PRIMEGENSV2: A SERVICE BASED PLATFORM USING OBJECT ORIENTED MODEL VIEW CONTROLLER DESIGN PATTERN

A Thesis presented to the Faculty of the Graduate School
University of Missouri

In Partial Fulfillment
Of the Requirements for the Degree

Master of Science

by

MUNEENDRA OJHA

Professor Dong Xu, Thesis Supervisor

MAY, 2009

The undersigned, appointed by the Dean of the Graduate School, have examined the thesis entitled

PRIMEGENSV2: A SERVICE BASED PLATFORM USING OBJECT ORIENTED MODEL VIEW CONTROLLER DESIGN PATTERN

Presented by Muneendra Ojha

A candidate for the degree of Master of Science

And hereby certify that in their opinion it is worthy of acceptance.

Professor Dong Xu

Dr. Jianlin Cheng

Dr. Michael Wang

Dedicated to my family….

ACKNOWLEGEMENTS

I am very grateful to my advisor Prof. Dong Xu for his constant guidance and support throughout my research and stay at Mizzou. His suggestions and ideas were extremely helpful.

I would also like to express my sincere gratitude towards Mr. Gyan Prakash Srivastava for his never ending source of encouragement.

I am also very thankful to Dr. Michael Wang and Dr. Jianlin Cheng for agreeing to serve on my master's defense committee.

Many thanks to all my fellow lab members for providing me a peaceful and cooperative work environment. I will always cherish the memories I gained in Digital Biology Laboratory.

At this occasion, I would like to pay my heartfelt gratitude to Mrs. Kala Gupta and Mr. Mangal S. Gupta, who have always showered me with blessings and unconditional love. It was only their support which made a person, as home sick as me, stay here in Columbia.

Last but not the least; I would like to thank my parents for their unconditional love and support they have rendered me throughout my life. Today, whatever I have achieved it would not have been possible without them. I thank God for giving me such a wonderful gift in my life.

Table of Contents

CHAPTER

APPENDIX

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

There are a number of tools and software packages dealing with the problem of efficient primer and probe design. Some of these tools are well accepted and very famous among biologists and some tools tend to disappear in oblivion as time passes by. In an era of fast paced research it is rather difficult for any researcher to change their favorite tool which they have been using for some time and are adept in. It has been seen that some very good algorithm and corresponding tools do not get wide publicity and acceptance in biological circles due to lack of users. So the question is how to bring such tools on a common platform and compare them for efficiency and level of accuracy.

Some of these tools are exceptional in designing primers and some are good at designing probes. Some alignment tools excel over others and some tools provide better alignment. None of the tools can be rejected altogether and none of these can be accepted for complete set of procedures. Thus instead of working against one another, the need is to use best functionalities of each tool so that result of one can be incorporated for use by another tool.

In this thesis I tend to develop a structure which will provide biologists and software developers, a common platform for benchmarking their primer design application and pit them against already renowned systems for performance evaluation. This platform would be easy to use and will have ports where third party tools and software packages can be plugged. These programs would help design primers as well as investigate their properties. For a common set of requirements, this platform can provide comparison between tools' performances and for different functionalities it can provide best possible

results of one tool to be passed on the other tool. This system is developed in Java language and uses latest Swing technologies which make it a lightweight application and mild in terms of system requirement. This application provides a very interactive graphical user interface for biologists. For computer scientists and developers, this application gives a combined approach towards service based architecture using object oriented methodologies of abstraction and encapsulation. Application release notes and its related manuals can be found on the webpage of Digital Biology Laboratory (http://digbio.missouri.edu/primegens/).

CHAPTER 1. INTRODUCTION

1.1. Introduction

PCR primer design is an everyday, but not trivial task which requires state-of-the-art software. As the importance and need of good quality primer is being accepted, a lot of primer design software and tools are coming out in research journals for biologists. A researcher on the other hand is generally apprehensive with the quality and reliability of tools freely available in the market. PCR is time consuming process and result of experiments cannot be relied upon the chances of new design algorithm. So as a safe resort, researchers depend on commercial products to save time and effort of trials.

There are numerous free software tools available online for low-throughput design of primers, some of which are highly configurable and well suited for primer design. But for high throughput data, none of the software tools provides a complete set of functionalities ranging from data collection, preprocessing, primer design, specificity checking, and targeted design to avoid DNA contamination [1]. Currently, researchers have to manually move data and information such as parameters, query sequences, databases and others from one tool to other. Sometimes user would have to perform some steps completely on their own such as specificity checking using an alignment package like DIALIGN [2]or ClustalW[3]. Manual steps make researchers loose lot of precious time and force them to make certain crucial interpretation of huge complex dataset, which require expert knowledge. Otherwise it may introduce bias in the result.

Even when the number of online databases is rapidly increasing, their relevance is very limited with respect to the scope of experimentation possibilities. Available databases tend to address a particular problem or scenario. They are practically useless when prerequisites of an experiment changes even for a single parameter. These databases also tend to cover few very famous species, such as human and mouse, which have almost become over represented. There are either very few transcripts of species represented or inappropriate design parameters are used for combined analysis requiring time consuming optimization or PCR amplification conditions.

Efficiency of a primer design algorithm can only be examined by researchers using and assessing the results. So the requirement of the moment is not a very revolutionary algorithm, instead a platform where already available tools can be plugged in and compared on similar datasets. We need is to develop a standard software framework where such tools can be plugged-in with very little or virtually no efforts. We must provide researchers with easy to use interface where they can process datasets in some format, use that dataset to generate PCR primer with a number of design tool, then preview the results generated using similar and familiar output format, examine results using alignment techniques and check the specificity.

1.2. Primer Design with PRIMEGENS

PRIMEGENS[4] identifies a fragment specific to an open reading frame (ORF) and then designs forward and reverse primers based on the selected gene specific fragment to allow ORF specific amplification by PCR. To perform this, PRIMEGENS first carries out a heuristic BLAST search for each query frame against all other query frames so that possible homologous sequences can be identified. It then performs alignment between query and each of the homologous ORFs using dynamic programming. It then uses Primer3 for primer design which takes a DNA fragment selected in the first task and designs PCR primer pairs. PRIMEGENS also accepts parameters for the primer design such as primer size, GC content, melting temperature and self complementarity. To ensure that a primer will not amplify multiple sequences it carries out gapless sequence alignment between two primers generated and all ORFs. If two primers are found to amplify multiple sequences, an alternative gene specific fragment is used to design new primers.

PRIMEGENSv2[5] commandline automates the process of retrieving target sequence database from the human genome database and design primers based on parameters such as distance to the TSS. It tries to avoid the problem of cross-hybridization of fragment-specific primers to any other place of thymine-rich methylated genome. To design primers, this algorithm first converts target sequences and human genome in bisulfite treated sequences. This process of converting cytosine (C) sites in original sequence to thymine (T), except those places where cytosine is preceded by guanine (G), is called Methylation of CG. It then runs Megablast for designed primers considering the bisulfite treated human genome as database. In the result it generates four variants of chromosome sequences as a model for the bisulfite treated sequence: bisulfite methylated forward sequence, bisulfite methylated reverse sequence, bisulfite unmethylated forward sequence and bisulfite unmethylated reverse sequence.

1.3. Present research work

Present work is an extension to the existing primer design software PRIMEGENS [6]. This software, PRIMEGENSv2 is developed with view in mind to provide biologists with an easy to use standalone software having a friendly graphical user interface and functionalities to play with the parameters of primer design and Blast[7] alignment. This software has further facilities to define the type of user input in terms of query sequence, database and type of algorithm. PRIMEGENS which designs primer pairs for high throughput data using Primer3[8] and checks for its uniqueness using BLAST has limitations in terms of user input and parameter choice. The current software overcomes the barrier of so many default values taken by PRIMEGENS and gives user options to modify Primer3 and Blast parameters. Users can generate PCR primers for large databases and view results in different formats. Generated result can be saved and revisited later. Database can either be submitted as a single file, multiple files or a chromosome sequence files.

First part of the present thesis gives a brief overview of software tools available in the field with commercial and academic purposes and their relative merits and demerits. Second part gives a detailed description of the methodology of current work. In the end I have given a brief report of the results generated by this tool and further scope of improvement.

1.4. Objectives for my research work

In my research work, I strive to achieve following objectives:

- Develop a structural model and framework which would provide easy accessibility to primer design tools.
- A design pattern should be developed to create an effective combination of process oriented and object oriented service based architecture. Aim is to develop a system where an individual part of the process can be modified without breaking the whole process.
- The resulting architecture should be plugin friendly and should provide support for introduction of other tools in the future.
- Language and architecture should be such that other developers might feel comfortable in following a pattern.
- System developed should be sufficiently interactive and users should have help or description of terminologies used, at hand.
- Options should be intuitive and sufficient flexibility in parameter values should be provided to the user.
- Returning users should have facilities to edit previous data, check on results, analyze previous run results and save the results for later use.

CHAPTER 2. PCR PRIMER DESIGN

2.1. Polymerase Chain Reaction (PCR)

Developed by Kary Mullis in 1984 [9], PCR or the Polymerase Chain Reaction has become the cornerstone of modern molecular biology the world over. Polymerase chain reaction is a technique for quickly "cloning" a particular piece of DNA in the test tube. The ability to reproduce a target section of a DNA sequence through the use of this technique has facilitated a wide array of amplification techniques [10]. Due to this procedure, one can make virtually unlimited copies of a single DNA molecule even though it is initially present in a mixture containing many different DNA molecules. There might be several reasons to amplify the DNA including DNA cloning for sequencing, DNA-based phylogeny, or functional analysis of genes; the diagnosis of hereditary diseases; the identification of genetic fingerprints (used in forensic sciences and paternity testing); and the detection and diagnosis of infectious diseases. The most famous example might be to create multiple copies of a rare piece of DNA in forensic analysis obtained on a crime scene. A sample, of even saliva on a postage stamp could reveal the identity of a person using this technology.

PCR is composed of cycles of heating and cooling with three major steps:

2.1.1. Denaturing: Cycle begins with denaturing the DNA by heating. This involves breaking apart the double strands of the DNA molecule into single stranded DNAs

2.1.2. **Annealing**: Primers which are short pieces of DNA complementary to a specific sequence bind to their specific part of every single-stranded DNA during cooling.

2.1.3. **Extension/Elongation**: The temperature is raised again and a specific enzyme called Taq polymerase is used to add complementary bases to the DNA from the end of the primer along the rest of the single-stranded DNA templates, thereby making multiple copies of the DNA segment between the two primers.

