

AN ALGORITHMIC PROGRAMMING STRATEGY  
USING THE CLIENT/SERVER MODEL IN  
POLYMERASE CHAIN REACTION  
(PCR) PRIMER DESIGN

By

HAIHUI HUANG

Master of Physiological Science  
Oklahoma State University  
Stillwater, Oklahoma  
1998

Bachelor of Medical Science  
Hunan Medical University  
Changsha, Hunan  
People's Republic of China  
1993

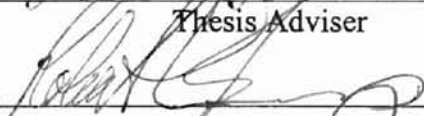
Submitted to the Faculty of the  
Graduate College of the  
Oklahoma State University  
in partial fulfillment of  
the requirements for  
the Degree of  
MASTER OF SCIENCE  
December, 1999

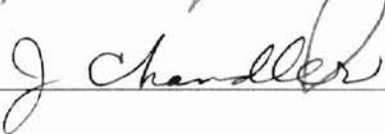
AN ALGORITHMIC PROGRAMMING STRATEGY  
USING THE CLIENT/SERVER MODEL IN  
POLYMERASE CHAIN REACTION  
(PCR) PRIMER DESIGN

Thesis Approved:

  
\_\_\_\_\_

Thesis Adviser

  
\_\_\_\_\_

  
\_\_\_\_\_

Wayne B. Powell  
\_\_\_\_\_

Dean of the Graduate College

## ACKNOWLEDGEMENTS

I have been blessed with such a grand opportunity to further my education in the computer field and do research with financial support. Many people have made the pursuit of my second Master's Degree here a very enjoyable experience. First and foremost, I would like to express my sincere appreciation to my advisor, Dr. George E. Hedrick, for his intelligent guidance and friendship during this study, and for his patience in correcting this thesis. Specially, I would like to thank his encouragement and effort in helping submit this for publication, which added important value regarding my work. I also wish to express gratitude to my committee member, Dr. Robert L. Burnap, who instructed me with detailed requirements, and who also supported me financially in this project. Thanks also go to Dr. John P. Chandler, for his kindness and guidance in many aspects during my graduate study beyond this research. I feel very fortunate to have had such excellent professors for my committee members.

I would also like to express my special thanks to my family and friends, without whose encouragement I would not have been able to complete the requirements for my degree in a timely manner. Thank you, Jun Huan and Bradley Postier, for your precious suggestions and ideas. In particular, I am deeply indebted to my dear husband, Haobo Liu, whose understanding, unconditional love and support have upheld me and let me to BE MYSELF during this study and throughout my life, whenever I am in a high mood or in depression. I will never be able to thank him enough.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION.....	1
II. LITERATURE REVIEW.....	3
2.1. Bioinformatics.....	3
2.2. The Client/Server Model.....	5
2.3. The Internet vs. Intranets .....	7
2.4. Relevant Database Concepts.....	8
2.5. Relevant Biomedical Concepts.....	8
2.5.1. DNA Chips (Micro-arrays).....	9
2.5.2. Genome Projects.....	9
2.5.3. PCR and Primer Design.....	10
III. THE CURRENT SOFTWARE.....	13
3.1. Present Problems.....	13
3.2. The Criteria For The Program Algorithm.....	15
IV. THE ENHANCED SOFTWARE.....	17
4.1. Capabilities .....	17
4.1.1. Serialization .....	18
4.1.2. Flexibility.....	18

Chapter	Page
4.2. System Architecture.....	19
4.2.1. A Brief Description Of The Algorithms .....	19
4.3. Implementation .....	21
4.3.1. Automated Modification.....	22
4.3.1.1. The Standard Input Format .....	23
4.3.1.2. The Contents of Global.txt.....	24
4.3.2. Output Primers Locally And Re-format .....	25
V. SUMMARY AND CONCLUSION.....	28
REFERENCES .....	30
APPENDICES .....	34
APPENDIX A: THE USER’S MANUAL.....	35
APPENDIX B: THE PROGRAMMER’S MANUAL.....	39
APPENDIX C: ACRONYMS AND ABBREVIATIONS .....	46
APPENDIX D: GLOSSARY .....	48

## LIST OF FIGURES

Figure	Page
1. Scope of Bioinformatics Area.....	4
2. A Simplified Client/Server Model .....	6
3. A Simplified Version of Polymerase Chain Reaction – One Cycle .....	12
4. The Enhanced Software System Architecture .....	20
5. Data Flow for the Automated Modification of Files .....	22
6. Local Primer Output .....	26

## LIST OF TABLES

Table	Page
1. Standard Input Format with Tags for Each Gene. ....	24
2. Parameters in File Global.txt .....	24

## CHAPTER I

### INTRODUCTION

The DNA chip (micro-array) technique is one of the most recent technologies used in biomedical research [21, 24]. It permits large amounts of genetic information to be analyzed simultaneously on a small piece of a DNA chip [28, 32]. This is significant when one realizes that biomedical scientist must handle thousands of pieces of gene data simultaneously. The polymerase chain reaction (PCR) is a major tool using this new technique. PCR is a simple, yet powerful tool to analyze the extremely small amounts of DNA from complex mixtures directly [25]. The PCR application of DNA is effected by using primers together with other biochemical considerations. Primer design is essential for the complete process to work properly. Several computer programs to facilitate this process have been written, and are available at university web sites [11, 13]. The main problem with these web-based computer codes is that they process only one gene at a time.

With the increasing use of micro-array technology, thousands of primers must be provided simultaneously for PCR experiments. Obtaining those primers rapidly is the goal of this research. It is driven by the biomedical community's need for rapid production of these primers.



This thesis work presents the resulting enhanced system to meet the goal indicated above. Chapter II of this thesis reviews the topics of bioinformatics, the client/server model, the internet vs intranets, relevant database concepts and relevant biomedical concepts, such as DNA chips (micro-arrays), genome projects and PCR primer design. Chapter III describes the current software with its present problems and the criterions for the program algorithms. Chapter IV focuses on the enhanced software, with its key capabilities, system architecture, and implementation. Finally, chapter V concludes the thesis work and provides some suggestions for future works.

## CHAPTER II

### LITERATURE REVIEW

#### 2.1. Bioinformatics

Bioinformatics, regarded as an intersection between molecular biology and computer science (see Figure 1 on page 4) [31], has attracted increasingly attention in the study of various biological problems by both scientists and software engineers since last decade. Simply stated, bioinformatics refers to the study of biological information involving computer strategies [3]. Those biological problems include the need to implement enormous amount of gene sequence and protein structure data in a database. This data is accumulated from rapid and advanced sequencing methods developed in the fields of molecular biology, genetics, and biochemistry during the past twenty years [5]. Driven by this trend, bioinformatics develops algorithms and computer programs integral to manage and to process the data in an efficient way. Actually, in this new field of bioinformatics, computer scientists and biologists benefit from each other [31]. Biologists need the help of computing to manage the enormous data being generated and convert it into useful information. Software engineers find nature is a great place to apply computational techniques.

Bioinformatics can be divided into three aspects: first, generating algorithms and computer programs specific for biological field; second, implementing biological information and linking with databases; third, designing an artificial molecular computer processor [1,31]. Generally speaking, this field involves the interaction of biology and computer science in four ways: first, gene sequence and protein structure data could be gathered and analyzed efficiently by computer databases; second, nature evolution could be predicted reasonably by computer simulation; third, an interactive data processor could be built usefully just for both biological and computer science; fourth, biology could be applied proficiently and interestingly by computer scientists as a nature source [31].

