctat agaagatgga taatcctggg
attaaataaa atagtaagaa tgtatagccc tgttagcatt ttggacataa gacaagggcc aaaagaaccc
ttcagggatt atgtagatag attctttaaa actctcagag ctgagcaagc tacacaggat gtaaaaaact
ggatgacaga aaccttgctg gtccaaaatg cgaatccaga ctgtaagtcc attttaagag cattagggcc
aggggctaca ttagaagaaa tgatgacagc atgccaggga gtgggaggac ccggccataa agcaagggtt
ttggctgagg caatgagtca agtacaacag acaagcataa tgatgcagag aggcaatttt aggggcccga
gaagaattaa gtgtttcaac tgtggcaaag aaggacacct agccaaaaat tgtagggccc ctaggaaaaa
gggctgttgg aaatgcggga aagaaggaca ccaaatgaaa gactgcactg agagacaggc taattttta
gggaaaattt ggccttccaa caaggggagg ccagggaatt ttcctcagag cagaccagag ccaacagccc
caccagcaga aatctttggg atgggggaaa agatgacctc ccctgcgaaa caggagctga aagacaggga
```

Figure 3.10: Current Build, (Forward) Primer Design, showing invalid primer selection

make the process much easier.

Previously, as discussed in section 3.3, the user would have to find the indices of the start and end of the sequence they wanted to copy. Now, as can be seen in figure 3.9, the user simply has to highlight the desired sequence and the indices are calculated automatically. This change means that users do not have to depend on the aforementioned unreliable line numbers (see Section 4.4), and can focus instead on the actual sequence. It also provides a clearer visual reference to the user's sequence, without the need to switch between this and the next panel, which was one of the issues brought up in the demonstration (see Section 4.2).



Primer Selection

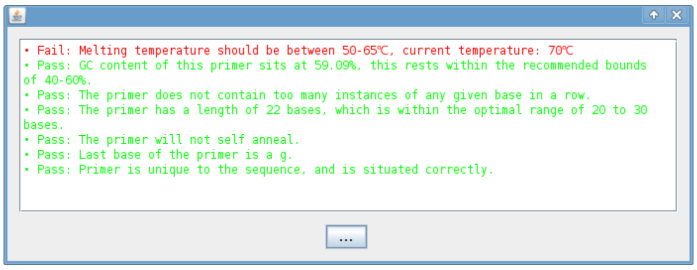
You now have to choose the forward and reverse primers to amplify the target. Manually type or copy and paste the desired primer sequence into the boxes below. Click the "Show Primer Design Rules" button below to see general primer design rules. The grey box next to where you enter the reverse primer will show the reverse primer in the opposite direction, so enter the reverse primer in 3'-5' direction.

Forward: aatgg gtgcgagagc gtcag

Reverse:

```
1     aagcttgcct tgagtgcttg aagtagtgtg tgcccgtctg ttatttgact ctggtaacta gagatccctc
71    agaccactat agactgtgta aaaatctcta gcagtggcgc ccgaacaggg actcgaaagc gaaagttcca
141   gagaagttct ctcgacgcag gactcggctt gctgaggtgc acacagcaag aggcgagagc ggcgactggt
211   gagtacgcct aaaatttttt gactagcgga ggctagaagg agagaaatgg gtgcgagagc gtcagtatta
281   agcgggaaaa aattagattc atgggagaaa attcggttaa ggccaggggg aaacaaaaaa tatagactga
351   aacatttagt atgggcaagc agggagctgg aaaaattcac acttaaccct ggccttttag aaacagcaga
421   aggatgtcag caaatactgg gacaattaca accagctctc cagacaggaa cagaagaact tagatcatta
491   tataatacag tagcagtcct ctattgtgta catcaaagga tagatgtaaa agacaccaag gaagctttaa
561   ataaaataga ggaaatgcaa aataagaaca agcaaaggac acagcaggca gcagctaaca caggaagcag
631   tcaaaattac cccatagtgc aaaatgcaca agggcaacca gtacaccagg ccttatcacc taggaccttg
701   aatgcatggg tgaaagtagt agaagacaag gctttcagcc cagaagtaat acccatgttt tcagcattat
771   cagagggagc caccccacaa gatttaaata tgatgctgaa tgtagtgggg ggacaccagg cagctatgca
841   aatgttaaaa gataccatca atgaggaagc tgcagagtgg gacaggttac atccagtgca tgcagggcct
911   attccaccag gccagatgag agaaccaagg ggaagtgaca tagcaggaac tactagcacc gttcaagaac
981   aaataggatg gatgacaggc aatccaccta tcccagtggg agacatctat agaagatgga taatcctggg
1051  attaaataaa atagtaagaa tgtatagccc tgttagcatt ttggacataa gacaagggcc aaaagaaccc
1121  ttcagggatt atgtagatag attctttaaa actctcagag ctgagcaagc tacacaggat gtaaaaaact
1191  ggatgacaga aaccttgctg gtccaaaatg cgaatccaga ctgtaagtcc attttaagag cattagggcc
1261  aggggctaca ttagaagaaa tgatgacagc atgccaggga gtgggaggac ccggccataa agcaagggtt
1331  ttggctgagg caatgagtca agtacaacag acaagcataa tgatgcagag aggcaatttt aggggcccga
1401  gaagaattaa gtgtttcaac tgtggcaaag aaggacacct agccaaaaat tgtagggccc ctaggaaaaa
1471  gggctgttgg aaatgcggga aagaaggaca ccaaatgaaa gactgcactg agagacaggc taattttta
      gggaaaattt ggccttccaa caaggggagg ccagggaatt ttcctcagag cagaccagag ccaacagccc
      caccagcaga aatctttggg atgggggaaa agatgacctc ccctgcgaaa caggagctga aagacaggga
```

Figure 3.11: Current Build, (Forward) Primer Design, showing perfect primer selection



```
• Fail: Melting temperature should be between 50-65℃, current temperature: 70℃
• Pass: GC content of this primer sits at 59.09%, this rests within the recommended bounds
of 40-60%.
• Pass: The primer does not contain too many instances of any given base in a row.
• Pass: The primer has a length of 22 bases, which is within the optimal range of 20 to 30
bases.
• Pass: The primer will not self anneal.
• Pass: Last base of the primer is a g.
• Pass: Primer is unique to the sequence, and is situated correctly.
```

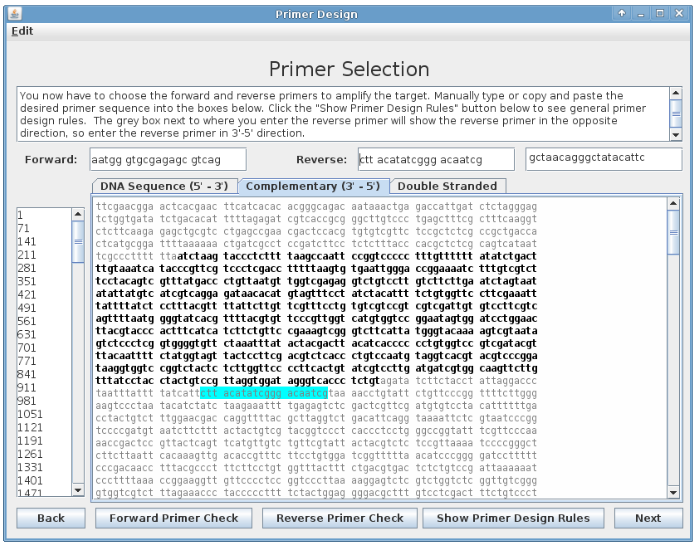
Figure 3.12: Current Build, Single Primer Feedback

Primer Design Primer design received a lot of attention in that it received the majority of new features in the application since the demo build.

Firstly, a feature referred to by the team as “dynamic highlighting” which, when a user enters a sequence of characters primer text fields, highlights that area within the larger sequence. This can be seen in figure 3.10. Initially, this highlighting was going to consist of a single colour, simply to help the user see if the primer is unique to the sequence, and so that they can visualise where the primer is within the sequence. The lack of visual aids in the demo build was something we strove to rectify especially after the feedback from the demonstration (section 4.2). To this end, we also made the highlighted area change colour depending on the correctness of the primer. How this is decided is discussed in section 3.4.3. This can be seen in figure 3.10 with an incorrect primer, and figure 3.11 with a perfect primer.

Another new feature added based on the demonstration feedback was the two new buttons at the bottom of the panel, which give feedback on a single primer, depending on which button the user presses. An example of what that feedback looks like can be seen in figure 3.12.

This feedback is also colour-coded to make any problems with the user’s primer(s) more immediately apparent than in the demo build.



Primer Selection

You now have to choose the forward and reverse primers to amplify the target. Manually type or copy and paste the desired primer sequence into the boxes below. Click the "Show Primer Design Rules" button below to see general primer design rules. The grey box next to where you enter the reverse primer will show the reverse primer in the opposite direction, so enter the reverse primer in 3'-5' direction.

Forward: aatgg gtgcgagagc gtcag

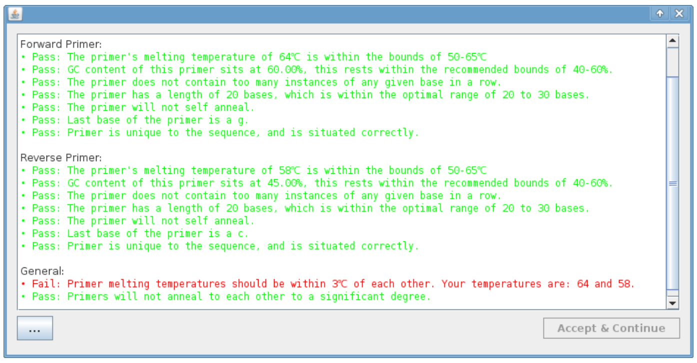
Reverse: ctt acatatcggg acaatcg

gctaacagggctatacattc

```
1
71
141
211
281
351
421
491
561
631
701
771
841
911
981
1051
1121
1191
1261
1331
1401
1471
```