2.2. Why do we need primers?

It is a Polymerase enzyme that drives a PCR. A polymerase will synthesize a complementary sequence of bases to any single strand of DNA provided it has a double stranded starting point. This is very useful because user can choose which gene they wish the polymerase to amplify in a mixed DNA sample by adding small pieces of DNA complimentary to the gene of interest. These small pieces of DNA are known as primers because they prime the DNA sample ready for the polymerase to bind and begin copying the gene of interest. They function in pairs known as the forward and reverse primers, which amplify a specific DNA sequence and, more importantly, anneal exclusively to that DNA target locus. Each cycle of PCR involves creating two new double stranded DNAs from each DNA molecule present. Thus theoretically, the amount of DNA doubles with every cycle of PCR. After two cycles the concentration of DNA increases by 2^2-folds, after 3 cycles it increases by 2^3-fold and after N cycles, PCR generates 2^N target DNA [11].

The high-throughput assays require designing thousands of PCR primers simultaneously for the experiment. Therefore, careful estimation of the PCR primer and its properties is essential for the success of primer design[12].

2.3. Primer design specifications

A major bottleneck in the polymerase chain reaction is the design of primer pairs, where researchers have to make several choices. Combination of these choices leads to the success or failure of the reaction. The basic oligonucleotides properties and their optimal combinations are well studied by many scientific groups in order to maximize the amplification efficiency[13]. However, in-depth examination of the repeats and the uniqueness of PCR primers in large genomes are still under discussion.

The design considerations that are crucial to specific amplification with high yield are as follows:

2.3.1. Primer length: Both specificity, temperature and time of annealing are partly dependant of primer length. So primer length becomes a very important factor for design of primer[14]. It is generally accepted that the optimal length of PCR primers should be 18-22 bases long. This length is long enough for adequate specificity, and short enough for primers to bind easily to the template at the annealing temperature. It’s important to note that four G’s or C’s are avoided in a row.

2.3.2. Melting temperature (Tm): Melting Temperature (Tm) is defined as the temperature at which one half of the DNA duplex will dissociate to become single stranded and indicates the duplex stability. Melting temperatures in the range of 52-58 degree Celsius is generally found to produce the best results. Primers with melting temperatures above 65 degrees Celsius have a tendency for secondary annealing. GC content of a sequence gives a better indication of the Tm. A working approximation[15] can be calculated as Tm = 2(A+T) + 4(G+C) but for better accuracy, the nearest neighbor thermodynamic[16] values should be used

2.3.3. **Primer annealing temperature (Ta):** The primer melting temperature is the estimate of the DNA-DNA hybrid stability and critical in determining the annealing temperature. Too high Ta will produce insufficient primer-template hybridization resulting in low PCR product yield. Too low Ta may possibly lead to non-specific products caused by a high number of base pair mismatches. Primer annealing temperature is usually 5°C below the calculated lower Tm. However it should be chosen empirically for individual conditions.

2.3.4. **GC content:** The GC content (the number of G's and C's in the primer as a percentage of the total bases) of primer should be 40-60%.

2.3.5. **GC clamp:** The presence of G or C bases within the last five bases from the 3' end of primers. GC clamp helps promote specific binding at the 3' end due to the stronger bonding of G and C bases. More than 3 G's or C's should be avoided in the last 5 bases at the 3' end of the primer.

2.3.6. **Secondary structures:** Presence of the secondary structures produced by intermolecular or intramolecular interactions can lead to poor or no yield of the product. They adversely affect primer template annealing and thus the amplification. They greatly reduce the availability of primers to the reaction.

2.3.7. Repeats: A repeat is a di-nucleotide occurring many times consecutively and should be avoided because they can misprime. For example: ATATATAT. A maximum number of di-nucleotide repeats acceptable is 4 di-nucleotides.

2.3.8. Runs: Primers with long runs of a single base should generally be avoided as they can misprime.

2.3.9. 3' End Stability: It is defined as the maximum ΔG value of the five bases from the 3' end. An unstable 3' end (less negative ΔG) will result in less false priming.

2.3.10. Template secondary structure: A single stranded nucleic acid sequences is highly unstable and fold into conformations (secondary structures). The stability of these template secondary structures depends largely on their free energy and melting temperatures(Tm). Consideration of template secondary structures is important in designing primers. If primers are designed on a secondary structure which is stable even above the annealing temperatures, the primers are unable to bind to the template and the yield of PCR product is significantly affected. Hence, it is important to design primers in the regions of the templates that do not form stable secondary structures during the PCR reaction.

2.3.11. Cross homology: To improve specificity of the primers it is necessary to avoid regions of homology. Primers designed for a sequence must not amplify other genes in the mixture. Commonly, primers are designed and then BLASTed to test the specificity.

CHAPTER 3. LITERATURE REVIEW AND MOTIVATION

Number of online tools and pre-designed primer databases is rapidly increasing. Some tools have come and gone without notice and some tools became famous only to be deserted by their own developers with websites rarely maintained or updated. Here we take a look on some well known online tools, software packages and pre-designed PCR primer or probe databases.

3.1. Current state of software development

There are some software utilities that are freely available to the academic community for the design of primer pairs for high throughput data[1], including Primer3[8] and its interfaces such as Primer3Plus[17], AutoPrime[18] and BatchPrimer3[19]. Some databases of pre-designed primers like RTPrimerDB[20], PrimerBank[21], qPrimerDepot[22], AtRTPrimer[23] and DATFAP[24] are also available over Internet. There are a number of simple standalone programs as well as complex and networked versions of commercial software available[25]. Major names engaged in software development for biologists include Accelrys which in turn itself is founded by bringing together companies like Molecular Simulations Inc. (MSI), Synopsys Scientific Systems, Oxford Molecular, the Genetics Computer Group (GCG®), and Synomics Ltd. Then there are other players such as: PREMIER Biosoft International, DNAStar, MedProbe and many more appearing rapidly.

3.2. State of primer design tools available for online use

Among the online primer design tools, CODEHOP[26] (COnsensus-DEgenerate Hybrid Oligonucleotide Primer) (http://bioinformatics.weizmann.ac.il/blocks/codehop.html) designs a pool of primers for the 3' degenerate core region having the most probable nucleotide predicted for each position in the 5' non-degenerate clamp region. Input for this program is set of related sequences or multiple alignment blocks generated using either EBI Clustal or BCM Search Launcher. There are options to change degeneracy, strictness, temperature and weight of favored sequence(s). GeneFisher[27] (http://bibiserv.techfak.uni-bielefeld.de/genefisher2) is an interactive web based program for designing degenerate primers. It is based on the assumption that genes with related function from different organisms show high sequence similarity so degenerate primers can be designed from sequences of homologues genes. It can accept unaligned sequences. This tool has further been enhanced in GeneFisher-P[28] with better service based process architecture leading to a collection of basic services from heterogeneous sources orchestrated in the Bio-jETI platform. BiSearch[29-31] web server is a useful tool for primer-design for any DNA template and especially for bisulfite-treated genomes. The ePCR tool for fast detection of mispriming sites and alternative PCR products in cDNA libraries and native or bisulfite-treated genomes are the unique features of the new version of BiSearch software.

Primer3 (http://frodo.wi.mit.edu/) is the most elaborate and famous tool being used by academic institutions. It’s a comprehensive PCR primer and hybridization probe design tool. Web Primer (http://compute.yeastgenome.org/cgi-bin/web-primer) is a tool which allows alternative design of primer for PCR or sequencing purposes. Primo Pro 3.4 (http://www.changbioscience.com/primo/primo.html) reduces PCR noise by lowering the probability of random primering. Primo Degenerate 3.4 (http://www.changbioscience.com/primo/primod.html) designs primer pairs based on single peptide sequence or multiple alignment of protein or nucleotides. Primer Generator[32] (http://www.med.jhu.edu/medcenter/primer/primer.cgi) analyzes the original nucleotide sequence and desired amino acid sequence and designs a primer that has either new restriction enzyme site or is missing an old one. EPRIMER3 (http://mobyle.pasteur.fr/cgi-bin/MobylePortal/portal.py?form=eprimer3) picks PCR primers and hybridization oligos.

PrimerQuest (http://www.idtdna.com/Scitools/Applications/Primerquest/) is another primer design tool from IDT which comes in three flavors, Basic, Standard and Advanced. It gives options to design primers or probes with real-timer PCR or PCR Primers parameter sets. MethPrimer[33] (http://www.urogene.org/methprimer/ index1.html) designs primers for methylation specific PCRs. Primer prim'er project (http://www-nmr.cabm.rutgers.edu/bioinformatics/Primer_Primer/previous_versions/3.x/ primer_primer_v3.2/Primer.html) aims to completely automate the primer design process. The CBS ProbeWiz Server[34] (http://www.cbs.dtu.dk/services/DNAarray/probewiz.php) server predicts optimal PCR primer pairs for generation of probes for cDNA arrays. OligoHeatMap (OHM)[35] is an online tool (http://bioinfo.unice.fr/softwares/ohm/) that provides estimates of T_m for a set of oligomers and a set of aligned sequences, not only as text files of complete results but also in a graphical way.

Among the newer ones are Primique[36], a graphical, user-friendly, web-based tool which designs (http://cgi-www.daimi.au.dk/cgi-chili/primique/front.py) specific primers for each sequence in an uploaded set. Primer3Plus[17] is a web interface (http://www.bioinformatics.nl/cgi-bin/primer3plus/primer3plus.cgi) to the famous and very successful Primer3[8] primer design program. It provides an intuitive user interface using web technologies and has been developed in collaboration with molecular biologists and technicians designing primers. AutoPrime[18] (http://www.AutoPrime.de) aims to find expressed sequences specific for the mRNA sequences instead of yielding a product from its genomic sequence to reduce even slight genomic contamination. BatchPrimer3 is a web based high throughput application for PCR and sequencing primer design. It uses Primer3 core program as a major primer design engine to choose the best primer pairs.

Table 1: Relative comparison of different primer design tools available for online use

Tool name	Free for academic use	Open source	User-defined database input	High-throughput scale	Hybridization check	Bisulfite mode
CODEHOP	Yes	No	No	No	No	No
GeneFisher-P	Yes	No	No	Yes	No	No
BiSearch	Yes	No	No	No	No	Yes
Primer3Plus	Yes	Yes	No	Yes	Yes	No
Web Primer	Yes	No	No	No	No	No
Primo Degenerate	Yes	No	No	No	No	No
Primer Generator	Yes	No	No	No	No	No
EPRIMER3	Yes	No	No	Yes	No	No
PrimerQuest	Yes	No	No	Yes	No	No
MethPrimer	Yes	No	No	No	No	Yes
ProbeWiz	Yes	No	Yes	No	Yes	No
Primique	Yes	No	Yes	Yes	No	No
AutoPrime	Yes	No	Yes	No	Yes	No

BatchPrimer3	Yes	No	No	Yes	Yes	No

3.3. Software packages and downloadable tools

A greater number of companies are creating commercial applications and packages providing several tools for various biological purposes. Some of these companies supply specifically designed and synthesized primer. Other companies provide software licenses for standalone machines so that a user can design the type of primer he/she wants. Let us take a look at some of the software. We start with PrimerSelect[37] (http://www.dnastar.com/products/primerselect.php). It is a module of Lasergene product developed by DNAStar (www.dnastar.com) and not available for free. It analyzes a template DNA sequence and chooses primer pairs for PCR and primers for DNA sequenceing. DNASIS software (http://www.miraibio.com/dnasis-max/dnasis-max-overview.html) from MiraiBio, a group of Hitachi Software is commercially available software for life scientists which can edit, annotate and analyze DNA, RNA and amino acid sequences. It can be expanded with optional homology search, multiple alignment and base calling and sequence assembly (Phred/Phrap) modules. Primer Premier (http://www.premierbiosoft.com/primerdesign/index.html) is a software tool from Premier Biosoft international which can be downloaded and be used for free for a limited time. It can be used for PCR, hybridization, cross species, sequencing and allele specific polymerase chain reaction.