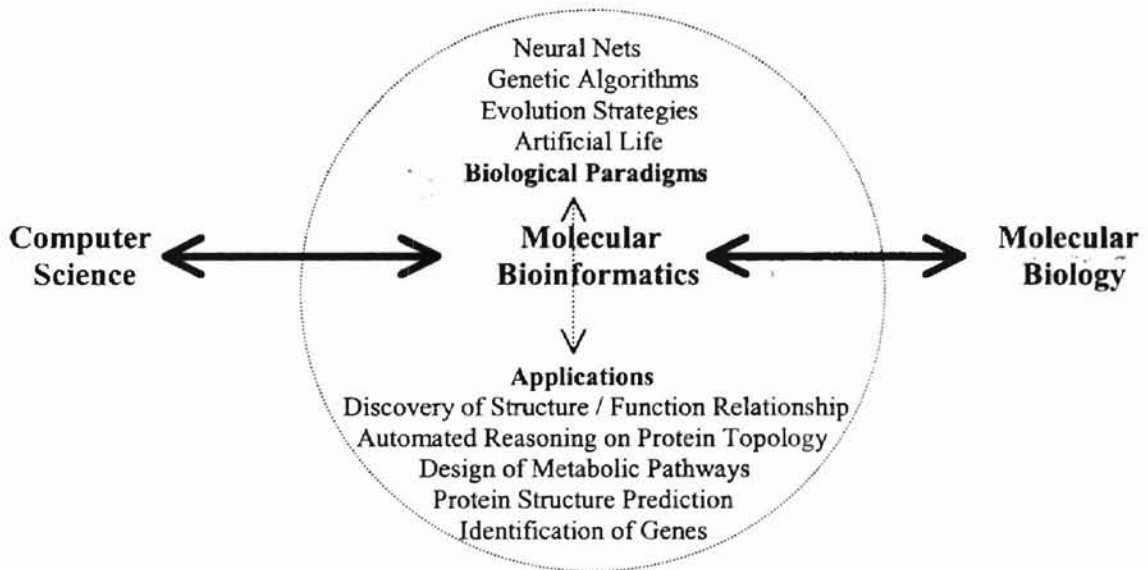


Figure 1. Scope of Bioinformatics Area  
 (modified from Schulzer-Kremer S., ©1996 [31])

In reality, bioinformatics is a new way to do hard, meaningful work as a science with its own rules. The popularity of computers becomes obvious in most molecular

biology laboratories today. For example, a researcher want to know whether a newly determined fragment of DNA sequence encodes a protein or if the sequence is related to a gene that has already been characterized. It will be very tedious and time consuming to look through all of the relevant sequences reported in the literature manually. However, computers can perform such tasks rapidly and correctly. Another good example is the very challenging Human Genome Project [14], which is scheduled to be completed by the year of 2003. This project depends much on the aid of bioinformatics techniques. And the data gathered from this project will benefit further in other gene discoveries and gene-based drug inventions. It may also reveal previously unknown relationships between the structure and function of genes and proteins with new interpretation.

The significance of bioinformatics lies in the availability of biological information freely and publicly, all in an open environment such as the internet or intranets. This affects almost all of the aspects in biological research area, including gathering, storing and analyzing raw data, plus generating new methods to examine those data. This will speed the biological research step and also avoid repetitious activity [17].

## 2.2. The Client/Server Model

The client/server model is a very popular and efficient model applied in various businesses nowadays, although it was introduced to the software industry only a few short years ago. This model divides a computer application into three basic components - a client, a server, and a network that connects the client with the server [20]. Generally speaking, a client refers to the computer with which the user interacts and makes a request of a server for a particular resource. A server refers to the computer that accepts

requests from one or more clients and sends back only the information the client requests (see Figure 2 on page 6) [30]. The client and the server actually communicate with one another through a network connection between them. The conversation is started by the client, and ended by the server.

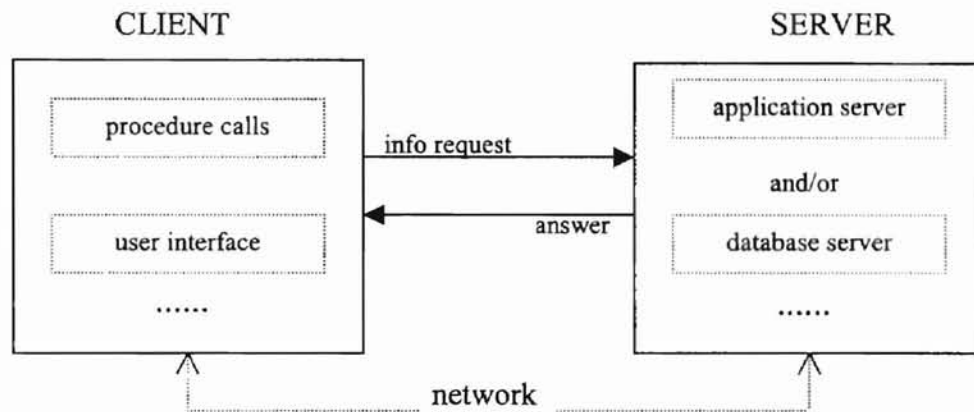


Figure 2. A Simplified Client/Server Model  
(modified from Schneberger, S.L., ©1997 [30])

In a client-server approach, a server not only has data files but also contains some application software. That software may be one to several independent programs, or it adapts some universal database servers such as Microsoft SQL Server or Oracle [20]. The server basically includes file servers, application servers, database servers, and web servers. It may be an advanced PC, a minicomputer or a mainframe [20]. The major function of a server is to manage the physical storage of the data and also response to the requests from clients manipulated by its own application software [30]. On the other hand, a client is the end-user processor of the servers, calling on the server for services mentioned above [30]. The client is a personal computer sitting on the user's desk in most cases.

The idea of client/server is to do all the database processing on the server, using all the available sources like disk, memory, processor, and operating system [20] from the server. The client and the server operate independently of each other. For example, when a client needs information contained in a server's files, the client sends a message to the server. The server processes those data and provides the needed answer – but not the entire files – to the client at a later time.

This client/server model has lots of advantages. First, file security is easier to maintain with the server in full control of file data, and data integrity is also easier to keep since only the server actually updates files [30]. Moreover, the network is not overloaded with entire files being transferred back and forth between processors. The server software runs on its own machine to get the answers for the client's request. The client doesn't have to deal with any aspect of database processing [30].

### 2.3. The Internet vs. Intranets

The internet, known well as the World Wide Web or WWW, frequently uses the client/server model. Information can be retrieved specifically for particular end-users from the web. It is a convenient way to communicate with people at a distance [30], but it is also notorious for its possibility of data inaccuracies and inappropriate manipulation by some ill-intentioned people. For this reason, intranets are used as alternative pathways especially for scientific research groups. An intranet, regarded as an internal internet [30], has the same characteristics as easy-access like web sites though with a more simplified format, and it is protected more restrictively. Intranets allow end-users

to run applications inside their familiar environments. The intranet is the new client/server model of choice [20].

From certain views, an intranet could ease client-server software maintenance problem for confidential data access and usage [30]. Having an intranet is of particular value within certain organization such as a scientific research group. Intranets also are protected, only the members in the same organization can access the internal web pages. Others are prohibited by certain restrictions, such as password requirements.

#### 2.4. Relevant Database Concepts

Database is key of any current client/server application [20]. A database consists of some collection of persistent data that is used by application systems of some given enterprise. A database system involves in four major components [16]: (1) Data (information), (2) Hardware, (3) Software: between the physical database and the user is a layer of software, called database management system (DBMS), and (4) User. DBMS is specialized software programs that store, retrieve, and manipulate data [16]. In the client/server model, database servers run DBMSs to implement those data sent from client sides. This kind of server is the dominant model for creating client/server applications currently, so as being very powerful in today's data manipulation of almost every field.

#### 2.5. Relevant Biomedical Concepts

Three relevant biomedical concepts: DNA chips, or micro-arrays; genome projects;

and polymerase chain reaction (PCR) underlying the development of the software for this project are described below.

#### 2.5.1. DNA Chips (Micro-arrays)

A new biomedical technique is expressing the profiles of different related genes simultaneously using micro-arrays [21, 24]. Similar to parallel processing, large amounts of gene data are expressed, then analyzed simultaneously from a small piece of a DNA chip on a glass slide. This takes place after a sequence of biochemical reactions including PCR amplification. Although knowing where, when, and how a gene is expressed in nature provides strong clues to its biological role; understanding the relationship of a gene among other genes in a real environment provides additional valuable information in a comparative study. The use of DNA chips can result in a high throughput of gene expressions and resulting discoveries since thousands of individual genes could be put on several chips together in a high density array arrangement [28, 32]. This technique has been applied to gene expressions in plant [28], yeast [32], and human [29] samples. All demonstrate the significance of DNA chips since the biomedical analyses can be completed in less space and in less time than previously, but they still provide more detailed gene output. This especially is meaningful for projects on a genome scale.