```
ttcgaacgga actcacgaac ttcatcacac acgggcagac aataaactga gaccattgat ctctagggag
tctggtgata tctgacacat ttttagagat cgtcaccgcg ggcttgtccc tgagctttcg ctttcaaggt
ctcttcaaga gagctgcgtc ctgagccgaa cgactccacg tgtgtcgttc tccgctctcg ccgctgacca
ctcatgcgga ttttaaaaaa ctgatcgcct ccgatcttcc tctctttacc cacgctctcg cagtcataat
tcgccctttt ttaatctaag taccctcttt taagccaatt ccggtccccc tttgtttttt atatctgact
ttgtaaatca tacccgttcg tccctcgacc tttttaagtg tgaattggga ccggaaaatc tttgtcgtct
tcctacagtc gtttatgacc ctgttaatgt tggtcgagag gtctgtcctt gtcttcttga atctagtaat
atattatgtc atcgtcagga gataacacat gtagtttcct atctacattt tctgtggttc cttcgaaatt
tattttatct cctttacgtt ttattcttgt tcgtttcctg tgtcgtccgt cgtcgattgt gtccttcgtc
agttttaatg gggtatcacg ttttacgtgt tcccgttggt catgtggtcc ggaatagtgg atcctggaac
ttacgtaccc actttcatca tcttctgttc cgaaagtcgg gtcttcatta tgggtacaaa agtcgtaata
gtctccctcg gtggggtgtt ctaaatttat actacgactt acatcacccc cctgtggtcc gtcgatacgt
ttacaatttt ctatggtagt tactccttcg acgtctcacc ctgtccaatg taggtcacgt acgtcccgga
taaggtggtc cggtctactc tcttggttcc ccttcactgt atcgtccttg atgatcgtgg caagttcttg
tttatcctac ctactgtccg ttaggtggat agggtcaccc tctgtagata tcttctacct attaggaccc
taatttattt tatcattctt acatatcggg acaatcgtaa aacctgtatt ctgttcccgg ttttcttggg
aagtccctaa tacatctatc taagaaattt tgagagtctc gactcgttcg atgtgtccta catttttga
cctactgtct ttggaacgac caggttttac gcttaggtct gacattcagg taaaattctc gtaatcccgg
tccccgatgt aatcttcttt actactgtcg tacggtccct caccctcctg ggccggtatt tcgttcccaa
aaccgactcc gttactcagt tcatgttgtc tgttcgtatt actacgtctc tccgttaaaa tccccgggct
cttcttaatt cacaaagttg acaccgtttc ttcctgtgga tcggttttta acatcccggg gatccttttt
cccgacaacc tttacgccct ttcttcctgt ggtttacttt ctgacgtgac tctctgtccg attaaaaaat
cccttttaaa ccggaaggtt gttcccctcc ggtcccttaa aaggagtctc gtctggtctc ggttgtcggg
gtggtcgtct ttagaaaccc taccccttt tctactggag gggacgcttt gtcctcgact ttctgtccct
```

Figure 3.13: Current Build, Reverse Primer (showing perfect primer)



```
Forward Primer:
• Pass: The primer's melting temperature of 64℃ is within the bounds of 50-65℃
• Pass: GC content of this primer sits at 60.00%, this rests within the recommended bounds of 40-60%.
• Pass: The primer does not contain too many instances of any given base in a row.
• Pass: The primer has a length of 20 bases, which is within the optimal range of 20 to 30 bases.
• Pass: The primer will not self anneal.
• Pass: Last base of the primer is a g.
• Pass: Primer is unique to the sequence, and is situated correctly.

Reverse Primer:
• Pass: The primer's melting temperature of 58℃ is within the bounds of 50-65℃
• Pass: GC content of this primer sits at 45.00%, this rests within the recommended bounds of 40-60%.
• Pass: The primer does not contain too many instances of any given base in a row.
• Pass: The primer has a length of 20 bases, which is within the optimal range of 20 to 30 bases.
• Pass: The primer will not self anneal.
• Pass: Last base of the primer is a c.
• Pass: Primer is unique to the sequence, and is situated correctly.

General:
• Fail: Primer melting temperatures should be within 3℃ of each other. Your temperatures are: 64 and 58.
• Pass: Primers will not anneal to each other to a significant degree.
```

Figure 3.14: Current Build, Feedback on both primers dialogue

With the emphasis on direction of the sequence being made more obvious in this version with the direction shown in the tab names, it is more obvious as to which direction the user should be thinking about when creating the reverse primer. To allow the user to only have to think about one direction, the "Reverse" button from the demo build was replaced by an uneditable text field which auto-generates the reverse order of the reverse primer, see figure 3.13.

When the user presses the "Next" button, they are presented with a dialogue box, giving a list of all the passed, failed and "close-fail"-ed rules for the primers individually and the more general rules (see figure 3.14. Again, these are colour-coded.

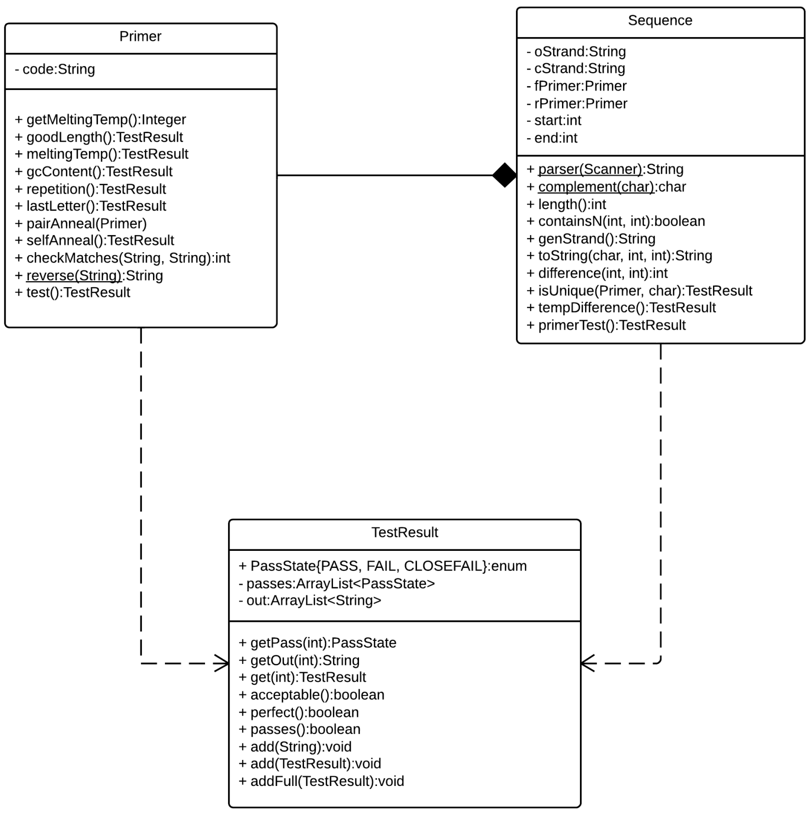


Figure 3.15: Model Class Diagram

3.4 Models & Custom Methods

3.4.1 Models

The models used in the application are few in number and do not have complex relationships with each other (as can be seen in Figure 3.15), as we strove to keep the structure as simple as possible in the interest of runtime efficiency and a clear design.

As seen in Figure 3.15, we used three classes to represent the data: Primer, TestResult and Sequence.

Primer The Primer class is purely designed to test user-designed primers. It has one instance variable, `code`, the String representing a primer which is tested against the primer test methods, which make up the remainder of the class. The class is primarily made up of methods designed to test `code` against the various rules described in Section 1.1. The method `test()` runs all above test methods and returns a TestResult indicating if the user's Primer is adequate without testing it against the opposing primer or its position in the sequence as a whole.

TestResult TestResult is a class used to format the output of one or multiple primer tests. TestResult uses an enumerated type called PassState with values `PASS`, `FAIL` and `CLOSEFAIL`, the last of which describes a state where the primer's value from a test lies outside of the recommended values, but is within 10% of a pass (in all tests with numerical boundaries) and as such is seen as acceptable, provided this is only the state of a minority of tests. TestResult uses two ArrayLists, one of PassStates (`passes`) and another of Strings (`out`), to keep track of the state and informative message to be displayed to the user for each test.

Its methods are concerned with concatenating results into larger TestResults. `perfect()` will return true if all entries in `passes` equal a `PASS`. `adequate()` will return true if the primer follows a requisite number of general design rules (60%) to function sufficiently, and is checked when deciding if a user can progress past the primer selection stage of the exercise.

Sequence This class contains two Strings, `oStrand` and `cStrand`, representing the strand of DNA that the user entered and the "complementary" strand that is generated by the `parser()` method (described below) when the sequence is constructed. Integers `start` and `end` represent the indexes of the start and end of the selected area in the sequence, and Primers `fPrimer` and `rPrimer` are representations of the user's forward and reverse primers, respectively.

`parser()` is used for both sequence entry and primer input, by taking in a String and returning a new String with all non-`atgc` characters removed. This is due to the formatting of the NCBI page, which includes line numbers that would interrupt the sequence. `complement()` is a very simple function used throughout the application that takes in one character representing a base, and returns its complement, i.e. `complement('a')` would return `t`. `isUnique()` and `tempDifference()` check the user's primers against their place in the larger sequence and the difference in their temperatures. `primerTest()` uses all other test methods to return a TestResult that will ultimately decide if the user's primer is adequate.

3.4.2 Primer Checking

The 'Primer Checks' methods are implementations of the established rules and guidelines which are used in the process of Primer Design (seen in Section 1.1) to evaluate the effectiveness of a given Primer when used in the PCR process.

As with many other aspects of the project, the primer checks were split between the two members of the 'back-end' sub-team. As well as making this task more manageable, this approach offered the added benefit of limiting the researching of primer design rules to two members, who each only had to learn how to apply the rules assigned to them.

The implementation of these rules varied widely in difficulty, from trivial checks such as "Primers should end in a base `g` or `c`" to more challenging problems such as checking how likely it is that a primer will self-anneal.

Firstly, as discussed in Section 1.4 none of the members of the team had any experience with PCR or Primer Design prior to the start of the project, and so every rule had to be thoroughly studied and understood before design of the methods could begin. However, even after spending time learning how the design rules and guidelines are used, some methods still proved problematic.

Melting Temperature

The melting temperature check was among the easiest to implement, as we were given a standard formula for calculating the temperature at which a primer would melt (`2*sum(a, t) + 4*sum(a, t)`) and the desired range in which a primer should rest (between roughly 50°C and 60°C).

Uniqueness

To check that the primer is correctly placed in the system, it is first checked that the primer cannot be found in the other strand (e.g. that the forward primer cannot be found on the complementary strand). When this is confirmed, it is then checked that the correct strand contains one and only one instance of the primer, and then that this instance is positioned correctly, relative to the area of the sequence to be replicated.

Primer Content

Many of the primer rules were trivial to implement in the system, given the string representation of a primer. Such rules include ensuring the length of the primer lies between 20 and 30 bases, checking the last base of a primer and confirming that no base is repeated more than 4 times in a row.

Annealing

As mentioned in Section 1.1, there is an increased chance of primer annealing happening if it is possible for a pair of primers (or a single primer folding over on itself) to line up in such a way as to produce 4 or more consecutive complementary bases between the primers (or the two sections of a folded over primer). To aid with the implementation of both the Self Annealing check and the Pair Annealing check a function called `checkMatches()` was created. This function takes in two primer subsequences (as strings) and returns the highest number of consecutive complementary bases in the two sequences.

Self Annealing

As described above, our `checkMatches` function compares complementary bases between two strings. Therefore, to use this function to check for annealing between two ends of a single primer the primer has to be split into two strings the second of which must be reversed before being sent to `checkMatches()` since during self annealing the primer folds over as shown in section B of Figure 1.1. To keep track of the index where the primer has been split there is a variable, `split`.

The main body of the function is a while loop which calls `checkMatches()` on the two sections of the primer which are separated by `split` then increments `split` and if loops again if `split` is at least 4 positions away from the end of the primer. During this process the highest result from the `checkMatches()` calls is recorded and this is used to determine whether or not the primer is likely to self-anneal.

Pair Annealing

This function works in a similar fashion to the Self Anneal checking method in that it compares the two primers (as opposed to two sections of one primer in Self Annealing) in every possible alignment and stores the highest number of matches.

The difference here is how the strings which are sent to the `checkMatches()` function are selected in the `pairAnneal()` function. The `pairAnneal()` function is invoked on one `Primer` instance and the other primer is passed as an argument. Firstly, the lengths of the primers are compared and the string variables `max` and `min` are set appropriately (if the primers are of the same length then which primer is set as `max` and which is the `min` is irrelevant so the primer which is passed as an argument is set as `min` and the primer which the method was invoked on is set as `max`). Then, finding the number of matches is done in three separate while loops. The first while loop checks all possible alignments when the shorter primer overlaps the longer one at the start, for example:

```
  agtcatcg
acatca
```

The second while loop checks the alignments when the shorter primer does not overlap either end of the longer primer, for example:

```
agatcgattgcagt
  agctaac
```

And the third while loop checks the alignments when the shorter primer overlaps the longer primer at the end:

```
agtacgtaggtc
     tccagtac
```

Once every possible alignment has been checked, the function uses the highest number of matches to evaluate the likelyhood of the primers annealing to each other.

3.4.3 Dynamic Primer Highlighting

Dynamic Primer Highlighting was suggested early on in the requirements elicitation process by the clients as something that would be very helpful to students studying primer design. The initial specification for this feature was as follows:

- As the user enters their choice of primer in one of the boxes at the top of the page, instances of the primer should be highlighted in real-time in the corresponding box below containing the DNA sequence for the strand on which this primer should appear.

In a later client meeting the following addition was made to improve the effectiveness of the feature:

- Primers should be highlighted in different colours to indicate their suitability.

This was a feature that all members of the team appreciated from a very early stage because, even with our limited knowledge about primer design we could see how this form of immediate feedback had the potential to improve the usability of the system if it were to be implemented well. Due to this level of popularity among both the clients and the team members the feature was given high priority. However, since we knew it would be complex to implement and would require calls to other planned modules of the system, such as the primer checking functions, it was also decided that this should be one of the last features to be implement.

This feature turned out to be one of the most problematic features to implement as it involved using features of Java which we had no real experience with, specifically the Swing classes `ChangeListener`, `Highlighter` and `Painter` as well as integrating other modules of our code to provide the required primer checking. This level of difficulty meant that this feature actually took multiple attempts to implement correctly.

Implementation

The implementation of this feature can be split into two main components:

- The code to listen for user input in the primer entry text fields and call the appropriate methods to deal with the input.
- The runnable objects `searchO` and `searchC` which are called by the listener and update the highlighted text.

Listening for Input

The first challenge when implementing this feature was deciding how to listen for user input in the `forwardPrimerTextField` and `reversePrimerTextField`.

Due to the fact that only one strand's display tab would be shown at any given time, only the TextField related to the active tab would have to be listened to and any user input in the other field could be ignored until the active tab changed. This meant that only one listener would be needed and the TextField it was listening for updates on could be switched when required.

Since this feature was implemented late on in the development cycle, code already existed to deal with changes upon switching tabs. Therefore, we were able to add the calls to switch the TextField being listened to into the existing `updateLineNums()` function which was only being used to update the line numbers upon switching between single and double-stranded views.

As well as switching the target of the listener upon switching tabs, a a call must be made to the search function related to the new tab since any user input into the corresponding TextField while the tab was not active will not yet be reflected in the display tab. This call is made using and `invokeLater()` call of the runnable search function as oppposed to a standard `invoke()` call to allow the system to continue listening for updates if a call to one of the search functions takes a long time to process.

Once we had worked out how we were going to switch the listener focus all that was left to do in terms of listening for input was to add code to the `DocumentListener` functions `insert-Update()` and `removeUpdate()` to run the appropriate search function. This just required an if statement to determine the source of the update and inside the if statement a call to the appropriate search.

Runnable Search Functions

After dealing with listening for user input, the next task was to create the functions to search for the user's primer in the DNA sequence and highlight the appropriate sections.

To perform this task we created two similar methods: `searchO()` to search the parsed version of the strand entered by the user, `parsedO`, and update forward primer highlights and `searchC()` to search the parsed complementary strand, `parsedC`, and update reverse primer highlights.

The first task these functions perform is to remove all existing highlights from the display. The other option here was to keep a list of all highlights and their positions in the sequence then check each one individually to determine if they were still valid after the latest input and highlight an extra base at the start of end of the old highlight. This list based solution was not chosen because, for example, if a user's first input was a single letter then the program would potentially have to store the positions of thousands of bases and perform thousands of comparisons on their next input.

The next step is to get the position of the first instance of the primer in the sequence using the Java `indexOf()` function, which returns the index of one String within another. This is then used as part of the condition of the while loop which carries out most of the functionality in these methods. The while loop is executed while the result of the `indexOf()` call is not negative (i.e. the primer is found in the text being searched) and the user input has a length greater than 0 (i.e. The call to the function was not because the user deleted everything in the text field).

Inside the while loop the colour ofthe current instance's highlight is determined by the following if statement from `searchC`:

```
if (sC.length() > 15){
    Primer rPrimer = new Primer(Primer.reverse(sC));
    rTest = new model.TestResult();
    rTest.addFull(rPrimer.test());
    rTest.add(
        rPrimer.isUnique(PrimerDesign.start.getInSequence(),
            'c'));
    if (rTest.perfect()){
        activePaint = perfectPaint;
    } else if (rTest.acceptable()){
        activePaint = acceptPaint;
    } else {
        activePaint = failPaint;
    }
} else {
    activePaint = failPaint;
}
```

Here, `sC` is the user's primer input and `perfectPaint`, `acceptPaint` and `failPaint` are different coloured `Painter` objects initialised when `PrimerSelectionPanel` is first loaded. The only difference between the three `Painter` objects is their colour with `perfectPaint` being blue, `acceptPaint` being yellow and `failPaint` being red. These were initially green, yellow and red but green was eventually substituted for cyan due to concerns about colour-blind users.

The late stage at which this feature was implemented meant that the process of evaluating the user's primer was a trivial task as all the methods check the primer were already written, as were the methods to evaluate the success of the primer overall. So the method simply has to create a new `Primer` instance with the user's input and create an instance of `TestResult` to store the results of the primer checking methods. Then the method calls the `perfect()` and `acceptable()` methods to determine the colour the primer should be highlighted.

Once the colour of highlight has been determined the highlight is applied to the correct section of the DNA sequence using the following code:

```
    endC = indexC + sC.length();
    highC.addHighlight(realIndex(indexC + checked, 10),
        realIndex(endC + checked, 10), activePaint);
```

`endC` is calculated by adding the length of the user's primer to the index of the current instance of the primer in the sequence, `indexC`. `realIndex` is a function which converts the indices in the parsed sequence into the indices needed for displaying the highlights in the display panes.

Once the current instance of the primer has been highlighted, a `substring()` is performed on the relevant parsed sequence to remove everything up to the end of the current primer instance. The index of the first instance of the primer in this new, shorter sequence is then found and if this index is greater than or equal to 0 then the while executes again to highlight this new instance of the primer. This is less than ideal, as it precludes the detection of matches that begin inside another match, however we are committed to finding a solution to this problem in our future work, as discussed in Section 4.4.

After all instances have been highlighted the parsed sequence is reset to the state it was in before the function was called.

3.5 Animation

Required Functionality Our initial plans for the animation were to make it depend on the actual user input in the exercise itself, particularly taking the sequence and the primers from the previous stages of the application. This was mainly to differentiate more from other similar PCR animations, like those demonstrated to us by the clients (see Section 1.4). Because of the high customisation requirements set by the task at hand and to provide maximum compatibility throughout the project we decided to develop the animation on Java utilising Java graphics package and Swing together, with the animation logic being developed from scratch. However, towards the end of the animation development it became apparent that the sequence required to be presented in the animation is just too large to be reasonably scaled with the individual bases being visible. To extrapolate, the individual bases were hard to make out as soon as the target PCR area reached about 150 bases in length. Finally, as the clients were satisfied with a static animation, it was decided that we would just use the animation with a sample input that was adequately sized. It was also suggested by the clients that the individual base's colour coding doesn't need to be explained in the animation as it was clear enough that the different colours represent different bases. Otherwise the aforementioned shift in the overarching animation design didn't affect it in any way. The final animation sreenshots and explanation of its individual parts are provided below.

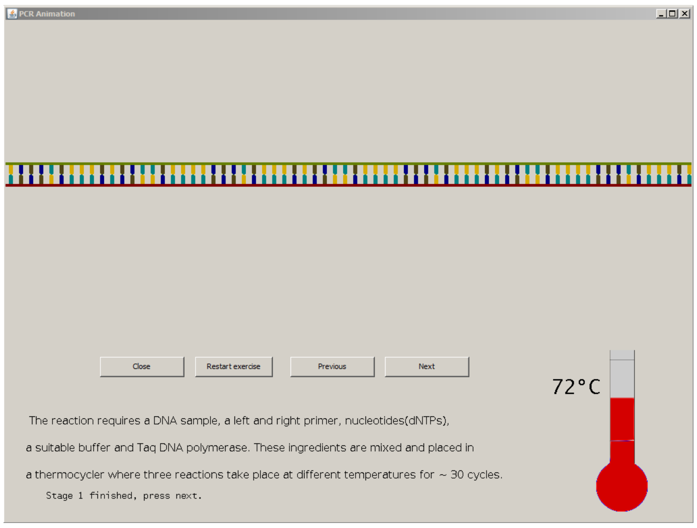
Implementation The whole animation is controlled with a large set of conditional statements by a Swing timer that constantly recalculates the time passed from the start of the animation, pausing if it reaches the end of a stage until a button press changes the stage, and therefore sets the time passed to a particular value. The models used in the animation were developed in the Inkscape vector graphics editor and are the different bases models, the Taq polymerase enzyme model and three models for the three different states of the thermometer. The individual bases are then drawn to make a sequence at a particular location.

Functionality Explained The animation is split into 7 stages, each with a related explanatory statement at the bottom. A thermometer is displayed at all times on the screen to show the current temperature at each stage of the process. Four buttons lie above the descriptive text: "Close", to close the application completely; "Restart", to go to the start of the exercise; "Previous" and "Next" to navigate between the animation stages. The animation won't proceed until the "Next" button is pressed, something which was suggested by the clients, as they pointed out that the user might not be able to read the text in the time provided. Finally, the PCR animation itself is in the top part of the screen.

Animation Screens In the stages 0 and 1, PCR is briefly explained in simple terms as the temperature rises to 72°C.

In stage 2, Melting and Annealing, the strands are separated and the primers bind to them, with the temperature level varying accordingly.

In stage 3, Adding nucleotides, the taq polymerase creates a complementary copy of each strand, with the temperature once again raised to 72°C.



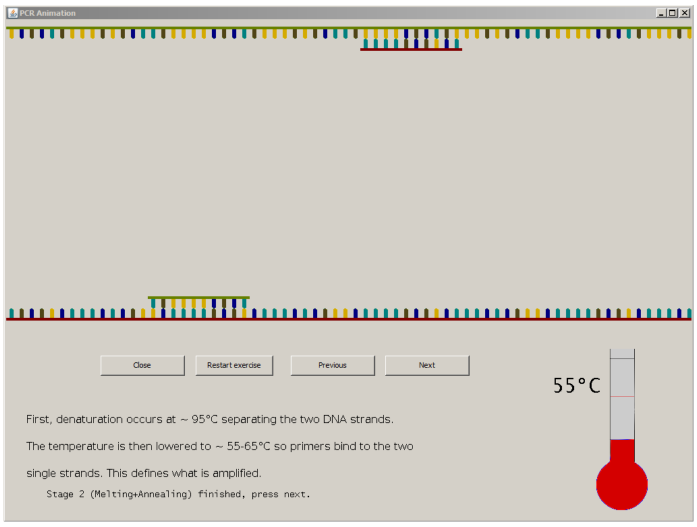
The reaction requires a DNA sample, a left and right primer, nucleotides(dNTPs),

a suitable buffer and Taq DNA polymerase. These ingredients are mixed and placed in

a thermocycler where three reactions take place at different temperatures for ~ 30 cycles.

```
Stage 1 finished, press next.
```

Figure 3.16: Animation, Stage 1



First, denaturation occurs at ~ 95°C separating the two DNA strands.

The temperature is then lowered to ~ 55-65°C so primers bind to the two

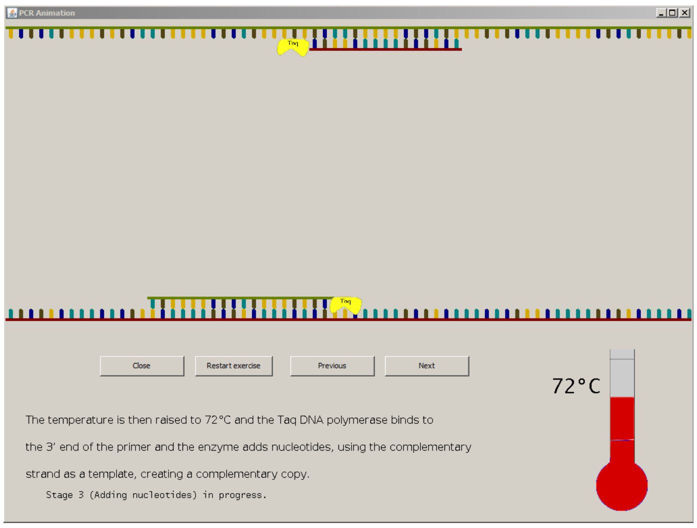
single strands. This defines what is amplified.

```
Stage 2 (Melting+Annealing) finished, press next.
```

Figure 3.17: Animation, Stage 2

In stages 4 and 5 another cycle of PCR is shown and the required sequence is generated for the first time.

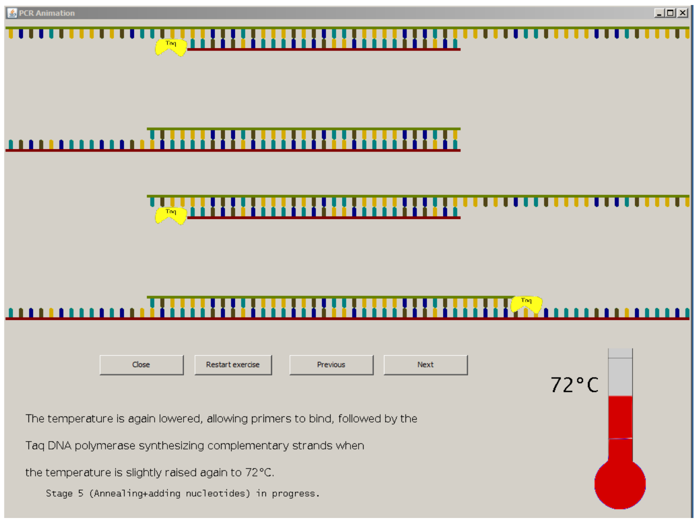
Finally, the last two stages explain how many copies of the target sequence is produced in subsequent cycles and the user is explained that they completed the exercise and thanked for participation.



The temperature is then raised to 72°C and the Taq DNA polymerase binds to the 3' end of the primer and the enzyme adds nucleotides, using the complementary strand as a template, creating a complementary copy.

```
Stage 3 (Adding nucleotides) in progress.
```

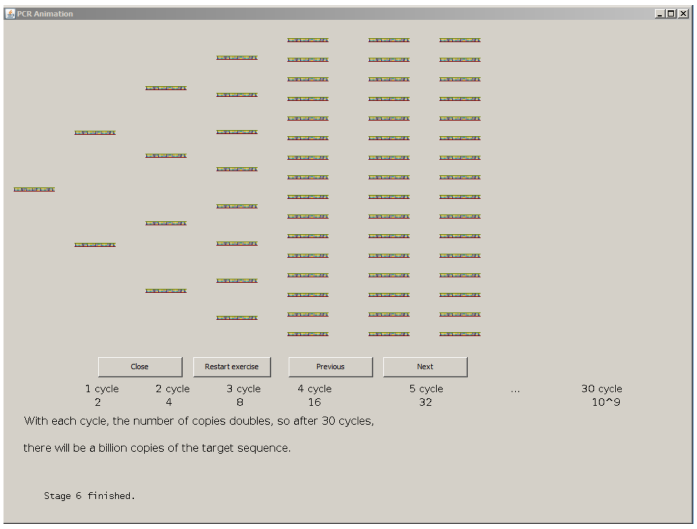
Figure 3.18: Animation, Stage 3



The temperature is again lowered, allowing primers to bind, followed by the Taq DNA polymerase synthesizing complementary strands when the temperature is slightly raised again to 72°C.

```
Stage 5 (Annealing+adding nucleotides) in progress.
```

Figure 3.19: Animation, Stage 5



1 cycle 2 cycle 3 cycle 4 cycle 5 cycle ... 30 cycle
2 4 8 16 32 10^9

With each cycle, the number of copies doubles, so after 30 cycles,

there will be a billion copies of the target sequence.