Premier Biosoft provides several other tools like NetPrimer (http:// www.premierbiosoft.com/netprimer/index.html) for comprehensive analysis of individual primer and primer pair, Array Designer (http://www.premierbiosoft.com/

dnamicroarray/index.html) for fast effective design of specific oligos or PCR primer pairs for microarrays and Beacon Designer (http://www.premierbiosoft.com/molecular_ beacons/index.html) for designing molecular beacons and TaqMan probes for robust amplification and fluorescence in real time PCR. All these are available for limited time free use. GenomePRIDE 1.0 (http://pride.molgen.mpg.de/genomepride.html) and PRIDE 2.0 (http://pride.molgen.mpg.de/pride.html) are primer design software available to academicians for a minimal fee. PRIDE automatically designs primers in single contigs or whole sequencing projects to extend the already known sequence and to double strand single-stranded regions and GenomePRIDE designs PCR primers or long oligos on an annotated sequence.

FastPCR (http://www.biocenter.helsinki.fi/bi/Programs/fastpcr.htm) is a freely downloadable program from Institute of Biotechnology, University of Helsinki. It is an integrated tool for automatic and manual PCR primers or probe design, alignment and any repeat searching. It has specific ready to use templates for many PCR and sequencing applications like standard and long PCR, inverse PCR, degenerate PCR, directly on amino acid sequences and multiplex PCR. OLIGO6 (http://www.oligo.net/) is commercially available software for Windows and Macintosh for designing and analyzing sequencing and PCR primers, synthetic genes, and various kinds of probes including siRNA and molecular beacons. The Genamics Expression (http://genamics.com/expression/primer.htm) from Genamics, features a powerful, simple, real-time interface to allow the rapid identification of theoretical ideal primers for PCR reactions.

Osprey[38] calculates oligonucleotides for sequence assembly, differential expression, and microarrays (cDNA and spotted oligos). It implements the thermodynamics model[16] and has other features like sensitive secondary binding checks, specification and reporting of secondary binding in terms of melting temperature margins, automated optimal length/melting temperature/self-folding energy calculations based on nucleotide composition and automated repeats filtering or discrimination. Osprey is available free for academic purposes (http://osprey.ucalgary.ca/). PRIMEGENS-v2[5] is the freely available tool developed by Digital Biology Laboratory in University of Missouri. It is capable of designing sequence specific primer pairs for bisulfite-treated sequences at the genomic scale. This method is incorporated into the second version of PRIMEGENS. It not only searches for appropriate primers but also checks for non-specific PCR amplification.

Table 2: Relative comparison of downloadable software tools available primer or probe design and analysis

Tool name	Free for academic use	Open source	High-throughput scale	Hybridization check	Bisulfite mode
PrimerSelect	No	No	Yes	Yes	No
Primer Premier	No	No	Yes	Yes	No
NetPrimer	Yes	No	No	No	No
Array Designer	No	No	Yes	No	No
GenomePRIDE	No	No	Yes	Yes	No
PRIDE	No	No	Yes	No	No
FastPCR	Yes	No	Yes	Yes	No
OLIGO6	No	No	Yes	Yes	No
Expression	No	No	Yes	No	No
Osprey	Yes	No	Yes	No	No
Primer3	Yes	Yes	Yes	No	No
PRIMEGENS-v2	Yes	No	Yes	Yes	Yes

3.4. Online pre-designed PCR primer and Probe databases

RTPrimerDB[20, 39, 40] (http://www.rtprimerdb.org) is a freely accessible database and analysis tool for real-time quantitative PCR assays. It includes records with user submitted assays that are linked to genome information from reference databases and quality controlled using an in silico assay evaluation system. PrimerBank[21] a huge collection of primers (http://pga.mgh.harvard.edu/primerbank/index.html) over 306,800 covering most human and mouse genes. Primers can be searched in several ways like using GenBank Accession, NCBI protein accession, NCBI Gene ID or Gene Symbol. qPrimerDepot[22] database (http://primerdepot.nci.nih.gov/) provides quantitative real time PCR primers for 99.96% human RefSeq sequences. All primers available here have annealing temperatures approximately 60C. There is no option for users to submit their primers in this database. DATFAP[24] is a database of transcription factors with alignment and primers from 13 plant species (http://cgi-www.daimi.au.dk/cgi-chili/datfap/frontdoor.py), and PCR primers for around 90% of them. Primers are provided for each transcription factor, designed by the software tool primique[36]. Uprobe[41, 42] currently consists of whole-genome sets of universal overgo-hybridization probes designed for screening mammalian or avian/reptilian libraries. Currently, it provides ability to search probes for use in broad groups of species, including mammals and reptiles, as well as more specific clades, including marsupials, carnivores, rodents and nonhuman primates. OligoArrayDb[43] is a comprehensive database containing pangenomic oligonucleotide microarray probe sets designed for most of the sequenced genomes that are not covered by commercial catalog arrays.

Table 3: Relative comparison showing online primer or probe databases available over Internet and their properties.

Database name	Free for academic use	Hybridization check	Bisulfite mode	Number of Primers	User submitted records
RTPrimerDB	Yes	Yes	No	7,780	Yes
PrimerBank	Yes	Yes	No	306,800	Yes
qPrimerDepot	Yes	No	No	NA	No
DATFAP	Yes	Yes	No	55,000	No
Uprobe	Yes	Yes	No	NA	No
OligoArrayDb	Yes	Yes	No	6,958,719	Yes

CHAPTER 4. PRIMEGENSV2: A USER'S PERSPECTIVE

4.1. Services offered by PRIMEGENSv2

Functionalities of the system have been divided into sections based on the services they offer.

- First section is devoted to the data input. We call it Input section. This is mainly concerned with the user input and its formatting. It takes users' files and converts them to a standard format which can be used for any primer design tool selected later.
- Second section is called Option section. It provides users with every kind of parameter options and lets them modify the default values already stored. This section lets users select what type of primer design they want, what design methodology or tools they want to use. Based on the choice of tool selected, it provides them option to modify the tool specific parameters. It then prepares a parameter configuration file which would be used by execution section.
- Third section deals with the execution of selected tools on data prepared. It accepts parameter configuration file created by second section and reformats that into an individual tool specific parameter configuration file. Exec section then runs the PRIMEGENS core that is basically written in C language and other applications to design PCR primers. The generated results are stored in default output folder otherwise in user specified location.

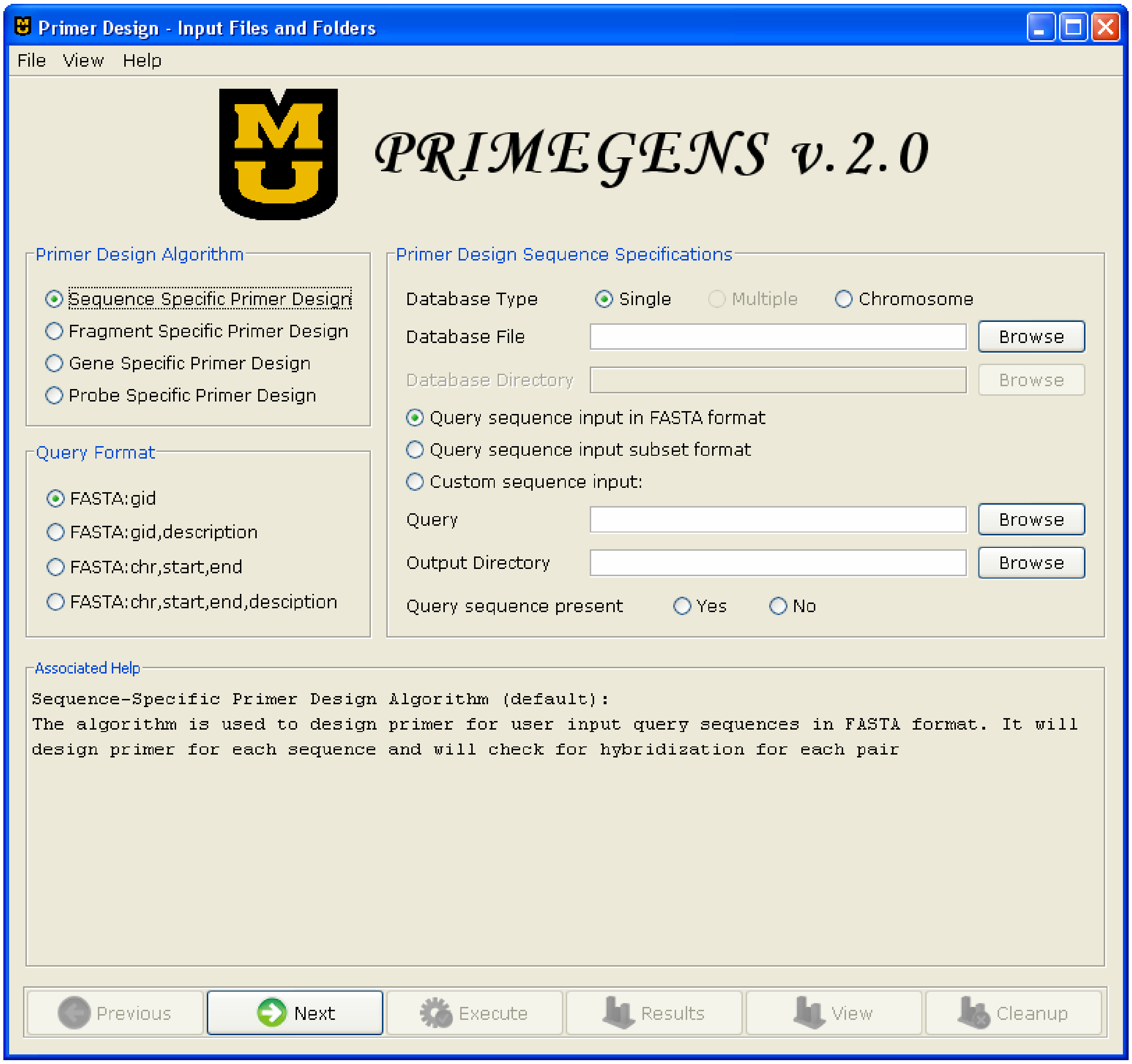
- Fourth section is called Result section and as the name suggests, it deals with display of results. Results generated in the output folder is parsed for data values and displayed in a predefined format. As the output is well read by the system, any query generating from users' action can be easily handled.
- Fifth section is called a Repository and it functions as a supporting static class which has variables and methods to be served for inter-sectional tasks. If a section needs variable methods whose application must be shared by another section then such members are stored in this section.
- Last is the Controller section which effectively keeps every section in order. It creates the starting environment, initiates default values and monitors the inter-sectional activities. It can also be termed as a guardian class.