#### 2.5.2. Genome Projects

The term genome refers to all of the genetic material in an organism. The genetic material is represented by linear sequences of combinations of the four chemical bases



adenine (A), cytosine (C), guanine (G), and thymine (T) [15]. Genome projects refer to research that emphasize gene analysis on a genome basis; for example, mapping and sequencing the entire genome of an organism. The complete sequences for about a dozen microbial genomes. The community is anticipating the successful exploration of several metazoan genomic sequences in the near future [15]. This could include the human genome project that is scheduled for completion in 2003 [14]. These genome projects help us understand complete organisms and can influence further research in the life sciences.

“Functional Genomics of Plant Stress Tolerances” is the plant genome project that needs this computer application [23]. In one sense, it is similar to the human genome project. The purpose of the project is to analyze the roles of all genes of representative plants in response to various stresses, especially stresses created by different water and ion concentrations. The anticipated research results from the complete project should give references for increasing the productivity of crop plants. Model organisms include “*Synechocystis* PCC6803, *Saccharomyces cerevisiae*, *Aspergillus nidulans*, and *Arabidopsis thaliana* [23].” All of these are well-studied models. *Synechocystis* has been sequenced completely for the entire genome; it has a length of about 3.6 megabases for about 4,000 genes [12].

### 2.5.3. PCR and Primer Design

The polymerase chain reaction, or PCR, is an *in vitro* biological technique that allows the amplification of a specific deoxyribonucleic acid (DNA) fragment that lies between two regions of a known DNA sequence [2, 4, 34]. Each DNA molecule consists

of two strands of nucleic acid, called a double helix structure, with a sugar-phosphate backbone attached by bases A, C, T, G. The two strands of this helix are held together by hydrogen bonds. Those hydrogen bonds form between the complementary bases A and T by two hydrogen bonds and G and C by three hydrogen bonds [22].

PCR amplification of DNA is effected by using primers and other required reaction conditions such as heat cycling parameters, a DNA polymerase and individual bases of the DNA molecule, dNTPs. These primers refer to short, single-stranded DNA fragments that are complementary to both ends of certain DNA region that is ready to be amplified. They usually are about 20-30 bases in length [22]. There are three steps in the PCR process: first, separating the double-stranded DNA (denaturation); second, annealing the primers (annealing); third, extending the primers by DNA polymerase with dNTPs (extension) [22]. This process could be repeated  $n$  cycles to get approximately  $2^n$  more DNA molecules over the original numbers. That is, each successive cycle leads to a doubling of the amount of DNA synthesized in the previous cycle. Thus the target DNA fragments are amplified at least a million folds just after 20-25 rounds [22] (see Figure 3 on page 12).

The PCR technique makes the direct analysis of small amounts of DNA from complex mixtures possible. It has been applied in almost every field of biological, medical and forensic sciences. Primers play a major role in the entire PCR procedure. Most rules for primer design are empirical and must be adjusted individually. However, careful follow-up to these rules will lead to a successful PCR result with more guarantee [22]. Primer design relies on accurate assessment of standard biochemical parameters (details please see Chapter III); they are empirical and require individual adjustment.

Although primer design can be performed manually, optimizing all the basic parameters of selecting primer pairs is tedious, error prone work.

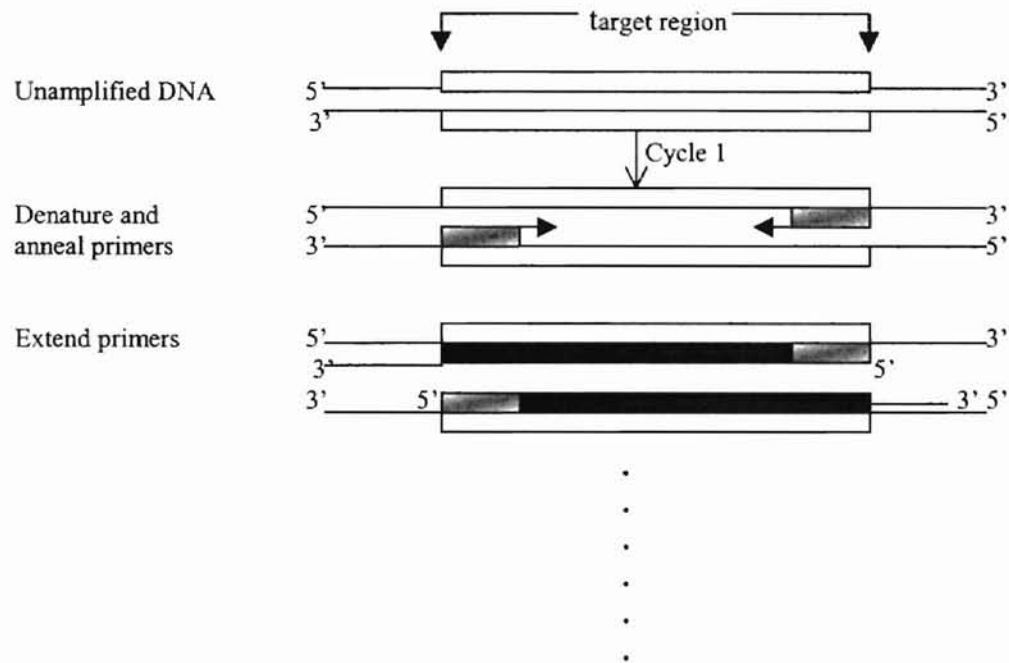


Figure 3. A Simplified Version of Polymerase Chain Reaction – One Cycle  
( modified from Newton & Graham, ©1997 [22] )

Several computer programs to facilitate the analysis are available publicly at web sites. One excellent site is at MIT:

[http://www.genome.wi.mit.edu/cgi-bin/primer/primer3\\_www.cgi](http://www.genome.wi.mit.edu/cgi-bin/primer/primer3_www.cgi) [13];

Another is at Stanford:

<http://genome-www2.stanford.edu/cgi-bin/SGD/web-primer> [11]. Both of those programs allow

users to paste together one gene sequence with its parameters, such as its melting point (  $T_m$  ), GC percent, and the lengths of the primers in order to obtain the optimal choices from a remote server at a later time.

## CHAPTER III

### THE CURRENT SOFTWARE

#### 3.1. Present Problems

As discussed in the previous chapters, primer design is essential for the entire PCR procedure. It also is a key step in genome projects and research using DNA chips. Although primer design can be determined manually, optimizing all the basic parameters of selecting primer pairs is tedious, error prone work.

The web-based programs from MIT and Stanford are a giant step in primer design compared to the manual techniques that were used previously. Analyzing data is still time consuming when many gene sequences must be processed simultaneously. Even when scientists want to adopt the latest micro-array technique for genome wide exploration of gene expression patterns, this is a problem. All web interfaces currently available allow only one sequence of gene data, with its relevant parameters, to be input for each run. The information needed for a single gene must be put on the web one piece at a time. A practical example is a prokaryote plant called *synechocystis* PCC6803. It has about 4K clarified genes and a genome length of 3.6 Mb [12]. Using today's web sources to design the primers, and assuming it requires five minutes to get the results for one gene (an extremely optimistic estimate), the entire work would require fourteen days. Of course, this also assumes that no error occurs in the manual processing.

Another practical problem is that some gene sequences are too short (less than 300 bases) to make primers from the programs available from the web. Also, some are too large (over 1,000 bases) to obtain optimal primers when gene fragments are chosen to be located near the beginning of the gene. This is due to the experimental fact that incomplete transcripts occur near the beginning of a gene ultimately resulting in an over estimation of the gene expression after hybridization with these incomplete transcripts. It is best to let the user decide the end points of the gene coding range from the genome sequence rather than using an individual gene basis. Usually, there are at least 150 bases away from both the upstream (5' end) and the downstream (3' end) of a gene; These bases do not overlap other genes. If a gene is too short to obtain the primers, then the program should have the ability to extend the length from either end of the gene automatically to make it possible to obtain the primers. If a gene is more than 1,000 bases long, then it is best to synthesize primers than amplify the last ½ of the gene toward the downstream end. This is not feasible with any of the currently available web programs.

Thus, a way to run all the gene data simultaneously and sequentially, with flexible input parameter choices, comes to be an important issue in primer design. My objective in this research focuses on this functional enhancement of computing for PCR primer design, in order to be used by our local OSU bioinformatics research group and benefit the researchers.