```
Stage 6 finished.
```

Figure 3.20: Animation, Stage 6

Chapter 4

Evaluation

4.1 Testing

As with any software development, testing is a necessary component. During our time with the PCR application, testing began almost as soon as implementation started. Our testing fell into two categories:

- Testing of the backend, such as primer checking formulas etc.
- Testing of the user interface.

Also, with reference to the Lethbridge textbook [18], we accommodated for both white and black box testing. Our testing had to ensure that the program met the requirements of the clients, which were gathered in several meetings with them.

4.1.1 Methods Of Testing

White Box Testing

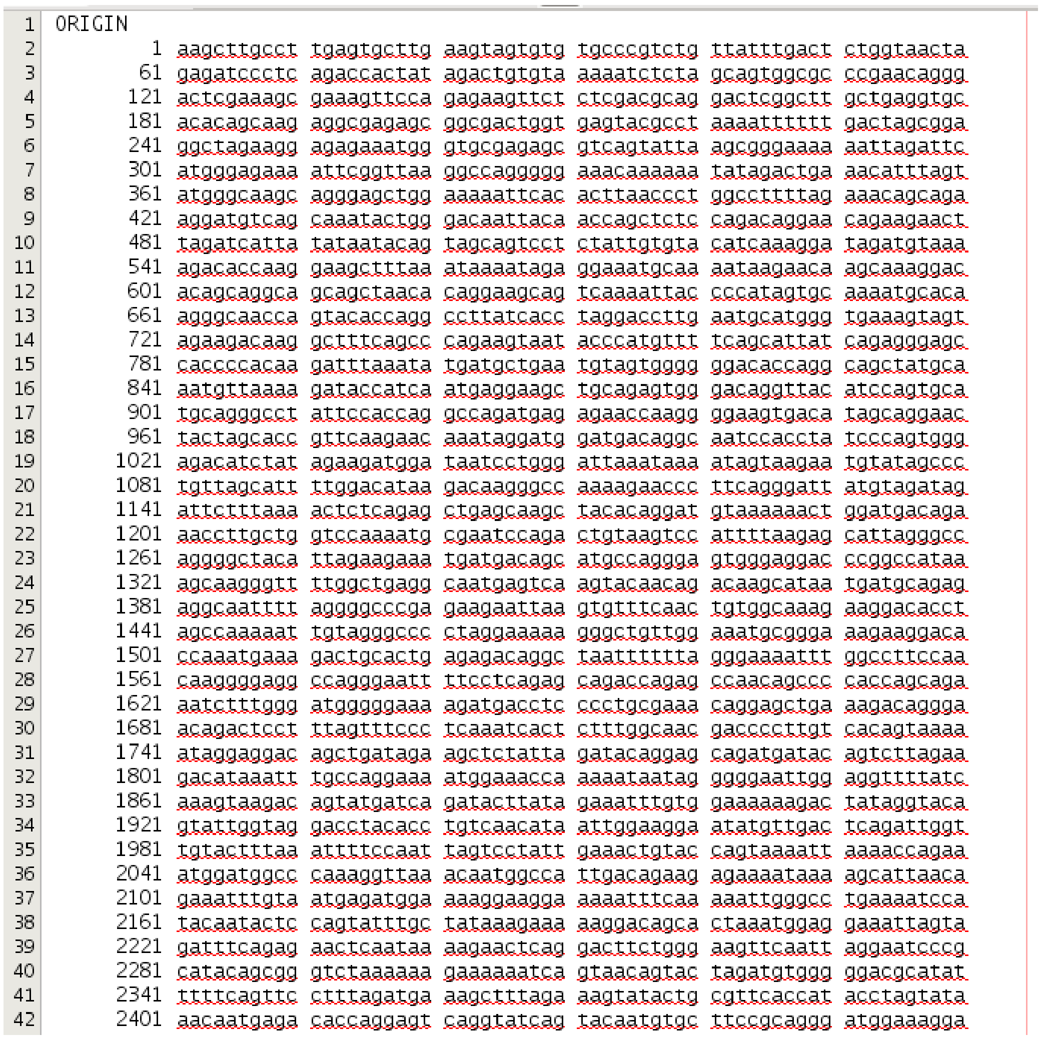
White box testing was used throughout the development of the implementation, as we all had an in-depth knowledge of the source code.

This enabled us to test for defects and errors by creating scenarios which tested the program to its limits and beyond, in order for us to assure our code could be used to check primers appropriately.

We had a default DNA sequence that we used to conduct white box testing (See: Figure 4.1). In this sequence, we knew we had good primers to test the system, and also bad primers to test the systems error checking. The following primers gave us a correct result for testing purposes:

- Forward Primer: `gagaagttctctcgacgcag`
- Reverse Primer: `cctgtattctgttcccggttt`

Due to the nature of white box testing, we were able to change the implementation when we came across errors quickly and efficiently as we knew what part of the code was failing by the nature of the error. It also allowed for us to deliberately try to create the program to produce errors, to ensure that when these errors happened for the user, it handled the situation in a safe and correct manner, and did not crash on the user.