I explain the first four sections here and last two sections would be explained in the next chapter.

4.2. Input Section

Input section has four basic algorithms namely Sequence Specific Primer Design (SSPD), Fragment Specific Primer Design (FSPD), Probe Specific Primer Design (PSPD) and Gene Specific Primer Design (GSPD). What algorithm is chosen, partly depends upon the variety of database and query sequence file. If a sequence is provided for which primers are to be designed, SSPD should be selected. If user wants probe design as well then PSPD should be selected. If genomic data is supplied, GSPD should be selected. If user wants primers to be generated based on fragments of the database, then FSPD design algorithm type should be selected.

4.2.1. Database: For every algorithm, the common item is database file. It is not important that the database file has to be in FASTA format. Database file can be a single large file or a set of multiple files, having for example chromosomal sequences. If database is in more than one file then, user must select "Multiple" or "Chromosome" in the "Database Type" and also browse to locate the address of folder where the entire database files are kept.



Primer Design Algorithm

● Sequence Specific Primer Design
○ Fragment Specific Primer Design
○ Gene Specific Primer Design
○ Probe Specific Primer Design

Query Format

● FASTA:gid
○ FASTA:gid,description
○ FASTA:chr,start,end
○ FASTA:chr,start,end,desciption

Primer Design Sequence Specifications

Database Type ● Single ○ Multiple ○ Chromosome

Database File

Database Directory

● Query sequence input in FASTA format
○ Query sequence input subset format
○ Custom sequence input:

Query

Output Directory

Query sequence present ○ Yes ○ No

Associated Help

```
Sequence-Specific Primer Design Algorithm (default):
The algorithm is used to design primer for user input query sequences in FASTA format. It will
design primer for each sequence and will check for hybridization for each pair
```

Figure 1: Screenshot showing first screen InputFrame. Users can specify their choice of algorithm, database type, query format and submit query sequence for primer design purpose.

4.2.2. Query sequence: This is the user supplied nucleotide sequence for which PCR primer pair is to be generated. User can submit this sequence in three formats.

- FASTA format: A file is submitted which has query sequences, written in FASTA format. Users can convert their files in FASTA format using tool provided by PRIMEGENSv2. An algorithm type of SSPD, FSPD, PSPD or GSPD can be selected for this. If input sequence file is not in proper FASTA format, system automatically converts it.
- Alternatively, a Subset file can be provided which would query sequence names/ids and for every id, a corresponding nucleotide sequence can be found in the database file. This file should be in sub-FASTA format where only sequence identification is provided like this:
  >ID1……..
  >ID2……..
  >ID3........
- If a relatively smaller number of sequences are to be provided, the third option of custom sequence input can be selected. User can provide database, specify output folder and select the algorithm as SSPD. On clicking "Next", user is taken to "Custom Sequence Entry" page where user can enter sequences. If description of the sequence is known, it can be written as IDs, otherwise just write in or paste the sequence. Every sequence should be separated by a blank line. System will convert the entered sequence into FASTA and provide a preview of the reformatted

data. eg.

>IDXYZABCD1234

CGAGCTCGAATTCGTTCGCTTAA….

CTTTATTTGATTGTATCGGTTCG….

Here a blank line is left between two sequences. PRIMEGENS will consider first sequence has an ID which represents the sequence following immediately and the sequence after blank line is a new sequence for which there is no ID. Application would give it a new ID “Unknown Sequence”.

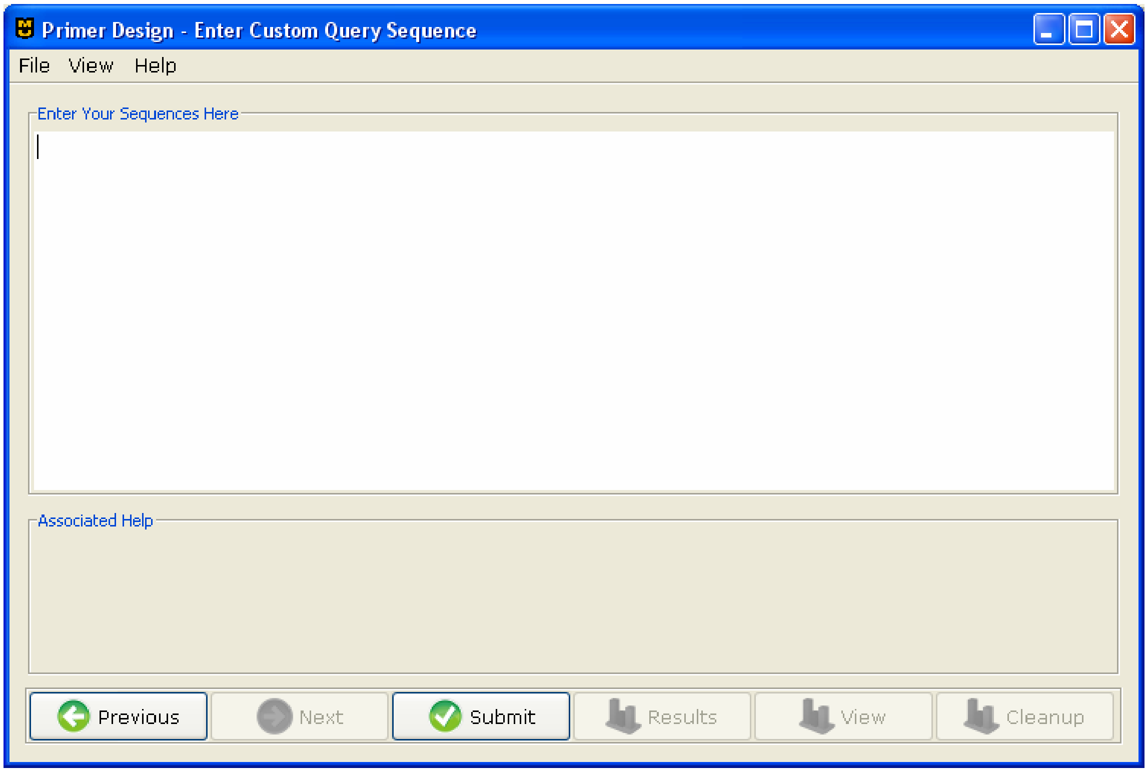

Primer Design - Enter Custom Query Sequence
File View Help
Enter Your Sequences Here
Associated Help
Previous
Next
Submit
Results
View
Cleanup

Figure 2: Screenshot of CustomEntryFrame where users can type in or copy paste their desired nucleotide sequence and after submitting it, system would generate a FASTA formatted version of their data.

4.2.3. Query Format: This is an extra advantage with the sequences a user enters. PRIMEGENSv2 can parse the ID provided for a sequence in four different formats namely: FASTA: gid; FASTA: gid, description; FASTA: chr, start, end; and FASTA: chr, start, end, description. This parsing enables the application to classify data based on description. The word FASTA here just represents that sequence should be in FASTA format. It doesn't need to be written in the descriptor.

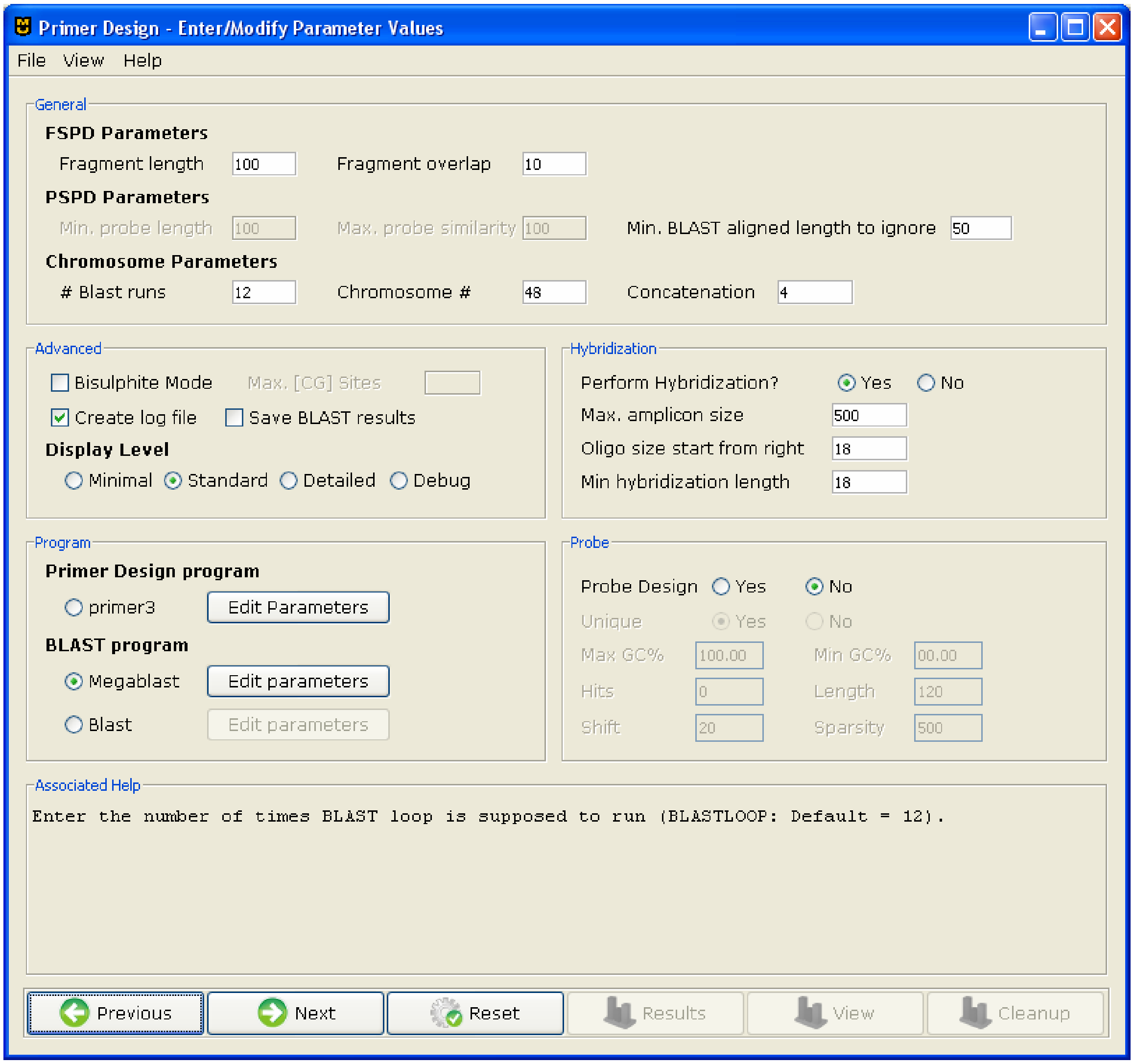
4.2.4. **Database Type:** Database can be provided as a single file, multiple files with part of database in each of them or huge genomic data. This must be mentioned as "Single" for a single file, "Multiple" or "Chromosome" for multiple files in one folder whose address must be provided.