### 3.2. The Criteria For The Program Algorithm

In general, the primers used in PCR are between 20 and 30 nucleotides (A nucleotide refers to a molecule of a base from “G, C, A, T” together with its phosphate and a sugar; it’s the basic structure of nucleic acids.) in length. The number of each of the four bases should be approximately equal in a primer. The following criteria are the guidelines that should be followed in the program written for PCR primer design [25]:

1. Unique 20-30 nucleotides in length, special for the target amplified DNA sequence.
2. “G.C clamp” at the 3’ ends of primers. This means avoiding runs three or more C’s or G’s at the 3’ ends.
3. Avoid self-complementary sequences (palindromes) within individual primers, particularly at the 3’ ends of primers.
4. No complementarity between each primer pairs. This also stresses the 3’ ends of the pair.
5. G/C content around 45-55% in base composition. The distribution of all four bases should be arranged randomly.
6. Avoid T’s in the 3’ ends of primers.
7. Match primer melting point (  $T_m$  ) between 55°C and 80 °C, following the general formula as  $T_m = 81.8 + 16.6( \log_{10}([Na^+]) ) + 0.41*( \%GC ) - 600/\text{length}$  [27].

In details, the first criterion for primer design is to choose primers that are unique to the region need to be amplified [22]. The most important region to check is at the 3’ end of a primer as this is where synthesis of PCR product begins as “primer extension”. This also applies to the following criteria. Based on experimental results, 20-30

nucleotides are adequate for a primer to take effects. The second parameter to be considered is the careful inclusion of a G/C residue (“G.C clamp”) at the 3’ end of the primer [22]. Only 1-2 G/C nucleotides at this end could avoid mispriming in G.C-rich regions. This is because of the stronger hydrogen bonding (three hydrogen bonds) generated by G/C base pair. A/T base pair only has two hydrogen bonds between themselves. Third, each individual primer should also be designed with no self-complementary structure (palindromes) [22], especially at the 3’ ends. Such structure can lead to “hairpin” formation (another name in biochemistry for palindrome structure within a single primer) and could inactivate the primer to combine with the template DNA strand effectively. Fourth, this is also true in designing a pair of primers for the PCR reaction [22]. Complementarity between the primer pair, especially at the 3’ ends could cause the formation of “primer-dimer” artifact (that is, palindrome structure within a primer pair) and reduce the yield of the desired PCR product. The fifth point is to control the G/C content in a primer composition between 45-55%, to maximize specific binding but also allow efficient melting during the PCR, with a general random distribution for all four bases [22]. Sixth, since T is the least discriminating nucleotide in the four bases, avoiding the presence of a T at the 3’ end will reduce the possibility of mismatches [22]. At last, it is also important to let primer pairs have similar melting temperatures (  $T_m$  ) to promote simultaneous annealing process at a given temperature (55-80°C) [22]. This temperature is determined by

$$T_m = 81.8 + 16.6( \log_{10}([Na^+]) ) + 0.41*( \%GC ) - 600/\text{length} [27].$$

## CHAPTER IV

### THE ENHANCED SOFTWARE

This chapter illustrates the capabilities, system architecture and implementation issues in developing the overall software. The source code available from the Whitehead Institute for biomedical research at MIT [27] is re-used, then modified. The resulting software allows users to access gene sequences and primers on a genome scale. All the primers can be synthesized automatically and sequentially. For example, running the new program takes approximately one minute on the Sun Enterprise 3000 under Solaris and yields almost all the primers for the 4,000 genes of the cyanobacterial genome, *Synechocystis* sp. PCC6803. This is a time saving of several orders of magnitude, compared to previous web processing. Correspondingly, the scientists using the latest DNA chip technique in their research can save both time and money.

#### 4.1. Capabilities

The key capabilities of this software are gene serialization and flexibility, based on the client/server model.



#### 4.1.1. Serialization

This is a major achievement embodied in the new software. As long as the complete genome data for one organism with each gene's general information: e.g., its open-reading-frame (ORF) name, the left and right boundary points of the gene in the genome, and the direction of the gene in the genome (direct or complement), are available, then individual gene sequences can be arranged sequentially to fit the required format of the MIT programs. Next, the batch sequence file is run automatically, without stopping, to output the primers to the users within minutes, regardless of the number of genes in a genome.

#### 4.1.2. Flexibility

Another benefit of the software arising from this research is the allowing of users to decide their preferred biochemical parameters. For example, amplified gene fragments could be restricted to a certain range for any or all of the genes. This is crucial in obtaining a balanced expression pattern of different genes on a DNA micro-array. The new program allows users to set up the left and right cut-off coordinates whenever necessary. For instance, when the program cannot obtain the primers for certain genes from its initial run, it allows users to extend either the upstream (5' end) and/or downstream (3' end) of sequences in the whole genome to certain length (number of bases) from the second and/or third run(s). Running the program again should yield acceptable primers in most instances provided that the overlapped region of the new sequence with the original gene is at least 200 bases. Finally, the program allows users to

pick up bases from the downstream (3' end) of the sequences when the gene exceeds 1,000 total bases. This is appropriate for experimental performance.

## 4.2. System Architecture

The diagram in figure 4 on page 20 shows the program architecture for the new PCR primer design system. Raw sequence data together with genes' general information are saved locally using FTP or a similar protocol. This data is stored in flat ASCII encoded files in directories set aside for use by these programs prior to the programs actually performing their data analysis. Automatic modification of the data in those files assures both data integrity and proper input format. To use the new system, users need only to enter the genome name, then the server does the remainder of the processing without further intervention. The users can set up the global parameters for making parameters, but it is not necessary. If the user elects not to specify global parameters, then the program uses default values stored on disk. The server runs the codes in conjunction with the MIT programs, and it stores the output in a local database, such as Microsoft Access [18].

### 4.2.1. A Brief Description Of The Algorithms

The parameters required for making primers corresponding to thousands of gene sequences from the genome of an organism used for DNA micro-arrays are used as global parameters during program execution in order to have a consistent, easy to use system. These parameters are stored on the local server in a file named Global.txt.

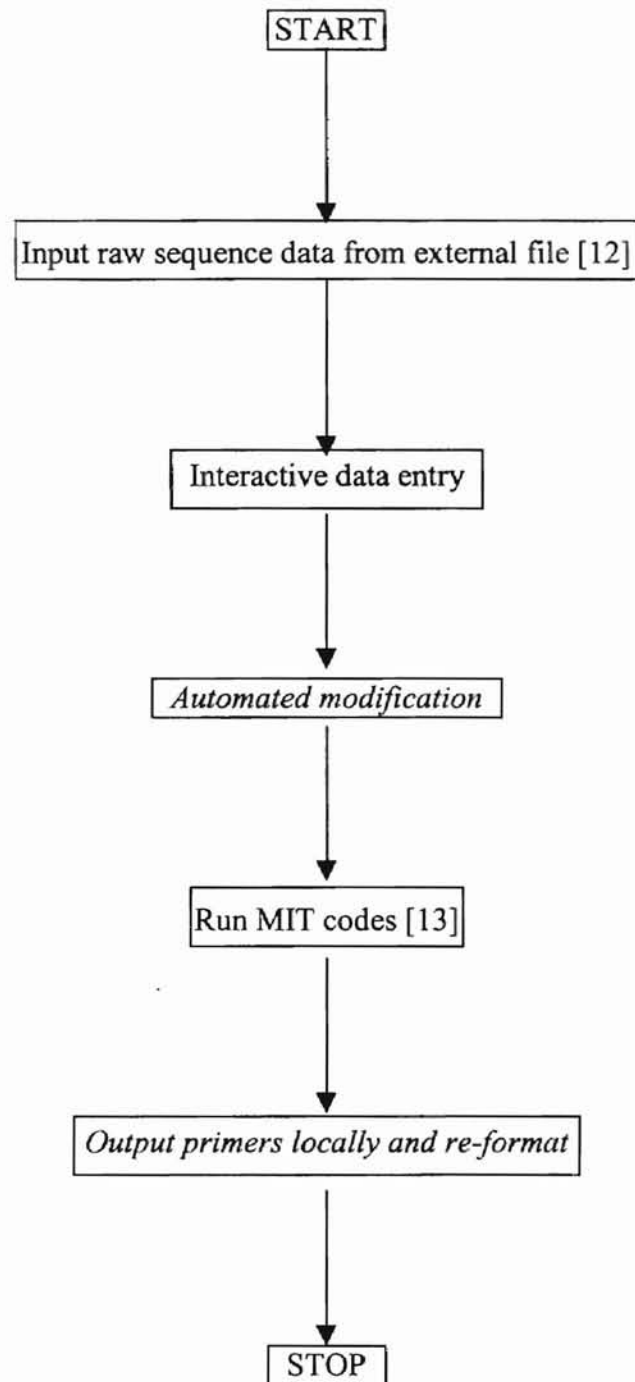


Figure 4. The Enhanced Software System Architecture (italics indicate components described in this thesis.)