```
ORIGIN
        1 aagcttgcct tgagtgcttg aagtagtgtg tgcccgtctg ttatttgact ctggtaacta
       61 gagatccctc agaccactat agactgtgta aaaatctcta gcagtggcgc ccgaacaggg
      121 actcgaaagc gaaagttcca gagaagttct ctcgacgcag gactcggctt gctgaggtgc
      181 acacagcaag aggcgagagc ggcgactggt gagtacgcct aaaatttttt gactagcgga
      241 ggctagaagg agagaaatgg gtgcgagagc gtcagtatta agcgggaaaa aattagattc
      301 atgggagaaa attcggttaa ggccaggggg aaacaaaaaa tatagactga aacatttagt
      361 atgggcaagc agggagctgg aaaaattcac acttaaccct ggccttttag aaacagcaga
      421 aggatgtcag caaatactgg gacaattaca accagctctc cagacaggaa cagaagaact
      481 tagatcatta tataatacag tagcagtcct ctattgtgta catcaaagga tagatgtaaa
      541 agacaccaag gaagctttaa ataaaataga ggaaatgcaa aataagaaca agcaaaggac
      601 acagcaggca gcagctaaca caggaagcag tcaaaattac cccatagtgc aaaatgcaca
      661 agggcaacca gtacaccagg ccttatcacc taggaccttg aatgcatggg tgaaagtagt
      721 agaagacaag gctttcagcc cagaagtaat acccatgttt tcagcattat cagagggagc
      781 caccccacaa gatttaaata tgatgctgaa tgtagtgggg ggacaccagg cagctatgca
      841 aatgttaaaa gataccatca atgaggaagc tgcagagtgg gacaggttac atccagtgca
      901 tgcagggcct attccaccag gccagatgag agaaccaagg ggaagtgaca tagcaggaac
      961 tactagcacc gttcaagaac aaataggatg gatgacaggc aatccaccta tcccagtggg
     1021 agacatctat agaagatgga taatcctggg attaaataaa atagtaagaa tgtatagccc
     1081 tgttagcatt ttggacataa gacaagggcc aaaagaaccc ttcagggatt atgtagatag
     1141 attctttaaa actctcagag ctgagcaagc tacacaggat gtaaaaaact ggatgacaga
     1201 aaccttgctg gtccaaaatg cgaatccaga ctgtaagtcc attttaagag cattagggcc
     1261 aggggctaca ttagaagaaa tgatgacagc atgccaggga gtgggaggac ccggccataa
     1321 agcaagggtt ttggctgagg caatgagtca agtacaacag acaagcataa tgatgcagag
     1381 aggcaatttt aggggcccga gaagaattaa gtgtttcaac tgtggcaaag aaggacacct
     1441 agccaaaaat tgtagggccc ctaggaaaaa gggctgttgg aaatgcggga aagaaggaca
     1501 ccaaatgaaa gactgcactg agagacaggc taattttta gggaaaattt ggccttccaa
     1561 caaggggagg ccagggaatt ttcctcagag cagaccagag ccaacagccc caccagcaga
     1621 aatctttggg atgggggaaa agatgacctc ccctgcgaaa caggagctga aagacaggga
     1681 acagactcct ttagtttccc tcaaatcact ctttggcaac gacccсttgt cacagtaaaa
     1741 ataggaggac agctgataga agctctatta gatacaggag cagatgatac agtcttagaa
     1801 gacataaatt tgccaggaaa atggaaacca aaaataatag ggggaattgg aggttttatc
     1861 aaagtaagac agtatgatca gatacttata gaaatttgtg gaaaaaagac tataggtaca
     1921 gtattggtag gacctacacc tgtcaacata attggaagga atatgttgac tcagattggt
     1981 tgtactttaa attttccaat tagtcctatt gaaactgtac cagtaaaatt aaaaccagaa
     2041 atggatggcc caaaggttaa acaatggcca ttgacagaag agaaaataaa agcattaaca
     2101 gaaatttgta atgagatgga aaaggaagga aaaatttcaa aaattgggcc tgaaaatcca
     2161 tacaatactc cagtatttgc tataaagaaa aaggacagca ctaaatggag gaaattagta
     2221 gatttcagag aactcaataa aagaactcag gacttctggg aagttcaatt aggaatcccg
     2281 catacagcgg gtctaaaaaa gaaaaaatca gtaacagtac tagatgtggg ggacgcatat
     2341 ttttcagttc ctttagatga aagctttaga aagtatactg cgttcaccat acctagtata
     2401 aacaatgaga caccaggagt caggtatcag tacaatgtgc ttccgcaggg atggaaagga
```

Figure 4.1: Test.txt, a sample DNA sequence

Black Box Testing

Another important testing method we utilised was black box testing, which allowed users with no knowledge of the implementation to test our program for errors and defects.

This was done by our Demo test carried out on the 8th February 2013, (See: Section 4.2), and again when the final version of the application was released, the results of which are disscussed further on in this chapter (See: Section 4.3).

Another example of black box testing is when we uploaded the final implementation on moodle, using their feedback from their experiences with it (See: Appendix B). This method of testing allowed us to gain new insights into what the program does well and what it doesn't, and allowed us to make further changes ontop of the changes already made thanks to our white box testing.

4.2 Demonstration

On the 8th February, our team carried out a demonstration run of our software with a group of users connected to the School of Life Sciences. In this test, we provided users with a comprehensive user guide and asked them to carry out a set of tasks using, what we called at the time, the demo build of the application. With this version of the program we had decided to disable the animation, as the animation had been developed using Java 7 [6], whereas the rest of the application had been developed using netbeans and Java 6, as had been decided before implementation, (See: chapter 3). However, the rest of the application we believed to be functioning at the time of the demo run, minus some major features we had yet to implement, such as the dynamic highlighting feature implemented in the final version (See: Section 3.4.3).

4.2.1 Running the demonstration

The demonstration took place between 12pm and 1pm in a computer lab in the Wolfson building, with a group of about 12 participants, consisting of students, demonstrators, lecturers, and the clients. The computers on which the application was run were all identical, all running on Windows 2000. We issued each participant with a user guide (See related documentation), detailing how to load the application from moodle and get it running, and then gave them clear instructions as to how to use the application to test for correct primers. Three members of our team (Ross Eric Barnie, Murray Ross and Ross Taylor) were on hand to help participants with any problems they may have had with the program. We let them work with the application for about half an hour before asking them to finish their work and sit down to have a round table discussion about their experiences with the system, and to fill out an evaluation sheet which we also provided them with.

4.2.2 Feedback from demonstration

At the round table discussion, we used a sound capturing device to record what people had to say about the application, which was later annotated in a text document (See: Appendix C). Also, we asked participants to fill out an evaluation feedback sheet at the end which was then read by our team and the points were collated and annotated into another text document (See: Appendix D). The main points that were taken from the feedback provided by the demonstration were:

- That the primer rules were unbreakable - you could not progress to the melting temperature screen without passing all the rules set by the program. In a typical DNA sequence, finding a

’perfect’ primer is extremely hard, and the rules should be there more as guidance, not as set in stone.

- Further to the first point, using the words ’PASS/FAIL’ is too harsh, and the capitalisation should be taken out.
- That the interface shown to the participants that whilst clear, was very clinical, very ’90s’.
- As a teaching tool, the application was definitely better than any paper version previously provided by the School of Life Sciences.
- The purpose of the application was misconstrued - it isn’t there to find primers for you, it is there for the user to find them and test them.
- It would be helpful if the program could check each primer individually, and not together at the same time.
- The application required the user to use keyboard shortcuts to copy/paste, and that a few participants did not use these shortcuts in common use and preferred graphical buttons to do these functions.

We then collected in the evaluation feedback sheets from the participants. The main points that we took from these sheets were:

- In general, it was always clear how to progress in the application from stage to the next.
- Aesthetically, the application did not have a ostentatious design but it met the demands and expectations of the user.
- The rules provided in the *Primer Design Rules box* were very helpful, but should point out these are only guidelines and not set in stone.
- If you infringed on a rule, it should allow you to pass through anyway, perhaps using an *override* button.
- Some users found it to improve their knowledge of primer design.
- It was noted that if you had a higher melting temperature in your first primer (i.e. 64) than the second (i.e. 62), but both were still in the melting temperature range, it would tell you that they were not within the required temperature range of each other, yet they clearly were.
- Reverse primers would dissapear from screen, forcing users to write them down if they wanted to remember them.
- A helpful idea would be that as the user entered their primer it highlighted it on the sequence for them to see.

With this feedback, we were able to decide on which changes were required to the program.

4.2.3 Changes proposed after feedback from demonstration

After we had met to discuss the feedback, the following changes to the application were decided upon:

- Implementation of the dynamic highlighting feature (See: Section 3.4.3) - whilst this was initially our intention during development, we had failed to implement it in time for the demo. However, from the feedback above where users were losing their primers during usage of the application, and also for general accessibility purposes, we decided implementing this feature had become a priority.
- Reduce the strictness of the primer design rules. From the feedback we received above, we found that users had previously used these rules as guidelines to follow in primer design, and that they were not necessarily adhered to all the time. Therefore, we decided that instead of requiring a 100% success rate in order to reach the melting temperature screen, we would allow users to 'override' the rules after 3 attempts, making the rules more like the guidelines users were expecting as opposed to set in stone.
- After users had notified us of the bug which tells a user that their temperatures were not in range when they were, we decided to analyse the formula for calculating the temperature in order to fix this fault.
- Some users had noted in the roundtable discussion that they would like to be able to check each primer individually, as opposed to both at the same time. To accommodate this, we decided to implement individual primer checker buttons on the primer selection panel to allow users to check each primer separately against the guidelines.
- Certain users were confused by the lack of copy/paste buttons on the user interface, and did not know the keyboard shortcuts to use in order to get around this. To allow users more accessibility, we decided to implement graphical buttons for copy/paste.
- The feedback we received indicated that they though the program was too harsh and clinical when it came to notifying users of their test results for their chosen primers. To help this, we decided to replace capital letters in the pass/fail messages to reduce harshness.

4.3 Questionnaire

In order to test the final system against the requirements set out in section 2.1, the team decided to produce a questionnaire which would be given to students using the application for them to give us feedback. The raw data for this feedback is shown in appendix B.

While other methods of testing exist (e.g. experimental) it was felt that with the time constraints in place we would not be able to run a test requiring our direct involvement and that we would receive more data from a questionnaire.

4.3.1 Questionnaire System

Many questionnaire services exist on the internet, such as SurveyMonkey [13], FreeOnlineSurveys [4], and Google Docs Forms [5].

The team decided to use the Google Form, simply because most members of the team have a Google account, and because it was free.

On closer inspection it could also export its responses to a Google Docs Spreadsheet which itself could be exported to a variety of formats.

4.3.2 Questions

Generally, the questionnaire needed to be as concise as possible to avoid any data being corrupted by frustration at the questionnaire. To this end we kept the number of questions in general to a minimum and only asked for a description if it was necessary.

Skill Level

At the demonstration, the team realised that not every user will be in the expected age bracket of 18 to 20-years-old who have used computers their entire lives. This led to the decision that the feedback should take the user's age into account, to ensure that there are no patterns in the data suggesting a particular age group struggles with the application.

In addition, we ask the user for their "confidence" with computers in general, on a scale from 0 to 5 ie no middle option so there has to be a bias to confident or not confident. This completely subjective question should allow us to see if people who perhaps do not use computers on a daily basis handle the application, since we believe we made the application (and accompanying user guide) as user-friendly as possible.

We are also aware of some colour-use in the program that could be problematic for colour-blind people so we ask the user if they are or not.

To get an idea of the user's knowledge of primer design, we ask the user for their self-assessed understanding before using the system, on a 0 to 5 scale.

All of this data provides us with the baseline skill level with computers and with primer design of the user, before they use the system.

User's Experience with the System

The following questions were designed to provide us with feedback on the user's experience with the system.

The first of these follows the self-assessed understanding of primer design before using the system, with a self-assessed understanding after using the system, on the same scale. This will provide us with test data towards the requirement to teach students about primer design and PCR.

Ideally, as discussed in section 2.1, the system would be used as a revision tool in students' own time and/or in a laboratory setting with tutors on hand to help the student with any problems. To see if we met this requirement, we asked the user if they would use the system to study one of the following options:

- Both Primer Design and PCR
- Just Primer Design
- Just PCR
- Neither

Which provides the user, and the team, with every possible answer to the question, in the most concise way possible.

Following this, we decided to ask the user if they felt they were given enough information from the application. In retrospect, this question should have been clearer in that it should have specified to disregard the user guide. We wanted to ask this to assess how easily people would be able to use the application without the user guide.