4.2.5. **Output Directory:** User has the option to provide an output directory where generated result ad log files will be saved. If no directory is supplied, system would generate the results in "home/output/" directory. As this directory is a temporary storage, it is users' responsibility to copy their files from this folder. These folder is cleaned up regularly by the application so precious data files might get deleted.

4.3. Option Section

Option section collects experimental parameters. This panel provides default values that are system chosen. Users have the option to change it. There are five sections here: General, Advanced, Hybridization, Program and Probe.

4.3.1. **General:** Values here change according to the design algorithm type selected in Input section. If FSPD is selected, users should provide FSPD length and overlap period values. If PSPD is selected, minimum probe length and maximum probe similarity should be provided. For chromosomal database number of blast runs, total chromosome count and no of chromosomes in a single runs should be provided.



General

FSPD Parameters

Fragment length	100	Fragment overlap	10

PSPD Parameters

Min. probe length	100	Max. probe similarity	100	Min. BLAST aligned length to ignore	50

Chromosome Parameters

# Blast runs	12	Chromosome #	48	Concatenation	4

Advanced

□ Bisulphite Mode Max. [CG] Sites

☑ Create log file □ Save BLAST results

Display Level

○ Minimal ● Standard ○ Detailed ○ Debug

Hybridization

Perform Hybridization? ● Yes ○ No

Max. amplicon size	500
Oligo size start from right	18
Min hybridization length	18

Program

Primer Design program

○ primer3 Edit Parameters

BLAST program

● Megablast Edit parameters

○ Blast Edit parameters

Probe

Probe Design ○ Yes ● No

Unique ● Yes ○ No

Max GC%	100.00	Min GC%	00.00
Hits	0	Length	120
Shift	20	Sparsity	500

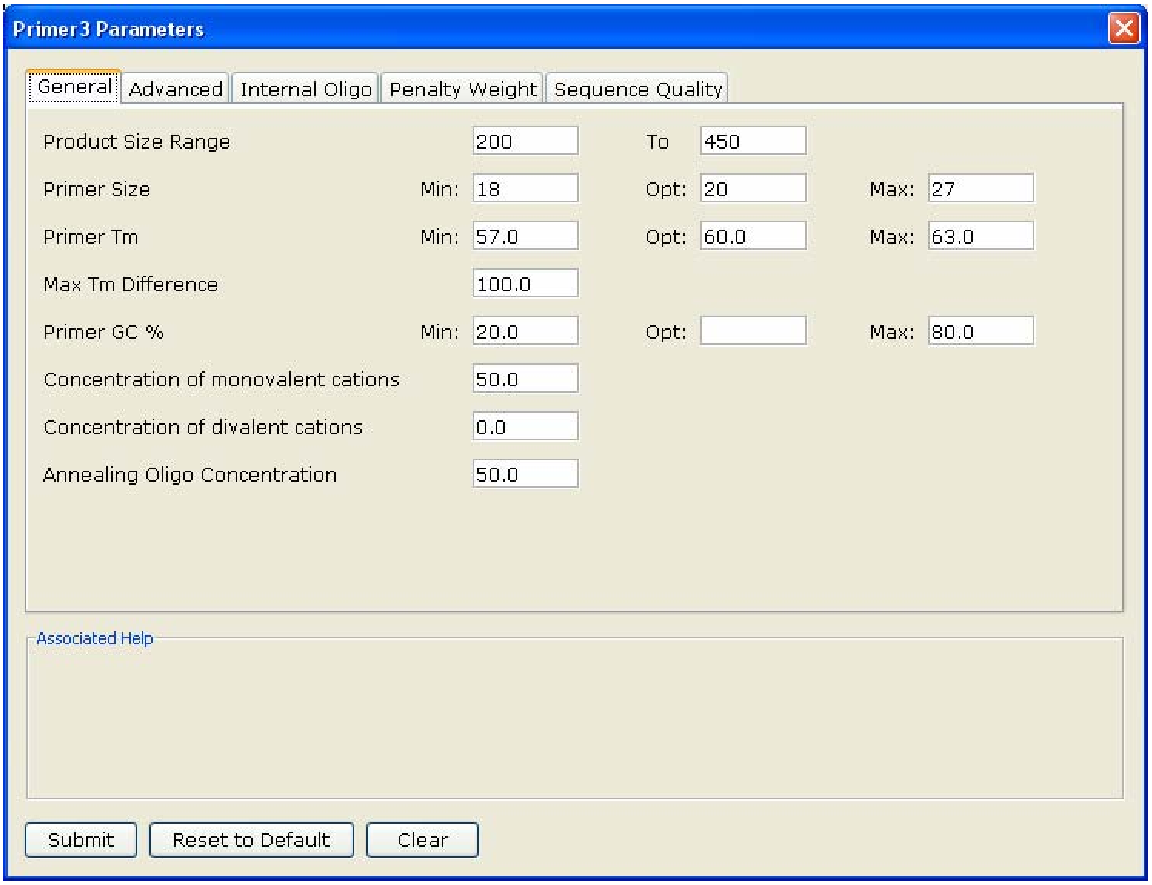
Associated Help

Enter the number of times BLAST loop is supposed to run (BLASTLOOP: Default = 12).

Figure 3: Screenshot of OptionFrame panel which prompts users to enter or modify the Primer design parameters. Currently this application supports Primer3 for primer design and Blast and Megablast programs for Alignment purposes.

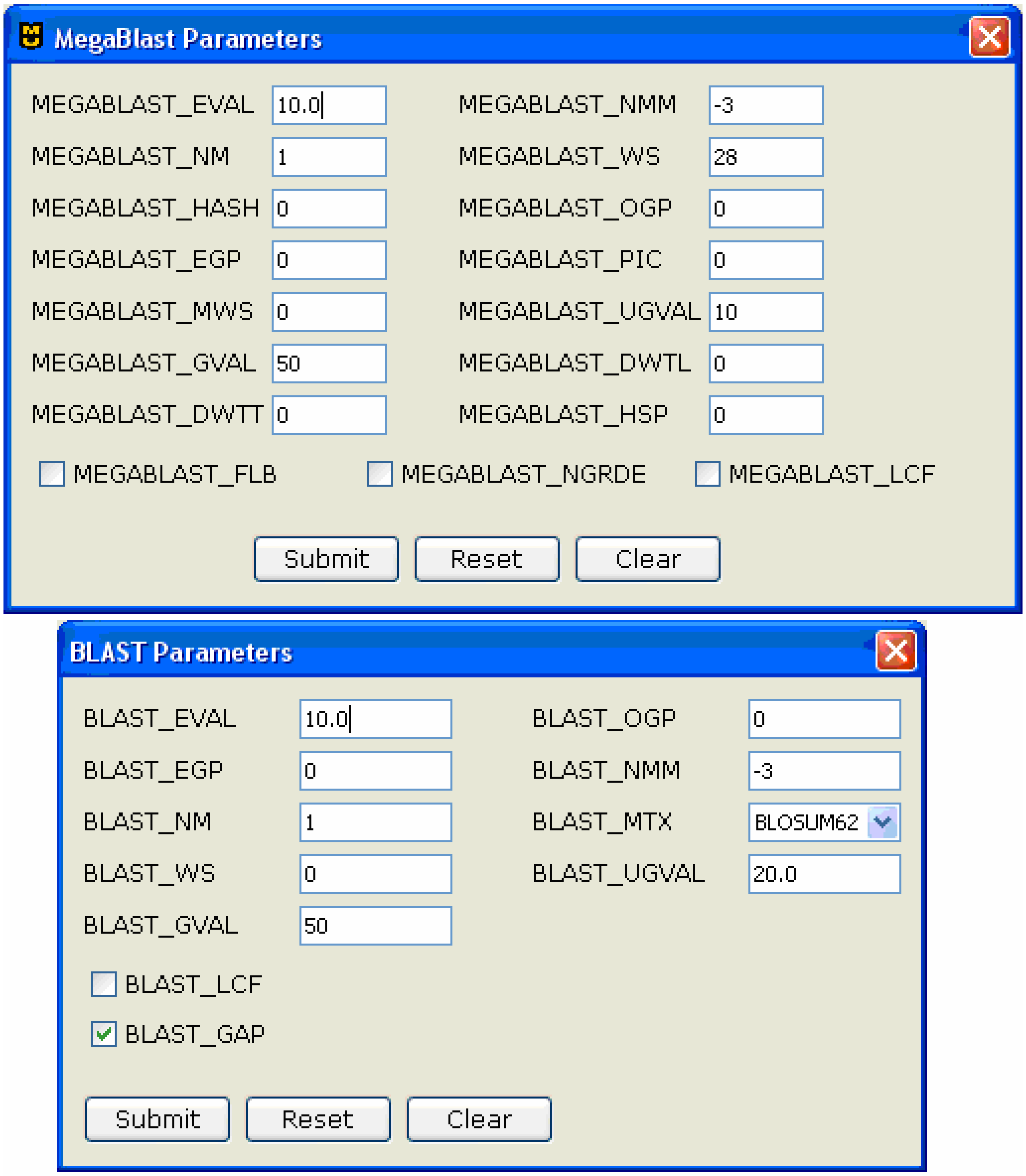
4.3.2. Advanced: Here users should specify if they want primer design in bisulphite mode and if they do, they must provide the maximum number of CG sites value. Users can select option of log creation and saving BLAST results. Display level is the level of runtime details generated by PRIMEGENS when it designs primer. As the name suggests, Minimal is selected if results are needed fast and execution details are not important. Standard is what normally is recommended. Detailed report is requested when step by step design details are required. A display level of debug is required by developers to decipher a bug.