A high level description of the new system follows.

**Step 1** Put the data for the genome into a single, one dimensional array.

Genome[GENOMESIZE]

**Step 2** Input the ORF gene names. Search to find the trace for the left and right cutting points as well as the direction for a specific gene.

gene(geneName, left, right, direction)

**Step 3** Cut the sequence from the genome and add a tag. After the initial run, the left and right coordinates extend outward.

**Step 4** Repeat steps 2 and 3 until all inputs have been processed.

**Step 5** Append Global.txt to the input file.

**Step 6** Run the MIT programs using the newly computed data as input.

**Step 7** Output re-formatted results to a local database, such as Microsoft Access.

#### 4.3. Implementation

The programs implementing in this research primarily are written by C. There are a few exceptions of UNIX shell programming and HTML specifications. Currently these programs run on UNIX or LINUX based machines. The development environment is a Sun Enterprise 3000 running Solaris 7.

The entire process is guided by a "Makefile". Once users are in the proper directory for this project, they can type the command **make** at the shell prompt. The system executes each step as needed, then it outputs the optimal primer results in the three files, submit.txt, submit2.txt, and submit3.txt.

An explanation of the components indicated by italics in figure 4 on page 20 follows.

#### 4.3.1. Automated Modification

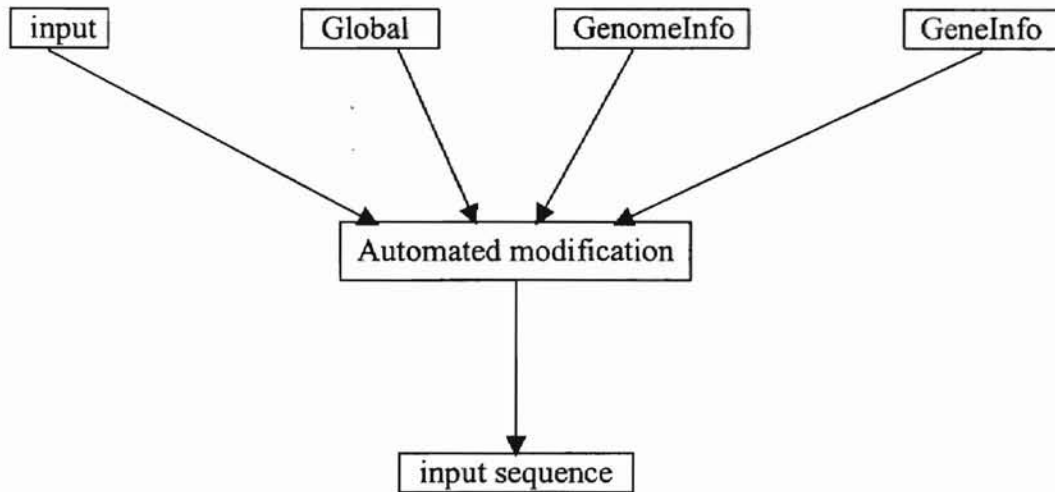


Figure 5. Data Flow for the Automated Modification of Files  
(all files have an extension of “.txt”.)

Figure 5 contains an expanded diagram of the data flow for the automated modification process. Four input files are required. The first is a file that contains the sequence data for the entire genome. This file is called GenomeInfo.txt. The second file stores information for each gene, and it is called GeneInfo.txt. GeneInfo.txt includes a specification of the ORF name, the left and right boundary points, and the direction (direct or complement) for each gene. The third file is Global.txt that was described earlier in this paper. A fourth file, input.txt may be used. If input.txt is not used, then the keyboard (standard input unit) is used as a default. The users can input an ORF name for

each gene for which they wish to obtain optimal primers. Use of keyboard input with immediate screen echoing and error messages provides an interactive way to set ORF names. The most common errors are use of a non-existent ORF name or missing information in the GeneInfo.txt file.

The first step when the program executes is to read the entire GenomeInfo.txt file into a single, one-dimensional array, Genome. Next, information from GeneInfo.txt is paired with gene sequences specified in input.txt. The corresponding information from the Genome array is written to the inputSequence.txt file. Special requirements for the chosen gene fragments must be given in the input.txt file. Special requirements might include items such as, maximum gene fragment length (in bases), starting point for the fragment. An example of the maximum gene fragment length would be "800" bases. In this case anything longer than 800 bases is disregarded. An example of the starting point for a gene fragment is "the 800<sup>th</sup>" point towards the 3' end of this gene. The two parameters, total gene fragment length and starting point, can assume default values or they can be set manually. Tags for distinguishing individual genes are placed into the inputSequence.txt file to fit the input format required by the MIT codes. Finally, the contents of the file Global.txt is appended to the inputSequence file to specify requirements for making specific primers.

#### 4.3.1.1. The Standard Input Format

Table 1 on page 24 shows the input format to specify one gene.

SEQUENCE\_ID = {the gene's ORF name}

SEQUENCE = {the gene fragment}

= {used to distinguish gene fragments}

Table 1. Standard Input Format with Tags for Each Gene  
(the identifiers on the left serve as prompts.)

#### 4.3.1.2. The Contents Of Global.txt

Table 2 on page 24 shows the parameters in the Global.txt file. The default values are shown in the table. All of these values are subject to a manual override.

Parameter name	Default value
PRIMER_PRODUCT_SIZE_RANGE	300-800
PRIMER_OPT_SIZE	24
PRIMER_MIN_SIZE	24
PRIMER_MAX_SIZE	24
PRIMER_OPT_TM	60.0
PRIMER_MIN_TM	58.0
PRIMER_MAX_TM	62.0
PRIMER_OPT_GC_PERCENT	50.0
PRIMER_MIN_GC	30.0
PRIMER_MAX_GC	80.0
PRIMER_SALT_CONC	50.0
PRIMER_SELF_ANY	8.00
PRIMER_SELF_END	3.00
PRIMER_NUM_RETURN	1

Table 2. Parameters in File Global.txt

Whenever optimal primers cannot be obtained from the initial run, it is possible and desirable to repeat the above processes for a second and a third run. The differences among these runs are the left and/or right end points are extended outward by a selected length from the normal gene location. The default extension length is 150 bases.

#### 4.3.2. Output Primers Locally And Re-format

Since the MIT programs produce results in addition to what we need, we re-format the output to meet our requirements. We first save the output in a file named Out.txt. Next, we filter the results into a file named filter.txt. The filter.txt file might contain such things as the gene's ORF name, primer pairs, and primer product size at the user's option. We then separate genes that have primer output from those that have no primer output. The file noError.txt contains the results for those genes with primer output; the file error.txt contains the names of genes for which there were no primer outputs. The error.txt file can be used as input for subsequent runs to obtain primers for those genes. Finally, we re-format the contents of the file noError.txt and place the result of re-formatting into a file submit.txt to be input into a local database. This process is shown in figure 6 on page 26.