Technical difficulties were fairly common in the demo build (section 4.2) but since members of the team were present, we could instantly know about them. However since the system was now being used from where the user happened to be, we had no direct way to be told of any technical issues or bugs. To this end, we asked the user for any technical issues they found while using the application.

Lastly, we ask for any further comments, in case the user wanted to tell us something we had not asked before this point in the questionnaire. To keep it light-hearted, we recommended telling us a joke in the description of the question, needless to say we had some interesting feedback for this particular question.

4.3.3 Feedback Analysis

In total we received 15 responses, see appendix B for the raw data. These responses have yielded some interesting results with a few anomalies.

Learning

Of course, the main objective of the system is to help users revise or solidify the idea of primer design within PCR, if we have not managed this, we have failed our users.

If the application was perfect in this regard and a graph was plotted for response against score given to the “Before” and “After” understanding questions, the graph would show that the “After”

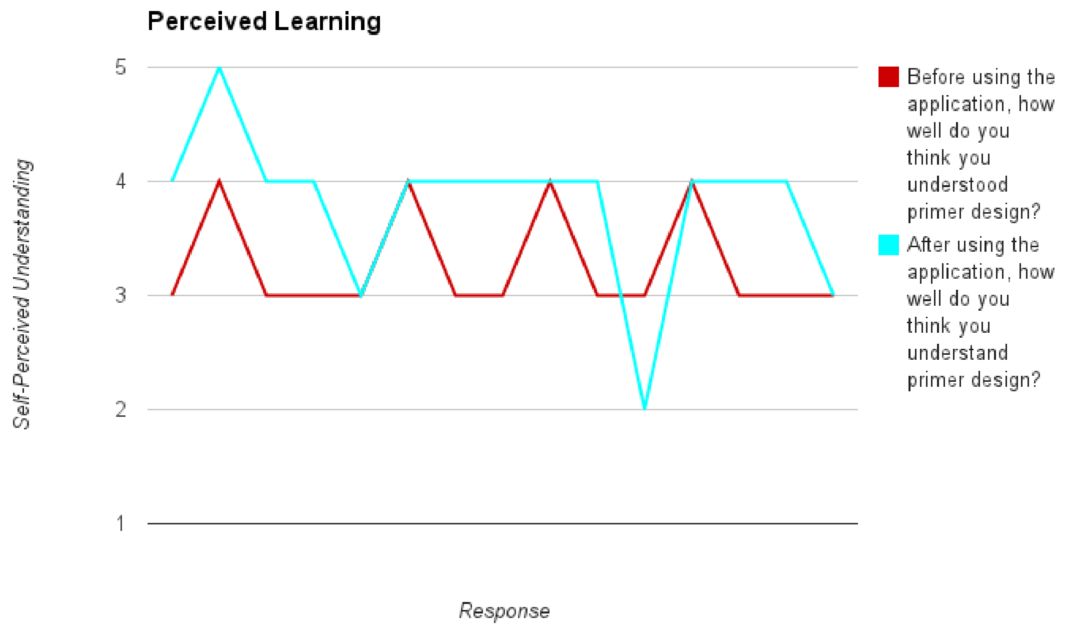


Figure 4.2: Perceived Learning Chart

responses line was consistently above the "Before" line, implying that users always see an improvement in understanding. Unfortunately, as can be seen in figure 4.2 which used the data in appendix B, this is currently not the case.

One third of responses stated that their understanding of primer design stayed the same before and after using the system and, more concerningly, one response claimed that using the system reduced their understanding. This still means that a majority of users, 60% of users who responded to the questionnaire, found using the system increased their understanding, however this still implies that 40% of users did not see an improvement in understanding.

On closer inspection of the users who saw no improvement and stated that they would not use the system for studying primer design or PCR, only one gave additional information as to their problem.

This responder stated that they were not confident in how to design primers and that they expected the system to guide them through the process. They go on to say that they had no way of getting additional information or hints as to how to progress and that there should have been some way of retrieving this information. On noting this, the team agreed that, while the "Primer Design Rules" and "Primer Feedback" functions of the system were useful to someone who had a basic understanding of primers, newer users would struggle since the rules and feedback themselves are not explicit in telling the user how they can improve their primers, though it is very heavily implied. The team agreed that while this was an isolated case, it may have been the reason for the other responses which stated no improvement in understanding, and has been added to future work (see Section 4.4).

Information From Application

Most of the responses we received to this question do not directly answer the question, since we asked for information **from the application** and most responses referred to the user guide. However, because so many (a third of respondents) state that the user guide was helpful (see appendix B), we can assume that not enough information was given by the application itself for a user to comfortably use the system without it (at least for the first time using it).

The user guide was created on the premise that the application had to familiarise people with the NCBI website [10] and we could not do this purely within the application so the user guide would provide this requirement. We were also working under the assumption that the student had never used the NCBI website [10] before. However, the team wanted to know if people found the information on the application helpful or whether the user guide could be the sole source of instruction.

In retrospect this was a null point and due to the wording of the question we have little data to support the idea of the user being unable to use the application. However, one respondent did note that they tried to use the system without the user guide and did not know how to use the NCBI website [10] to retrieve a DNA sequence, then read the user guide and found it very helpful.

Technical Difficulties Encountered

Of the 15 respondents, three mentioned something for the technical difficulties question. However, one of these responses was a problem continuing to the next panel, and does not mention anything suggesting this was a bug and not simply user error. Since no other respondent mentioned this, we are considering this as user error rather than a technical fault. We therefore have two technical difficulties to consider.

Firstly, one respondent noted that if the user was to enter the sequence as capital letters instead of lower case, the highlighting would not appear. This is a simple problem but one we intend to fix at the earliest opportunity, as noted in the Future Work chapter (Section 4.4). This respondent also comments that the animation does not appear correctly on their screen and helpfully gives us the resolution of their screen, namely 1366 x 768 pixels. Given that the height of the animation was 768, rather than the 600 used by the rest of the system, and that a small number of pixels are used for most operating systems' equivalent of a taskbar, this is perfectly understandable and is a known issue. Unfortunately the animation is hard-coded to be 768 pixels high and would require a relatively large overhaul to fix and is not something we currently intend to fix.

Lastly, another respondent stated that the animation "…did not run." on a University PC. Unfortunately, with no specification of operating system or other circumstances this will be difficult to recreate, although the team intend to investigate this further (as discussed in Section 4.4).

Shortfalls of the Questionnaire

We received no data to insinuate that a particular age group had problems with the application, with only 3 responders stating that their age was higher than 20, each with varying base skill level and responses. Therefore we do not have enough data to suggest an age-related correlation.

None of the reponding users stated that they were colour blind, so we are unable to state that colour blind people would be able to use the system as easily as those who are not.

The question regarding information from the application was evidently unclear to the students who frequently responded by talking about the user guide, rather than the application. This should have been clarified and perhaps a separate question was needed for the user guide itself, regarding its usefulness and perhaps to ask if it was too verbose.

Further Comments

We had few constructive responses for further comments (that were not already mentioned in previous questions) however we did receive some amusing comments. For example, the following joke was sent, which perhaps suggests that students in the School of Life Sciences have a worse of humour than Computing Science students.

> What did the antibody go to the Halloween costume party as?
>
> As an ”immunogobulin”!

4.4 Future Work

With regards to future work on the project, we feel there are various improvements and tweaks that could be made to the application in order to improve it. These changes can be made in the future, in order to make it a better and more complete teaching tool for the Biology department.

Also with regards to the feedback (See: Appendix B), we will continue to monitor any feedback from the students for any bugs we may need to fix, or any improvements they suggest. Whilst the feedback we recieved has been mostly positive, it has highlighted areas in which the program can be modified and improved.

4.4.1 Future Improvements

Adding a highlighting primer function on the double stranded screen. This feature was never implemented due to the time constraints of the project. However, for the final application it would certainly improve usability for the user. It would also increase familiarity for the user, as they would see the same features implemented over each primer selection screen.

Whilst we did not envisage for this to be a problem, thanks to the feedback (See: Section 4.3) provided it was clear that a user may try to enter their primer in using capital letters, for whatever reason. Therefore, a function that ensures that if the user enters a sequence of capital letters into a primer box, that it will highlight it on the DNA sequence regardless, should be implemented in the code for the program, to ensure this bug is fixed.

It has been noted that the standard resolution of the application begins at 800 x 600, but when the application reaches the animation panel it changes to 1024 x 768. A further improvement to be made therefore would be to fix the inconsistencies in the resolution, making it a standard size throughout usage.

We have discussed perhaps implementating a 'hints' system for primer design. This would try to guide the user when they enter an incorrect primer, perhaps suggesting some tips or methods to improve it, especially if it is close to passing the tests. The rules however are already provided for the user to view at their discretion however.

One issue that was raised in the questionaire feedback (See: Appendix B) was that the animation would not run on a computer located on the Glasgow University campus (They say level 8 - as there isn't many buildings with computers this high we have taken the presumption that they mean a machine in the library). Due to this, we have decided to investigate the portability of the application further, with particular regards to the animation sequence. Whilst we had tested for this beforehand more testing is required to find out the cause of this problem.

Another possible improvement to the application would be to remake the application in flash [**?**]. Flash is a global standard platform for static animations, and could possibly bring improvements to the look and feel of the application.

The font size in the application is set to a standard size, and cannot be changed. An improvement that has been suggested to the implementation is that we allow the user to change the font size, to allow more accessibility.

A bug has been noted in further testing in the dynamic highlighting function. This occurs when a sequence is entered into the box, and due to the combination of the letters two sequences overlap each other, it will only highlight the first sequence on the chosen DNA. The proposed change to the function would highlight both these sequences, in different colours to indicate the difference.

Users have commented that the program looks very clinical and simple, keeping to a functional layout. Whilst this is a strong point of the program, it could perhaps be altered slightly with such alterations as images, different colour schemes, diagrams etc.

Chapter 5

Conclusion

We introduce the things the here!

5.1 Project Successes

5.2 Project Failures

5.3 Lessons Learned

Appendix A

Glossary

TP or TP3 Team Project 3, a compulsory Level 3 Computing Science module where students are required to produce some piece of software. In our case, this system.

GitHub Refers to the website `github.com` which hosts the git repository for our project's documentation and implentation at `https://github.com/Dan-McElroy/Team-Project--Q`

Git A version control system used by Team Q to keep track of all digital files related to the system. Not to be confused with the term of 'endearment' in the English language.

Portability A system which is portable is able to be used on a different computer to the one on which it was developed with no, or minimal, changes.

pdf A file format used to display documents.

LaTeX A document mark-up language used for all project documentation.

PCR Polymerase chain reaction, a biochemical technology used to amplify pieces of DNA by several orders of magnitude, creating copies of a particular DNA sequence.

Primer A strand that serves as a starting point for DNA synthesis.

Java An object-oriented computer programming language.

IDE An integrated development environment.

Netbeans An IDE for developing primarily in Java.

Eclipse An IDE for developing primarily in Java.

Swing The primary Java GUI widget toolkit.

JavaFX A software platform for creating applications.

GUI Graphical User Interface, an interface which a user interacts with using images.

PC Personal Computer, the terminals which the application will be run on.

DNA Deoxyribonucleic acid, a molecule that contains the genetic instructions used in the functioning of living organisms and many viruses.

Base The a,t,g and c symbols that comprimise a DNA sequence.

Nucleotide Biological molecules that form the building blocks of nucleic acids.

NCBI National Center for Biotechnology Information, a branch of the National Institudes of Health. From their website, users retrieve DNA sequences.

Appendix B

Questionnaire Responses

Age	Are you colour - blind?	Confidence with Computers	Before using the application, how well do you think you understood primer design?	After using the application, how well do you think you understand primer design?	Do you feel that you were given enough information from the application?	Describe any technical problems you faced while using the application.	Any further comments	Would you use the application again to study primer design and/or PCR?
18 to 20	No	4	3	4	Yes, the PDF provided all the information required to carry out the application. Very informative, well laid out and easy to read.	No technical problems occurred	I liked the way the application highlighted the base pairs as you went, made the task much easier! Very useful application, well done :)	Just Primer Design
21 to 30	No	5	4	5			The only problem I had was with the text size in the instruction manual. It was rather uncomfortable to read as it was too small (even when zooming in). I would also have preferred the figures to be near the text where they were being referred to.	Both Primer Design and PCR