4.3.3. Hybridization: This section accepts users' choice for hybridization, which is by default "Yes". The default values of Maximum Amplicon size, Oligo size and minimum hybridization length are pre-posted. User can turn off the hybridization by selecting "No" option.



Product Size Range		200	To	450		
Primer Size	Min:	18	Opt:	20	Max:	27
Primer Tm	Min:	57.0	Opt:	60.0	Max:	63.0
Max Tm Difference		100.0				
Primer GC %	Min:	20.0	Opt:		Max:	80.0
Concentration of monovalent cations		50.0				
Concentration of divalent cations		0.0				
Annealing Oligo Concentration		50.0				

Figure 4: Screenshot of dialog box where users can enter or modify parameter values for Primer3 program. This screen has been taken care to be made as similar as possible to the latest version of PrimerPlus over Internet to avoid any kind of confusions to practiced users.



MegaBlast Parameters

MEGABLAST_EVAL	10.0	MEGABLAST_NMM	-3
MEGABLAST_NM	1	MEGABLAST_WS	28
MEGABLAST_HASH	0	MEGABLAST_OGP	0
MEGABLAST_EGP	0	MEGABLAST_PIC	0
MEGABLAST_MWS	0	MEGABLAST_UGVAL	10
MEGABLAST_GVAL	50	MEGABLAST_DWTL	0
MEGABLAST_DWTT	0	MEGABLAST_HSP	0

□ MEGABLAST_FLB □ MEGABLAST_NGRDE □ MEGABLAST_LCF

BLAST Parameters

BLAST_EVAL	10.0	BLAST_OGP	0
BLAST_EGP	0	BLAST_NMM	-3
BLAST_NM	1	BLAST_MTX	BLOSUM62
BLAST_WS	0	BLAST_UGVAL	20.0
BLAST_GVAL	50		

□ BLAST_LCF

☑ BLAST_GAP

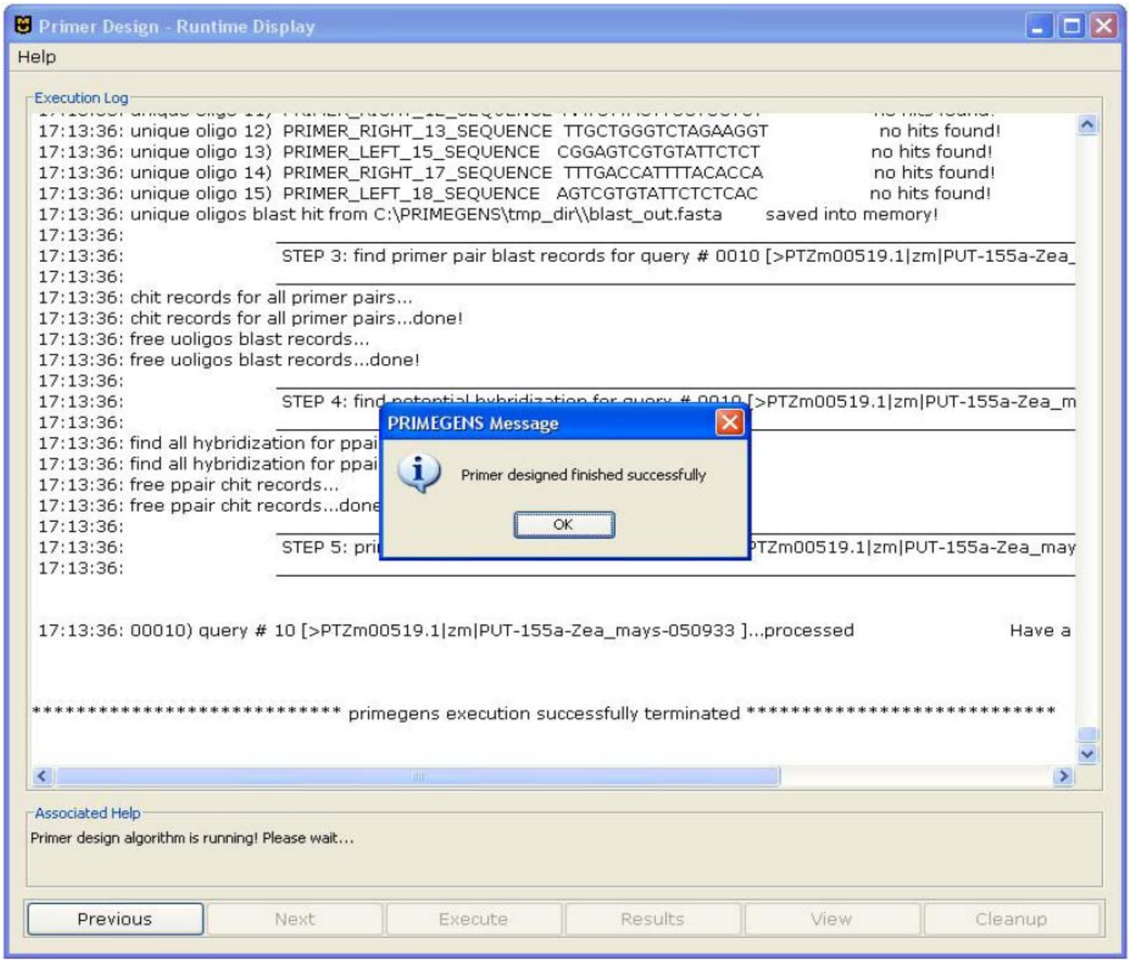
Figure 5: Screenshots showing Megablast and Blast parameter entry dialog boxes.

4.3.4. Program: This section get user's choice for primer design program to be used. It also gives freedom to use different BLAST functions. Currently only Primer3 is being supported but based on out framework we have plans to include more famous primer design algorithms to be included. Individual program parameters can be entered by clicking on the “Edit Parameter” buttons. For the currently supported Primer3 program we have based our input on the latest version of PrimerPlus[17]. Blast or Megablast parameters can also be similarly supplied.

4.3.5. Probe: If user wants to do a probe design, the probe parameter values can be supplied in this part.

4.4. Execution Section

There are not many visually amazing things in this section but an “Execute” button. Based on the user’s choice of algorithm, sequence, parameter values, primer design program and blast program selected this section executes the command-line version of PRIMEGENSv2. For the users familiar with PRIMEGENS command-line version, all the execution-time description displayed on command prompt is displayed here in this window depending upon the level of display selected. Once execution is over, it will show a completion message and enable the result display buttons.



```
17:13:36: unique oligo 12)  PRIMER_RIGHT_13_SEQUENCE  TTGCTGGGTCTAGAAGGT              no hits found!
17:13:36: unique oligo 13)  PRIMER_LEFT_15_SEQUENCE   CGGAGTCGTGTATTCTCT             no hits found!
17:13:36: unique oligo 14)  PRIMER_RIGHT_17_SEQUENCE  TTTGACCATTTTACACCA              no hits found!
17:13:36: unique oligo 15)  PRIMER_LEFT_18_SEQUENCE   AGTCGTGTATTCTCTCAC             no hits found!
17:13:36: unique oligos blast hit from C:\PRIMEGENS\tmp_dir\\blast_out.fasta       saved into memory!
17:13:36:
17:13:36:                    STEP 3: find primer pair blast records for query # 0010 [>PTZm00519.1|zm|PUT-155a-Zea_
17:13:36:
17:13:36: chit records for all primer pairs...
17:13:36: chit records for all primer pairs...done!
17:13:36: free uoligos blast records...
17:13:36: free uoligos blast records...done!
17:13:36:
17:13:36:                    STEP 4: find potential hybridization for query # 0010 [>PTZm00519.1|zm|PUT-155a-Zea_m
17:13:36:
17:13:36: find all hybridization for ppai
17:13:36: find all hybridization for ppai
17:13:36: free ppair chit records...
17:13:36: free ppair chit records...done
17:13:36:
17:13:36:                    STEP 5: pri                              PTZm00519.1|zm|PUT-155a-Zea_may
17:13:36:


17:13:36: 00010) query # 10 [>PTZm00519.1|zm|PUT-155a-Zea_mays-050933 ]...processed                    Have a


************************** primegens execution successfully terminated **************************
```

PRIMEGENS Message

Primer designed finished successfully

OK

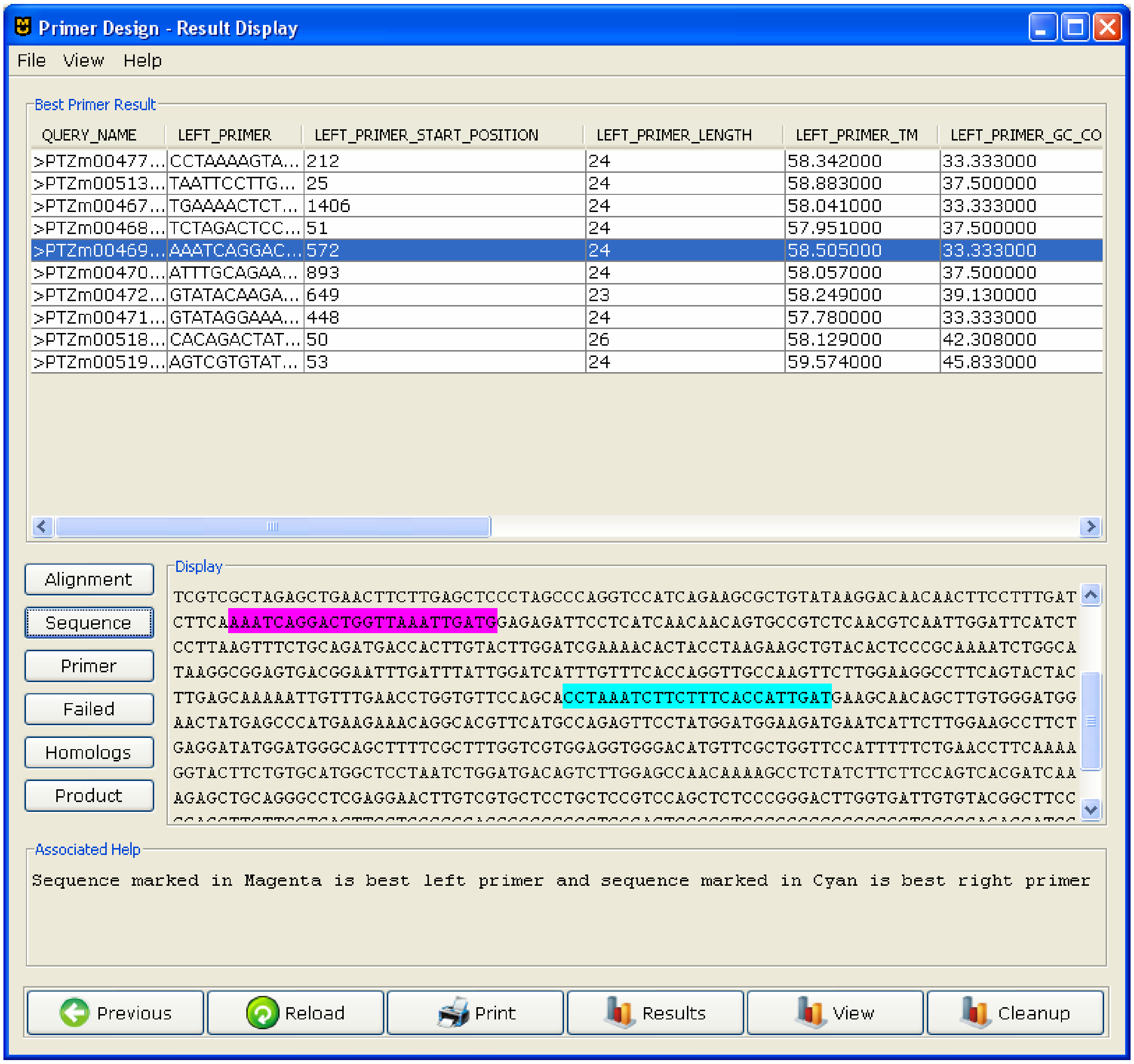
Primer design algorithm is running! Please wait...