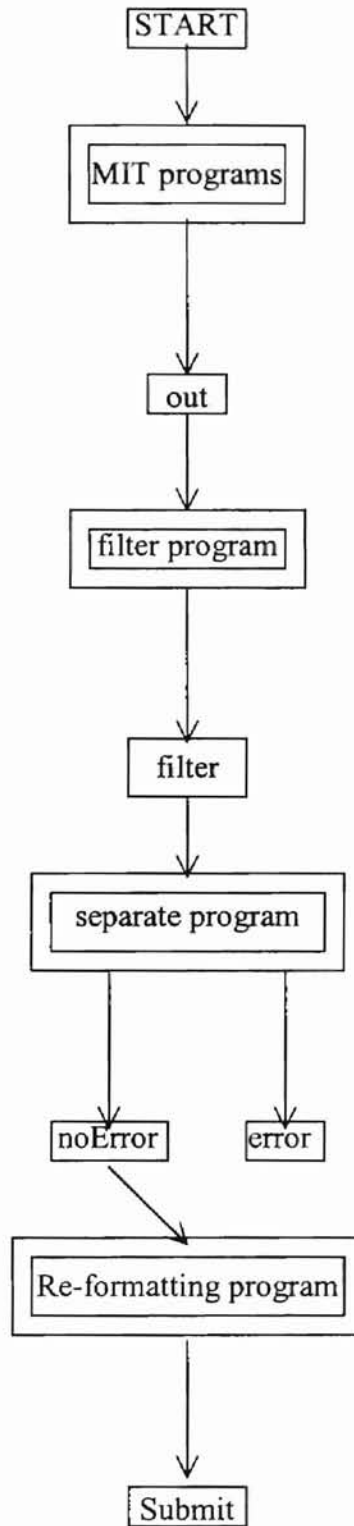


Figure 6. Local Primer Output  
(all files have an extension of ".txt".)

Generally, the program is implemented by using the client/server model:

Functions for the client:

1. Set up an interface to pass batch of gene sequence data from genome sequentially at the same time.
2. Pass the crucial biochemical parameters for primer design with those sequence data on the same interface, such as the melting point.
3. Let users to define some special cutting points of a gene from the genome.

Functions for the server:

1. Run batch of gene sequence data automatically without stopping.
2. Output the optimal primers by those raw sequence data and biochemical parameters
3. Repeat the entire process if no optimal primers were obtained from the previous run automatically, until acceptable primers are made. Maximum run is three times.
4. Transfer those optimal primer choices back to end-users in real time.

## CHPATER V

### SUMMARY AND CONCLUSIONS

In this research, the author designs and implements an algorithmic programming strategy using the client/server model applied in PCR primer design. C language is used to develop this strategy. This project presents the possibility of developing new algorithms when used in conjunction with existing algorithms extend their power from producing a single primer pair to producing thousands of them. The use of the new system incorporating (re-using) old and new software allows the handling of massive amounts of gene sequence data for primer design.

The resulting software allows users to access gene sequences on a genome scale and obtain optimal primers in real time. All the primers can be synthesized automatically and sequentially. End-users also have more flexibility to input the parameters for primer design they need in order to fit various experimental conditions. The time for the entire process is decreased dramatically to being minutes, compared to work that previously took months. This favors researchers who need to handle lots of gene sequence data at a given time, especially in experiments using the latest DNA chips technique.

Future improvement and research can be directed at connecting those primer outputs with a database server for better visualization. Also the source code can be

modified to become a web-based program, in order to benefit the scientists on a world-wide basis to manipulate DNA information more efficiently.

## REFERENCES

- [1] Alphey, L. What is DNA sequencing? In DNA Sequencing: From Experimental Methods to Bioinformatics. New York: Springer, ©1997.
- [2] Bankier, A. T., K. M. Weston, and B. G. Barrell. Random cloning and sequencing by the M13/dideoxynucleotide chain termination method. *Methods in Enzymology*, v. 155, pp. 51-93, 1987.
- [3] Baxevanis, A. D. The internet and the biologist. In: *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. Edited by A. D. Baxevanis and B.F.F. Ouellette. New York: Wiley-Interscience, ©1998.
- [4] Berg, C. M., G. Wang, L. D. Strasbaugh, and D. E. Berg. Transposition facilitated sequencing of DNAs cloned in plasmids. *Methods in Enzymology*, v. 161, p. 218, 1993.
- [5] Butler, B.A. Sequencing analysis using GCG. In *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. Edited by A. D. Baxevanis and B. F. F. Ouellette. New York: Wiley-Interscience, ©1998.
- [6] Crandall, B.C. and J. Lewis (Eds.). *Nanotechnology: Research and Perspectives*. New York: Wiley-Interscience, ©1992.
- [8] Graham, I. S. *The HTML Source Book: a Complete Guide to HTML 3.0*. New York: John Wiley and Sons, Inc., ©1996.

- [9] Gundavaram, S. CGI Programming on the World Wide Web. Sebastopol, CA: O'Reilly & Associates, Inc., ©1996.
- [10] Hagey, J. Programming Perl 5.0: CGI Web Pages for Microsoft Windows NT. Emeryville, CA: ZD Press, ©1996.
- [11] <http://genome-www2.stanford.edu/cgi-bin/SGD/web-primer>.
- [12] <http://www.kazusa.or.jp>.
- [13] <http://www.genome.wi.edu/cgi-bin/primer/primers.cgi>.
- [14] [http://www.oml.gov/TechResources/Human\\_Genome/hg5yp/](http://www.oml.gov/TechResources/Human_Genome/hg5yp/).
- [15] <http://www.oml.gov/hgmis/publicat/glossary.html>.
- [16] Henry, F. K. and A. Silberschatz. Database System Concepts. New York: McGraw-Hill, ©1991.
- [17] Krawczak, M. and D. N. Cooper. The human gene mutation database. Trends in Genetics, v. 13, pp. 121-122, ©1997.
- [18] Kruglinski, D. J. Inside Visual C++. Redmond, WA: Microsoft Press, ©1997.
- [19] Lemay, L. and R. Cadenhead. SAMS Teach Yourself Java 2 Platform in 21 Days. Indianapolis, IN, ©1999.
- [20] Linthicum, D. David Linthicum's Guide to Client/server and Intranet Development. New York: John Wiley & Sons, Inc., ©1997.
- [21] Marshall, A. and J. Hodgson. DNA chips: an array of possibilities. Nature Biotechnology, v. 16, pp. 27-32, January 1998.
- [22] Newton, C.R., and A. Graham. PCR. New York: Springer-Verlag, Inc., ©1997.

- [23] Project proposal of OSU bioinformatics research group. Functional genomics of plant stress tolerance. NSF grant 98-13360.
- [24] Ramsay, G. DNA Chips: state-of-the art. *Nature Biotechnology*, v. 16, pp. 40-44, January, 1998.
- [25] Rickwood, D. PCR: Essential Techniques. Chichester, England: John Wiley & Sons, ©1996.
- [26] Robinson, B.H. and N. C. Seeman. The design of a biochip: a self-assembling molecular-scale memory device. *Protein Engineering*, v. 1, no. 4, pp. 295-300, 1987.
- [27] Rozen S. and H. Skaletsky.  
[http://www-genome.wi.mit.edu/genome\\_software/other/primer3.html](http://www-genome.wi.mit.edu/genome_software/other/primer3.html).
- [28] Schena, M., D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA micro-array. *Science*, v. 270, pp. 467-531, 1995.
- [29] Schena, M., D. Shalon, R. Hellen, A. Chai, P. O. Brown, and R. O. Davis. Parallel human genome analysis: micro-array based expression monitoring of 1,000 genes. *Proceedings of the USA National Academy of Sciences*, v. 93, pp. 10614-10619, 1996.
- [30] Schneberger, S. L. *Clinet/Server Software Maintenance*. New York: McGraw-Hill, ©1997.
- [31] Schulze-Kremer, S. *Molecular Bioinformatics: Algorithms and Applications*. Berlin: Walter de Gruyter, ©1996.

- [32] Shalon, D., J. S. Smith, and P. O. Brown. A DNA micro-array system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Research*, v. 6, pp. 639-645, 1996.
- [33] Stevens, W. R. *UNIX Network Programming*. Saddle River, NJ: Prentice Hall, Inc, ©1990.
- [34] Sulston, J., Z. Du, K. Thomas, R. Wilson, et al. The *C. Elegans* genome sequencing project: a beginning. *Nature*, v. 356, pp. 37-41, 1992.
- [35] Witherspoon, Cr. and Co. *Witherspoon. LINUX for Dummies*. Foster City, CA, ©1997.



## APPENDICES

APPENDIX A  
THE USER'S MANUAL

## The User's Manual

This enhanced software could be used for rapid production of PCR primers at a given time. It is particularly convenient and economical in experiments involving the latest DNA micro-array technology.

First of all, the end-users must make sure those three files are available through “vi” command on a Unix/Linux platform. One file contains the information about the entire genome sequence in an organism, named GenomeInfo.txt. Another file is about the information of each gene, as ORF name, left/right end points, and direction in the genome, called GeneInfo.txt. The third one is a Global.txt file which contains the required parameters for primer design.