21 to 30	No	4	3	4	User guide good but could have been easier to use if the pictures were within the text.	In the primer design capital letters didn't highlight. The PCR animation didn't fit on my screen properly, it looked funny. Resolution is 1366 x 768.	My computer's my best friend you see, It's better than any girlfriend could be, Except it only has bugs, Instead of nice jugs, And a tendency to crash constantly.	Just Primer Design
18 to 20	No	4	3	4				Both Primer Design and PCR
18 to 20	No	5	3	3		keep saying the location of forward primer of wrong which i couldn't access further of the exercise		Neither

18 to 20	No	4	4	4	Yes, the instruction manual was very useful but I would put the figures closer to the relevant text- it was frustrating having to scroll right to the bottom to see the figure I was reading about in the text. I also think it would be better if the application gave better feedback than 'pass' or 'fail' - particularly how to improve on a 'fail' rather than just continually flashing this up.	n/a	useful program for studying primer design, I found it helpful. Thanks.	Both Primer Design and PCR
18 to 20	No	5	3	4			I think it would be good if sequences were provided in the program rather than entering your own, and the desired range of nucleotides that was wanting to be the PCR product was already selected	Both Primer Design and PCR
21 to 30	No	5	3	4	The first section was a little confusing when it asked you to search for a gene on NCBI as it didn't tell you where to look for this on the website. However, I then used the instruction document which was very helpful.	None	Well done :) 9/10	Both Primer Design and PCR

21 to 30	No	4	4	4	Yes, the instructions file was usefull.	The animation at the end didn't run. I was on a university PC, not sure which windows OS they use, it was on level 8.		Just Primer Design
18 to 20	No	4	3	4	YES	It all worked well		Both Primer Design and PCR
18 to 20	No	5	3	2	There was no description of how to actually design the forward and reverse primers. I wasn't very confident about this when I started, hoping that the application would guide me through the process. Instead, I couldn't progress any further than this early stage as I couldn't figure out the reverse primer sequence and there was no way of getting a hint or additional information. I feel that there should be a way of getting the correct answer after several failed attempts, so that the user can progress with the primer design exercise to see the full protocol.	n/a	A hint/help button would be useful so that the user can progress with the application instead of having to give up!	Neither
21 to 30	No	5	4	4				Neither
18 to 20	No	2	3	4				Just Primer Design
18 to 20	No	4	3	4	Yes	n/a	What did the antibody go to the Halloween costume party as? As an "immunogobulin" :D	Both Primer Design and PCR
18 to 20	No	4	3	3				Neither

Appendix C

Roundtable Feedback

Team Project Demonstration Round Table Discussion

8th February 2013

Team Member: So, can I start by asking how many people got to the melting temperature screen?
inaudible
T: So, three to four. Okay.

feedback requested
Participant: The repetition rule should tell you exactly how many you missed in a row, so that you'll know to avoid 4.
T: Alright. Any more comments before we get into the specifics?
P: Overall, the problem was that the primer design rules were unbreakable. So, maybe it should have more of a guidance system than a rule-based system.
Scott: Do you think, then, that pass/fail are the wrong words to use? **P:** Yeah, probably.
P: Or just a manual override so that you can accept that you've failed, and move on.
general agreement
P: Maybe implement something like a close fail system?
Conversation about food follows.

TEMPERATURE PANEL IN USER GUIDE SHOULD BE CHANGED

T: What did you think of the look of the program?
P: It was quite clear to figure out how to get from one point to the next. The appearance wasn't too bad.
Veitch: I thought you could have something like a DNA helix in the corner to make it look... a bit less clinical. A bit less Computer Science. No offence!
Offence taken.
P: I thought the interface was very 90s.
T: That is not our fault. If you're still using Windows XP, that's not our fault!
Le humour.

P: I thought the program was very easy - I could come into the lab and do it without any problems, or at home and I wouldn't need someone there.
P: I thought the double strand view was very helpful, and the reverse button just made things easier.
T: As a teaching tool, do you think it was better than a paper version?
unanimous affirmation

P: Why doesn't it find primers for you?
T: That's kind of the point.
P: I didn't realise it was supposed to teach me. I thought it was supposed to help me.
P: Yeah, I was using it as a program to, you know, make a primer.

P: The pass/fail breakdown is actually quite good, breaking down what's good about the primer and what's bad. **P:** A bit more detail would be good, actually.
T: Yeah, and having "FAIL" all in capital letters is perhaps a bit harsh.

P: It was good that you could amend the sequence on the fly though, if just a little bit of it was wrong you could go in and fix it.
P: Although, when I kind of passed my reverse primer, it disappeared out the box.
P: Would it be possible to have it check one of the primers first before doing the other one?

Scott: I was thinking, maybe after the temperature panel you could have it show just the sequence, in bold, with the primers at either end? Just the sequence with the actual primers, with a back button in case you want to change it.

P: Copy-pasting was all keyboard shortcuts. I don't use keyboard shortcuts.

Appendix D

Demonstration Feedback

First Demonstration - Feedback

February 12, 2013

1 Evaluation Questions

1. Was it always clear what to do to progress in the application? If it wasn't at any stage, please explain your issues:

   - Yes - but lab book needs pictures interwoven with instructions.
   - Yes - can click "run" instead of save.
   - No right-click menu for "paste" - Ctrl-V must be used. Pressing 'next' instead of return confused me for a second - hitting "return" after the "to" box to progress to the next page might be an idea.
   - The various stages are well-described. It was easy to understand the various transitions. It would have been helpful to have the figures beside the instructions.
   - It was very clear what I have to do.
   - The progress is clear.
   - Always clear :)
   - Yes, the program was easy to use.
   - Yes.

2. Aesthetically, did the application meet your demands and expectations? How could the visual aspect be improved?

   - It could, but it was okay.
   - Visually - could have a few images eg. DNA double helix.
   - It was easy to understand and it was beneficial to see the different strands, complementary otherwise.
   - Probably not be able to type into the sequence box as by accident you can write or delete nucleotides. It would be more easily to copy and paste with the mouse.
   - Same colour. Instructions should be bolder and larger.
   - While there was a clear layout, it would be handy if the pass & fail could remain open while reworking the primer. And showing where exactly your own primers start and end.
   - The design was simple and very clear. It could be improved by adding some colour, but it's not necessary. The simple design made it easy to use.
   - Very 90s, but does it the job :)
   - It isn't a fancy design but don't think it has to be.

3. Did you find the *Primer Design Rules* box to be a helpful summary of these rules? How could it be changed to provide more assistance?

   - Allow it to stay on screen while you are working.
   - Yes.

- It was useful, but it could be helpful to have a lot more detail in what would be considered a fail.
- It was really useful in order to design a correct primer.
- Good summary - should make clear that these are for guidance only and not set rules that can't be broken.
- It would be helpful if the rules were not 'set in stone' and you could be in one or two degrees difference.
- I found it helpful; right now I don't remember if you had included the melting temperature equation in the rules box, but if not, it should be added.
- Yes, summarised the rules nicely.
- Yes.

4. If you used the primer rules provided at any time, did you find the information provided useful in re-attempting the problem, and were the rule infractions covered in enough detail?

    - Rule infractions could be highlighted in the sequence and an override provided for small errors.
    - Yes.
    - Yes.
    - When it came to fixing the primers when failing then the rules proved to be a bit insufficient in helping with the new design of the primer.
    - Yes.
    - Yes. Description of where primers specifically failed was very useful.
    - PASS/FAIL is good.
    - I was aided but being out by 1 degree I thought would be fine.

5. Now that you have completed the application, do you feel your understanding of PCR and primer design techniques has improved?

    itemI think it would if this was my first time doing this.

    - -Blank-
    - Yes.
    - No.
    - Yes, much better especially at how to design a primer and choose the correct one.
    - I don't feel like they have improved, since we have already had to study these things, but it really helps you to notice how slight changes in primer sequence can change the properties(e.g. temperature) quite a lot.
    - Definitely a good tool for design assistance.

6. Did you run into any technical issues with the application? If so, please describe them.

    - No.
    - Yes - primers were at correct Tm (64 & 62) but software told me they were not within 32̆106 of each other.
    - At one point, when viewing the double-stranded sequence, the first group of 10 starting at 1051 was the opposite colour of what they should be, i.e. the top was blue and the bottom was orange.
    - No technical issues.
    - Primer disappearing when passed, no right click copy & paste.
    - No. :)
    - My reverse primer vanished after it was approved... and I hadn't written it down.

7. If you have any other comments you wish to make about the application ro the demonstration, please enter them here.
    - Generally good, simple program. Immediate feedback on primer design.
    - Allow for imperfect primers to be designed - this often happens in the real world! Sometimes they still work. On ”primer selection” page - no ”copy” & ”paste” button - most will know Ctrl-C/V, but not all students! Add back button onto ”DNA Sequence Entry” panel.
    - If the primers that you've chosen were highlighted in the sequence, that would have been helpful. If the program could check the primers one at a time?

2 Handout Notes

- ”In order to search for a compatible sequence...” not very clear about nucleotides.
- 2.2, does it have to be saved? Can we just run?
- 3.3, first paragraph: Not clear.
- 3.4, it should tell you how many repeats of a single base that you are allowed.
- Could you put in ”close fail” rather than fail. For example, if your melting temperature is just outside the range, it should give a ”close fail” so you could choose to ignore it.
- Primers failed due to self-annealing in 4 places but gives no indication where this is happening and at what temperature they would separate.
- Should have ability to override rejection manually.
- Where primer is rejected should still remain highlighted as lose place - only want to modify slightly.
- Should tell you number of bases you have selected for primer.
- Instruction text should be larger and bold.
- Start button prominence.
- Should let paste woth right-click option.
- Shuld be able to highlight DNA sequence want to use.
- No back button on page 2 (DNA Seq Entry page)
- No copy & paste function on ”primer selection” panel.
- When click on double strand view ”Primer Selection” panel the number don't move with the sequence & sequence is not highlighted.
- Should have an override button if students think that one particular parameter can be allowed.
- ”Primer Selection” window - remind them primer sequence must be 5' -¿ 3' and the view they see of complementary sequence is 3' -¿ 5'.
- What happens if you can't find a suitable sequence? - Go with what you can get!
- Allow box to get bigger - special needs.
- When error boxes come up they need expanded.

Bibliography

[1] About java. http://www.java.com/en/about/.

[2] Eclipse IDE. http://www.eclipse.org/downloads/moreinfo/java.php.

[3] Emacs website. http://www.gnu.org/software/emacs/.

[4] Freeonlinesurveys website. http://www.freeonlinesurveys.com/.

[5] Google docs overview. http://www.google.com/drive/start/apps.html.

[6] Java 7Swing API summary. http://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html.

[7] Java programming 2 moodle site. http://fims.moodle.gla.ac.uk/course/view.php?id=162.

[8] Java Swing API summary. http://docs.oracle.com/javase/6/docs/api/javax/swing/package-summary.html.

[9] JavaFX overview. http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm.

[10] Ncbi website. http://www.ncbi.nlm.nih.gov/.

[11] NetBeans IDE. http://netbeans.org/.

[12] Scientific linux (sl) website. https://www.scientificlinux.org/.

[13] Surveymonkey website. http://www.surveymonkey.com/.

[14] Trail: Creating a gui with jfc/swing. http://docs.oracle.com/javase/tutorial/uiswing/.

[15] Vim website. http://www.vim.org/index.php.

[16] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[17] John M.S. Bartlett and David Stirling. *PCR Protocols*, chapter A Short History of the Polymerase Chain Reaction. Humana Press, New York, 2003.

[18] Timothy C.Lethbridge and Robert Laganiere. *Object-Oriented Software Engineering: Practical Software Development using UML and Java, Second Edition*. McGraw Hill, second edition, 2005.

[19] Nick DeKanter. Gaming redefines interactivity for learning. *TechTrends*, 49(3):26–31, 2004.

[20] User 'DNALearningCentre'. Polymerase Chain Reaction (PCR). http://youtu.be/XXkG6m3yT1M, 2010.

[21] User 'dwildridge'. Taq Extension (test).wmv. http://youtu.be/XXkG6m3yT1M, 2012.

[22] Genetic Science Learning Center. PCR Virtual Lab. http://learn.genetics.utah.edu/content/labs/pcr/, August 2012.

[23] S. Kwok, D. H. Mack, K. B. Mullis, B. Poiesz, G. Ehrlich, D. Blair, A. Friedman-Kien, and J. J. Sninsky. Identification of human immunodeficiency virus sequences by using in vitro enzymatic amplification and oligomer cleavage detection. *Journal of Virology*, 61:1690–1694, 1987.