Figure 6: Screenshot of the ExecFrame which when executed displays runtime messages generated from PRIMEGENS commandline program. When program finishes, it shows a completion message otherwise prompts users about the error that occurred.

4.5. Results

4.5.1. Viewing immediately generated results: Result screen graphically displays the output files generated by PRIMEGENS. Upper window lists best primer pair results for individual query sequences provided in tabular format. For a single row in the table, user can choose to view sequence for that pair, its alignment and the 20 best primers generated for that sequence. Then sequences for which no primers were generated can be seen by clicking

"Failed" button. Homologs and products for a sequence can also be seen similarly.



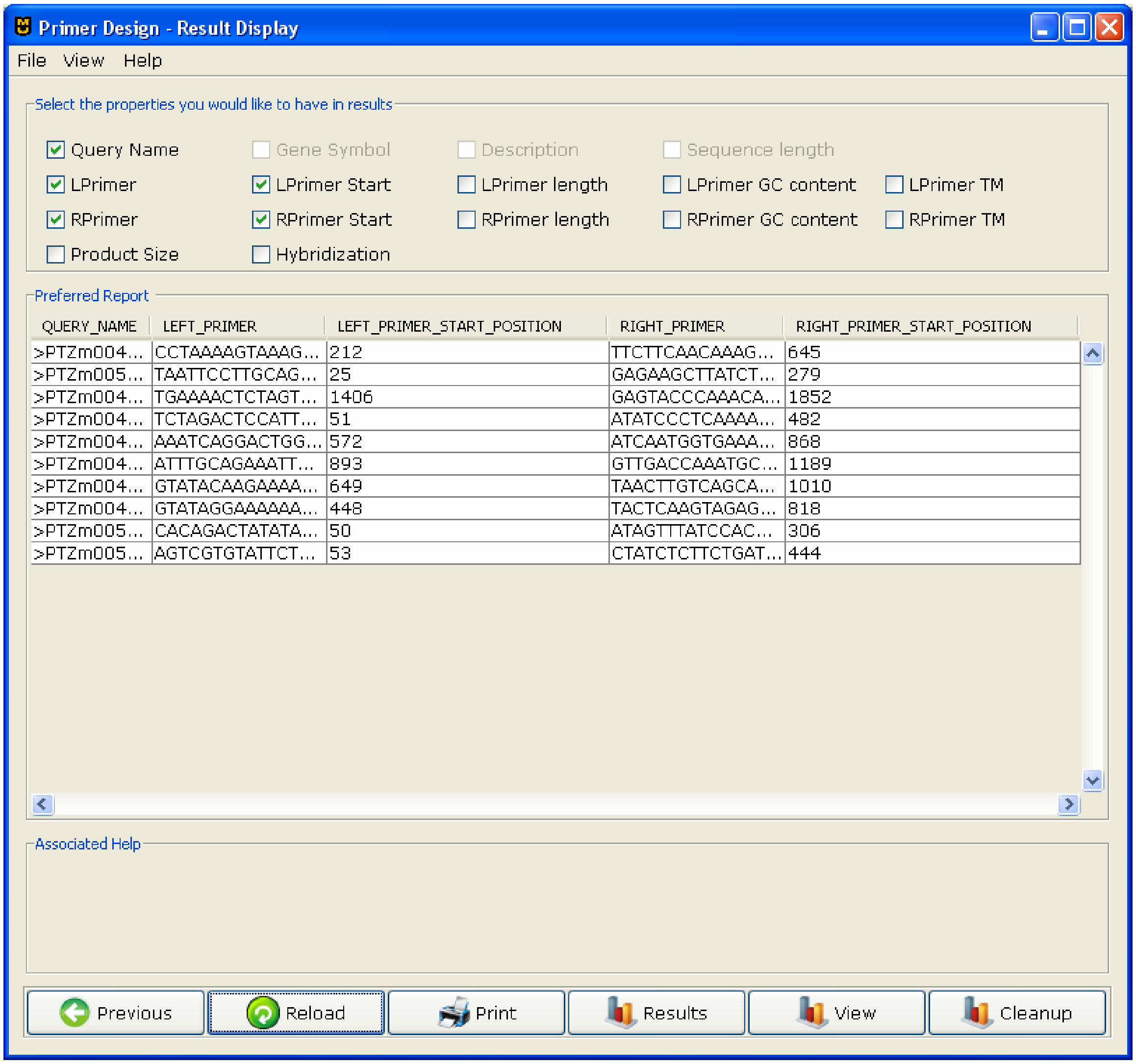
QUERY_NAME	LEFT_PRIMER	LEFT_PRIMER_START_POSITION	LEFT_PRIMER_LENGTH	LEFT_PRIMER_TM	LEFT_PRIMER_GC_CO
>PTZm00477...	CCTAAAAGTA...	212	24	58.342000	33.333000
>PTZm00513...	TAATTCCTTG...	25	24	58.883000	37.500000
>PTZm00467...	TGAAAACTCT...	1406	24	58.041000	33.333000
>PTZm00468...	TCTAGACTCC...	51	24	57.951000	37.500000
>PTZm00469...	AAATCAGGAC...	572	24	58.505000	33.333000
>PTZm00470...	ATTTGCAGAA...	893	24	58.057000	37.500000
>PTZm00472...	GTATACAAGA...	649	23	58.249000	39.130000
>PTZm00471...	GTATAGGAAA...	448	24	57.780000	33.333000
>PTZm00518...	CACAGACTAT...	50	26	58.129000	42.308000
>PTZm00519...	AGTCGTGTAT...	53	24	59.574000	45.833000

```
TCGTCGCTAGAGCTGAACTTCTTGAGCTCCCTAGCCCAGGTCCATCAGAAGCGCTGTATAAGGACAACAACTTCCTTTGAT
CTTCAAAATCAGGACTGGTTAAATTGATGGAGAGATTCCTCATCAACAACAGTGCCGTCTCAACGTCAATTGGATTCATCT
CCTTAAGTTTCTGCAGATGACCACTTGTACTTGGATCGAAAACACTACCTAAGAAGCTGTACACTCCCGCAAAATCTGGCA
TAAGGCGGAGTGACGGAATTTGATTTATTGGATCATTTGTTTCACCAGGTTGCCAAGTTCTTGGAAGGCCTTCAGTACTAC
TTGAGCAAAAATTGTTTGAACCTGGTGTTCCAGCACCTAAATCTTCTTTCACCATTGATGAAGCAACAGCTTGTGGGATGG
AACTATGAGCCCATGAAGAAACAGGCACGTTCATGCCAGAGTTCCTATGGATGGAAGATGAATCATTCTTGGAAGCCTTCT
GAGGATATGGATGGGCAGCTTTTCGCTTTGGTCGTGGAGGTGGGACATGTTCGCTGGTTCCATTTTTCTGAACCTTCAAAA
GGTACTTCTGTGCATGGCTCCTAATCTGGATGACAGTCTTGGAGCCAACAAAAGCCTCTATCTTCTTCCAGTCACGATCAA
AGAGCTGCAGGGCCTCGAGGAACTTGTCGTGCTCCTGCTCCGTCCAGCTCTCCCGGGACTTGGTGATTGTGTACGGCTTCC
```

Sequence marked in Magenta is best left primer and sequence marked in Cyan is best right primer

Figure 7: Screenshot showing ResultFrame which displays a list of best primer pairs for query sequences entered by user. It also gives options to view alignment, sequence, top 20 best primer pairs for a selected sequence, homologs and products.

Results can be printed by clicking "Print" button. Next screen can be seen by clicking "View" button. It shows the result excel file in tabular format. Here user can choose what fields he wants in the display. User can select the desired field from listed checkboxes and click "Refresh" to regenerate the table. Regenerated table can also be printed.



Select the properties you would like to have in results

☑ Query Name □ Gene Symbol □ Description □ Sequence length

☑ LPrimer ☑ LPrimer Start □ LPrimer length □ LPrimer GC content □ LPrimer TM

☑ RPrimer ☑ RPrimer Start □ RPrimer length □ RPrimer GC content □ RPrimer TM

□ Product Size □ Hybridization

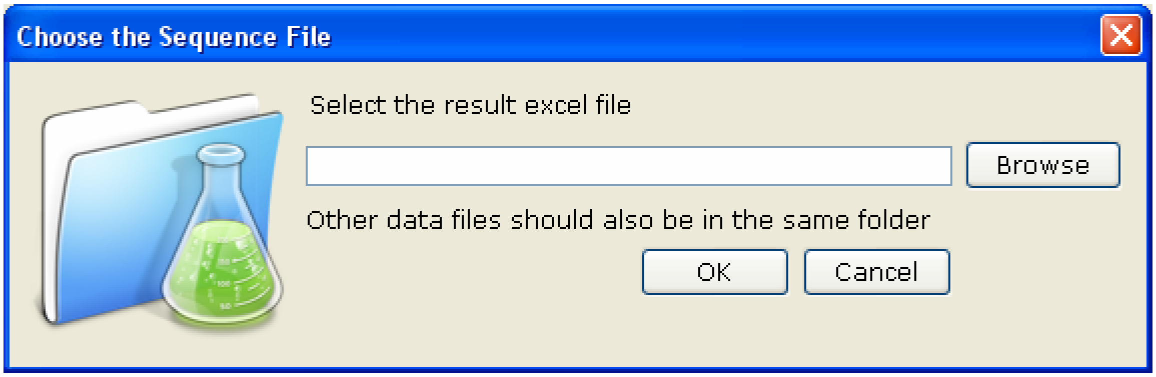
Preferred Report

QUERY_NAME	LEFT_PRIMER	LEFT_PRIMER_START_POSITION	RIGHT_PRIMER	RIGHT_PRIMER_START_POSITION
>PTZm004...	CCTAAAAGTAAAG...	212	TTCTTCAACAAAG...	645
>PTZm005...	TAATTCCTTGCAG...	25	GAGAAGCTTATCT...	279
>PTZm004...	TGAAAACTCTAGT...	1406	GAGTACCCAAACA...	1852
>PTZm004...	TCTAGACTCCATT...	51	ATATCCCTCAAAA...	482
>PTZm004...	AAATCAGGACTGG...	572	ATCAATGGTGAAA...	868
>PTZm004...	ATTTGCAGAAATT...	893	GTTGACCAAATGC...	1189
>PTZm004...	GTATACAAGAAAA...	649	TAACTTGTCAGCA...	1010
>PTZm004...	GTATAGGAAAAAA...	448	TACTCAAGTAGAG...	818
>PTZm005...	CACAGACTATATA...	50	ATAGTTTATCCAC...	306
>PTZm005...	AGTCGTGTATTCT...	53	CTATCTCTTCTGAT...	444

Associated Help

Figure 8: Screenshot showing ViewFrame. Here users can choose the properties to be displayed in result.