The format of GenomeInfo.txt is:

ATCGCGATATGGCTTT... {the gene bases; no return symbol within those bases.}

The format of GeneInfo.txt is:

ORF name	left	right	direction
slr1327	3,000	6,000	complement
sll1439	200	14,000	direct
.....			

The format of Global.txt is:

Parameter name	Default value
PRIMER_PRODUCT_SIZE_RANGE	300-800
PRIMER_OPT_SIZE	24
PRIMER_MIN_SIZE	24
PRIMER_MAX_SIZE	24
PRIMER_OPT_TM	60.0
PRIMER_MIN_TM	58.0
PRIMER_MAX_TM	62.0
PRIMER_OPT_GC_PERCENT	50.0
PRIMER_MIN_GC	30.0
PRIMER_MAX_GC	80.0
PRIMER_SALT_CONC	50.0
PRIMER_SELF_ANY	8.00
PRIMER_SELF_END	3.00
PRIMER_NUM_RETURN	1

Then, the actual manipulation towards the end-users side is very simple. On the computer screen of a Unix/Linux platform, the user only need to type the command word: **make**. Then hit the Return key. The system follows this command, executes each compiling step according to its priority order in a file named Makefile, and output the final optimal primer results saved in three files called submit1.txt (the initial run), submit2 (the second run), and submit3.txt (the third run). All those generated files are saved under the same directory the user implements the command.

To retrieve the results of those optimal primers, the user could view them through the commands: “view submit1.txt”, “view submit2.txt”, and “view submit3.txt” under

the same Unix/Linux platform. Or more conveniently, the user could transfer those outputs and save them in a local drive (C drive) using FTP transmission.

The third way is to transfer those outputs in a local database server, such as Microsoft Access, for better visualization at a later time.

APPENDIX B  
THE PROGRAMMER'S MANUAL

## The Programmer's Manual

The programs implemented in this research primarily are written in C. There are a few exceptions where UNIX shell programming and HTML specifications are urged. Currently these programs run on UNIX or LINUX based machines. The development environment is a Sun Enterprise 3000 running Solaris 7.

The entire process is guided by a "Makefile". Once users are in the proper directory for this project, they can type the command **make** at the shell prompt. The system executes each step as needed, then it outputs the optimal primer results in the three files, *submit.txt*, *submit2.txt*, and *submit3.txt*.

An explanation of the components indicated by italics in figure 4 on page 20 follows.

\* Automated Modification

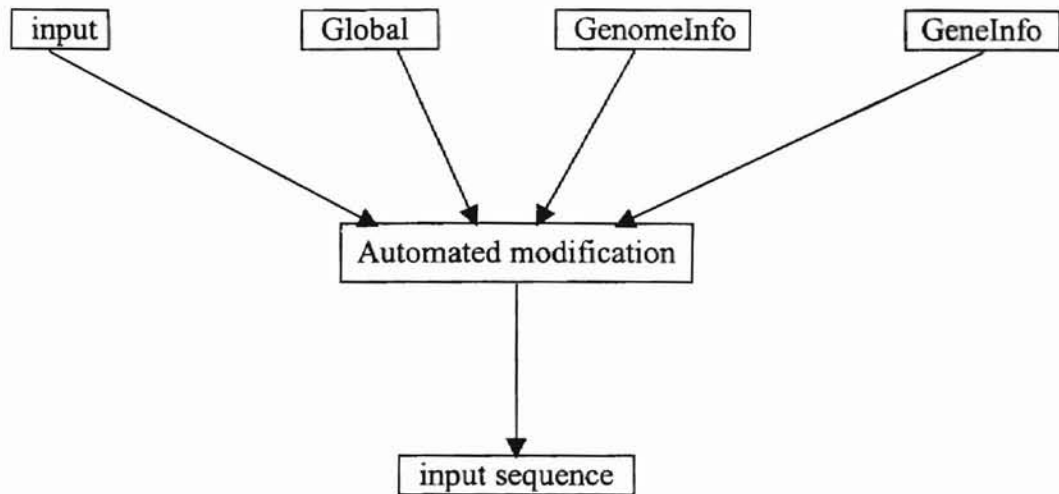


Figure 5. Data Flow for the Automated Modification of Files  
(all files have an extension of “.txt”.)

Figure 5 contains an expanded diagram of the data flow for the automated modification process. Four input files are required. The first is a file that contains the sequence data for the entire genome. This file is called GenomeInfo.txt. The second file stores information for each gene, and it is called GeneInfo.txt. GeneInfo.txt includes a specification of the ORF name, the left and right boundary points, and the direction (direct or complement) for each gene. The third file is Global.txt that was described earlier in this paper. A fourth file, input.txt may be used. If input.txt is not used, then the keyboard (standard input unit) is used as a default. The users can input an ORF name for each gene for which they wish to obtain optimal primers. Use of keyboard input with immediate screen echoing and error messages provides an interactive way to set ORF



names. The most common errors are use of a non-existent ORF name or missing information in the GeneInfo.txt file.

The first step when the program executes is to read the entire GenomeInfo.txt file into a single, one-dimensional array, Genome. Next, information from GeneInfo.txt is paired with gene sequences specified in input.txt. The corresponding information from the Genome array is written to the inputSequence.txt file. Special requirements for the chosen gene fragments must be given in the input.txt file. Special requirements might include items such as, maximum gene fragment length (in bases), starting point for the fragment. An example of the maximum gene fragment length would be "800" bases. In this case anything longer than 800 bases is disregarded. An example of the starting point for a gene fragment is "the 500<sup>th</sup>" point of the 5' end of this gene. The two parameters, total gene fragment length and starting point, can assume default values or they can be set manually. Tags for distinguishing individual genes are placed into the inputSequence.txt file to fit the input format required by the MIT codes. Finally, the contents of the file Global.txt is appended to the inputSequence file to specify requirements for making specific primers.

\* The Standard Input Format

Table 1 on page 43 shows the input format to specify one gene.

SEQUENCE\_ID = {the gene's ORF name}

SEQUENCE = {the gene fragment}

= {used to distinguish gene fragments}

Table 1. Standard Input Format with Tags for Each Gene  
(the identifiers on the left serve as prompts.)

\* The Contents Of Global.txt

Table 2 on page 43 shows the parameters in the Global.txt file. The default values are shown in the table. All of these values are subject to a manual override.

Parameter name	Default value
PRIMER_PRODUCT_SIZE_RANGE	300-800
PRIMER_OPT_SIZE	24
PRIMER_MIN_SIZE	24
PRIMER_MAX_SIZE	24
PRIMER_OPT_TM	60.0
PRIMER_MIN_TM	58.0
PRIMER_MAX_TM	62.0
PRIMER_OPT_GC_PERCENT	50.0
PRIMER_MIN_GC	30.0
PRIMER_MAX_GC	80.0
PRIMER_SALT_CONC	50.0
PRIMER_SELF_ANY	8.00
PRIMER_SELF_END	3.00
PRIMER_NUM_RETURN	1

Table 2. Parameters in file Global.txt

Whenever optimal primers cannot be obtained from the initial run, it is possible and desirable to repeat the above processes for a second and a third run. The differences among these runs are the left and/or right end points are extended outward by a selected length from the normal gene location. The default length is 150 bases.

#### \* Output Primers Locally And Re-format

Since the MIT programs produce results in addition to what we need, we re-format the output to meet our requirements. We first save the output in a file named Out.txt. Next, we filter the results into a file named filter.txt. The filter.txt file might contain such things as the gene's ORF name, primer pairs, and primer product size at the user's option. We then separate genes that have primer output from those that have no primer output. The file noError.txt contains the results for those genes with primer output; the file error.txt contains the names of genes for which there was no primer output. The error.txt file can be used as input for subsequent runs to obtain primers for those genes. Finally, we re-format the contents of the file noError.txt and place the result of re-formatting into a file submit.txt to be input into a local database. This process is shown in figure 6 on page 45.

To retrieve the results of those optimal primers, the user could view them through the commands: "view submit1.txt", "view submit2.txt", and "view submit3.txt" under the same Unix/Linux platform. Or more conveniently, the user could transfer those outputs and save them in a local drive (C drive) by FTP transmission.

The third way is to transfer those outputs in a local database server, such as Microsoft Access, for better visualization at a later time.

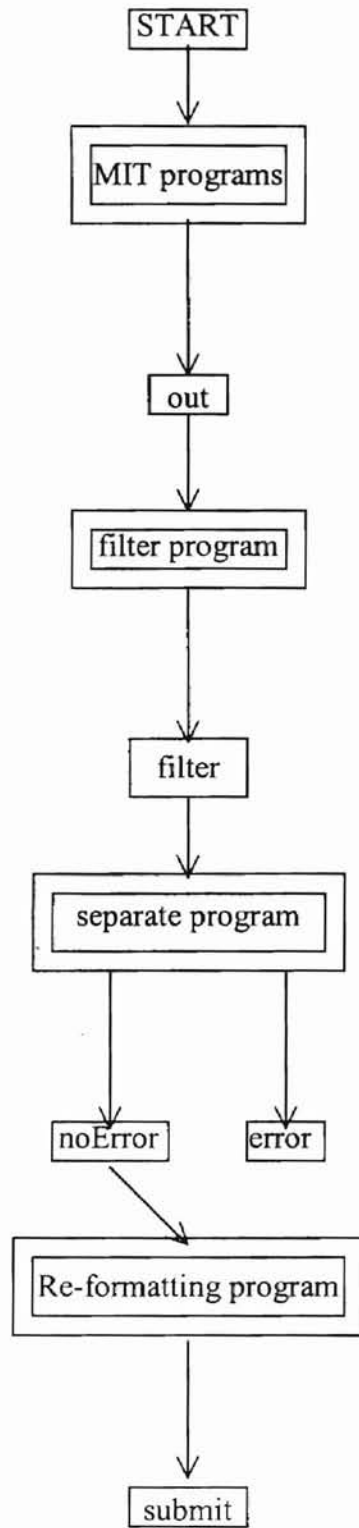


Figure 6. Local Primer Output  
(all files have an extension of ".txt".)

APPENDIX C  
ACRONYMS AND ABBREVIATIONS

## ACRONYMS AND ABBREVIATIONS

DNA	deoxyribonucleic acid
dNTP	a mixture of dATP (A), dCTP (C), dGTP (G), and dTTP (T)
GC	guanine and cytosine (base pair)
GUI	graphic user interface
LAN	local area network
Mb	megabase
ORF	open reading frame
PCR	polymerase chain reaction
RNA	ribonucleic acid
T <sub>m</sub>	melting temperature of a DNA sequence

APPENDIX D

GLOSSARY

## GLOSSARY

**adenine** A nitrogen-containing, single-ring, basic purine that occurs in nucleotides of DNA and RNA. One member of the base pair A-T.

**algorithm** Any sequence of actions (e.g., computational steps) that perform a particular task.

**amplification** Increasing the number of copies of a specific DNA molecule.

**annealing** The process in which two strands of nucleic acid interact by hydrogen bonding between complementary base pairs to form a duplex molecule.

**base pair (bp)** A pair of complementary nucleotides in double stranded DNA.

**base sequence** The arrangement of bases (A, T; G, C) that defines the sequence of nucleotides in DNA ultimately to specify the sequence of amino acids in a protein.

**chemical base** An alternative name for DNA bases (adenine, thymine, guanine, cytosine).

**clarified gene** A gene whose base sequence has been identified.

**client** A computer that makes requests of another computer (server).

**client/server** A computer hardware configuration whereby one or more computers (the clients) make requests of another computer (the server) over a network.

**cytosine** a nitrogen-containing, double-ring, basic pyrimidine that occurs in the nucleotides of DNA and RNA. One member of the base pair G-C.



**denaturation** The process of separating two complementary strands of double-strand DNA.

**DNA (deoxyribonuclei acid)** Nucleic acid comprising the nucleosides deoxyadenosine, deoxyadenosine, deoxycytidine, deoxyguanosine and deoxythymidine.

**DNA polymerase** An enzyme which synthesizes DNA, using a single-stranded nucleic acid as a template. DNA polymerases generally require a double-stranded region to extend.

**DNA template** A single-stranded nucleic acid, serving as the place where the primers will anneal to and extend along with.

**downstream** the 3' end (ending part) of one DNA sequence.

**download** To transfer a file from a remote host to a local machine via FTP, etc.

**FTP** The file transfer protocol used to move computer files from one computer to another on the internet.

**gene** The fundamental physical and functional unit of heredity. Many genes encode proteins.

**gene sequence** A sequence of nucleotides that codes for a protein.

**genome** All of the genes in an organism.

**genome scale** It refer to work or process applied to the entire genome rather than on individual genes.

**guanine** A nitrogen-containing, single-ring, basic pyrimidine that occurs in the nucleotides of DNA and RNA. One member of the base pair G-C.

**hybridization** The process of complementary base pairing between two single strands of nucleic acid.

**hybridization analysis** The analysis procedure about the result of hybridization.

**internet** A system of linked computer networks used for the transmission of files and messages between hosts.

**intranet** A computer network internal to a company or organization. Intranets are often not connected to the Internet or are protected by a firewall.

**local server** A computer that process requests issued from client machine that are nearby (usually on the same LAN).

**oligomer** A short segment from its entire structure.

**oligonucleotide** Several nucleotides joined together to form a short, single-stranded DNA molecule.

**ORF (open reading frame)** A region of gene sequence between the gene's stop codons.

**PCR amplification** Increasing the number of copies of a specific DNA molecule by PCR method.

**polymerase** An enzyme which synthesizes DNA or RNA, using a single-stranded nucleic acid as a template. DNA polymerase generally require a double-stranded region to extend.

**polymerase chain reaction (PCR)** A technique for producing large amounts of specific segments of DNA without resorting to cloning. In vitro amplification of a specific segment of DNA bounded by two oligonucleotide primers by repeated denaturation, annealing, and extension steps.

**primer** A single stranded DNA, often an oligonucleotide, used in DNA synthesis to serve as a starting point for polymerization of a second chain.

**primer pair** A pair of primers designed for two strands of DNA in the PCR process.

**purine** One of the basic chemical structures in DNA bases. It is linked to the sugar of DNA at N<sub>9</sub>.

**pyrimidine** One of the basic chemical structures in DNA bases. It is linked to the sugar of DNA at N<sub>1</sub>.

**remote server** A computer that processes requests issued from remote locations by client machines.

**server** A computer and/or its software that processes requests issued from remote locations by client machines.

**thymine** A nitrogen-containing double-ring, basic purine that occurs in the nucleotides of DNA. Thymine is a member of the base pair, A-T.

**T<sub>m</sub> (melting temperature)** The temperature at which the transition from double-stranded to single-stranded DNA is 50% complete.

**transcription** Synthesis of RNA directed by a DNA template using the enzyme RNA-polymerase.

**upstream** the 5' end (starting part) of a DNA sequence.

**World Wide Web (WWW)** A document delivery system capable of handling not-text-based media of various type.

## VITA

Haihui Huang

Candidate for the Degree of

Master of Science

Thesis: AN ALGORITHMIC PROGRAMMING STRATEGY USING THE  
CLIENT/SERVER MODEL IN POLYMERASE CHAIN REACTION (PCR)  
PRIMER DESIGN

Major Field: Computer Science

Biographical:

Personal Data: Born in Hunan, P.R. China, on April 09, 1970, the daughter of Dongling Yin and Weixian Huang; married to Haobo Liu in 1995.

Education: Graduated from YaLi Middle School, Changsha, Hunan, P.R. China in June 1987. Received Bachelor of Medical Science degree from Hunan Medical University, Changsha, Hunan, P.R. China in May 1993. Received Master of Science degree in Physiology from Oklahoma State University, Stillwater, Oklahoma in July 1998. Completed the requirements for the Master of Science degree with a major in Computer Science at Oklahoma State University in December 1999.

Experience: Employed by the Baoan People's Hospital as a Doctor of Gynecology and Obstetrics, Shenzhen, Guangdong, P.R. China, 1993 - 1995. Employed by the Department of Physiological Science at Oklahoma State University in Stillwater, Oklahoma as a Teaching Assistant, 1996 - 1998. Employed by Euronet Services, Inc. at Kansas City, KS as an intern of Software Engineer, June 1999 to August 1999. Employed by the Department of Microbiology and Molecular Genetics at Oklahoma State University in Stillwater, Oklahoma as a Research Assistant, April 1999 to October 1999.