4.5.2. View results stored earlier: Saved results can also be revisited by pressing "ctrl+o" keys or selecting "File → Open Saved Results". On the dialog box, browse to the saved excel file and click OK button. This would load and enable the "Result" and "View" buttons on the screen. It is important to remember that all the generated result files are supposed to be in the same folder where excel file is located. System will generate errors if files are not found.



Select the result excel file

Other data files should also be in the same folder

Figure 9: Screenshot for OpenResult dialog box. Users can view previously saved result and reedit it.

Once the results are seen, and saved in desired location for later use, “Cleanup” action should be run to clean the temporary output files generated during runtime. If this is not done regularly, system would keep on accumulating unwanted garbage and occupy disk space.

CHAPTER 5. A COMBINED PROCESS AND OBJECT ORIENTED SERVICE BASED FRAMEWORK

5.1. Swing based architecture

In my application I have used Java Swing for program development. Swing offers much functionality and the greatest advantage it bears is the platform independent support of Java. Swing provides a richer set of components than AWT. Swing provides JTable and JTree components and added functionality to AWT-replacement components. Its components can change appearance based on the "look-n-feel" library used in current work. Swing components follow Model-View-Controller (MVC) paradigm and thus provide much more flexible user interface. It provides extra facilities such as icons on many components, tool-tip-text, decorative borders etc. Its components are lightweight thus less resource intensive than AWT. Swing provides built-in double buffering (which means an off-screen buffer [image] is used during drawing and then the resulting bits are copied onto the screen. The resulting image is smoother, less flicker and quicker than drawing directly on the screen).

5.1.1. Swing Action Architecture: The Swing Action architecture is used to implement shared behavior between two or more user interface components. For example, the menu items and the tool bar buttons will be performing the same action no matter which one is clicked. Another distinct advantage of using actions is that when an action is disabled, all the components, which use this Action, become disabled. The javax.swing.Action interface extends the

ActionListener interface and is an abstraction of a command that does not have an explicit UI component bound to it. The Action architecture is an implementation of a command design pattern. This is a powerful design pattern because it allows the separation of controller logic of an application from its visual representation. This allows the application to be easily configured to use different UI elements without having to re-write the control or call-back logic.

Defining action classes:

```
class FileAction extends AbstractAction {

    FileAction(String name) {        //Constructor
        super(name);
    }

    public void actionPerformed(ActionEvent ae){
            //add action logic here
    }
}
```

To add an action to a menu bar:

```
FileAction newAction = new FileAction("New");
JMenuItem item = fileMenu.add(newAction);
item.setAccelarator(KeyStroke.getKeyStroke('N',
Event.CTRL_MASK));
```

To add action to a toolbar

```
private JToolBar toolbar = new JToolBar();
toolbar.add(newAction);
```

So, an action object is a listener as well as an action.

5.2. Model View Controller (MVC) Design Pattern in Swing

MVC stands for Model View Controller architecture. Swing “J” components (e.g. JTable, JList, JTree etc) use a modified version of MVC. MVC separates a model (or data source) from a presentation and the logic that manages it.

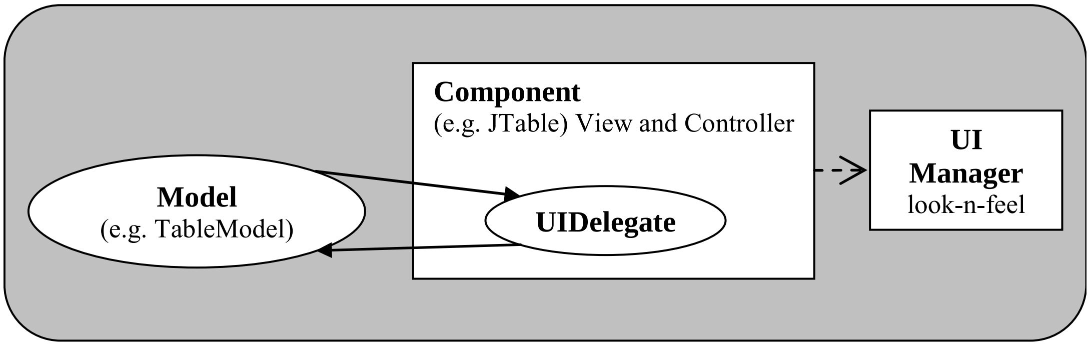


Figure 10: Model View Controller (MVC) design pattern followed in Java Swing System Architecture

- Component (e.g. JTable, JTree, and JList): coordinates actions of model and the UI delegate. Each generic component class handles its own individual view-and-controller responsibilities.
- Model (e.g. TableModel): charged with storing the data.
- UIDelegate: responsible for getting the data from model and rendering it to screen. It delegates any look-and-feel aspect of the component to the UI Manager.

Swing components do not control the actual laying out on the user screen. They delegate the functionality to layout managers. The layout managers make use of the strategy design pattern, which encapsulates family of algorithms for laying out components in the containers. If a particular layout algorithm is required other than the default algorithm, an appropriate layout manager can be instantiated and plugged into the container. For example, panels by default use the FlowLayout but it can be changed by executing:

```
panel.setLayout(new GridLayout(4,5));
```

This enables the layout algorithms to vary independently from the containers that use them. This is one of the key benefits of the strategy pattern.

5.3. Top level architecture:

There are two locations that have been recognized as suitable for third party tools plugin. First is the primer design program, which is at present Primer3 and the other location is alignment program, which are BLAST and MEGABLAST at present. Overall functionalities of PRIMEGENSv2 application is entirely separate from the plugged in programs. Each plugged in program is given a devoted module which takes care of its system and environment needs, and helps the plugged in program to interact with platform. It is like a device driver program which acts as an interface between operating system and hardware.

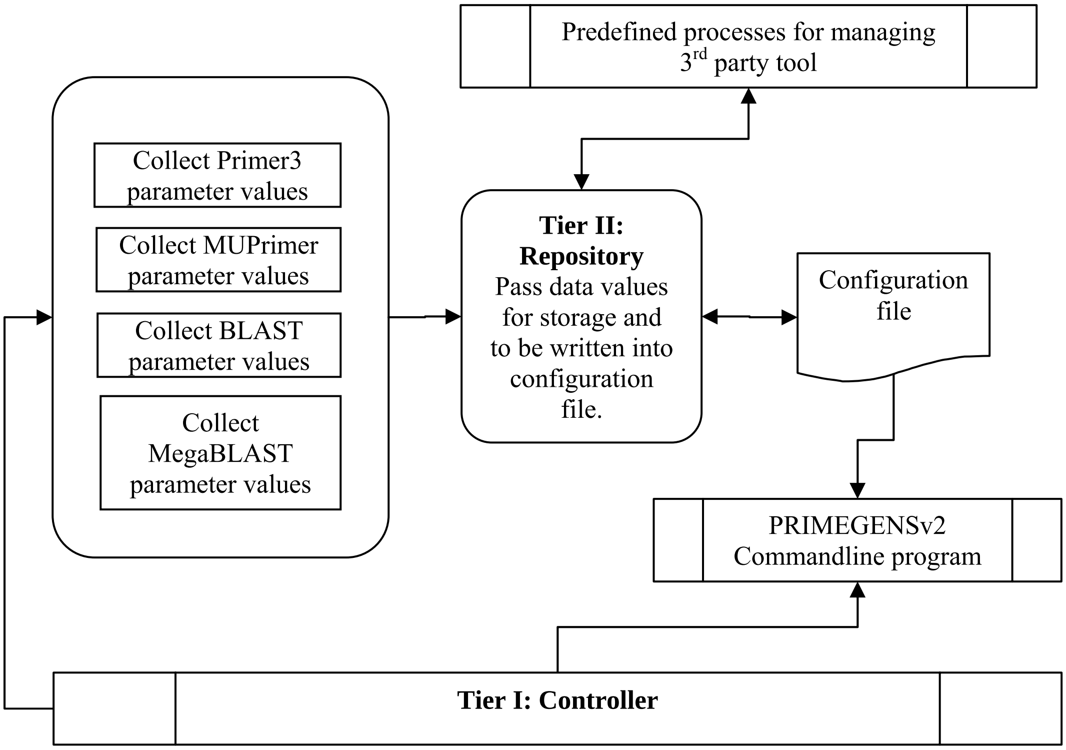

Predefined processes for managing 3rd party tool
Collect Primer3 parameter values
Collect MUPrimer parameter values
Collect BLAST parameter values
Collect MegaBLAST parameter values
Tier II: Repository
Pass data values for storage and to be written into configuration file.
Configuration file
PRIMEGENSv2 Commandline program
Tier I: Controller

Figure 11: Diagram showing interaction between platform modules and plugged in tools.

Third party tools communicate with PRIMEGENS commandline program with the help of a special configuration file. PRIMEGENS commandline reads parameter values from this file. All the required parameters for any particular program is set a value, either default or user supplied and written into configuration file. PRIMEGENS commandline then reads this file and while running that application uses its set of parameter values.

5.4. Structure of the platform developed

Basic motivation behind development of this application is the separation of individual services. This platform follows 3-tier architecture. First tier is the Controller level which governs the whole application. This tier is responsible for creating conducive environment for the application to run. Second tier is responsible for display. It can be regarded as the interface of application. It accepts user input and serves generated results on the UI. Third tier acts as a repository which supplies data required by each module. This level also includes several modules delivering support services to repository.

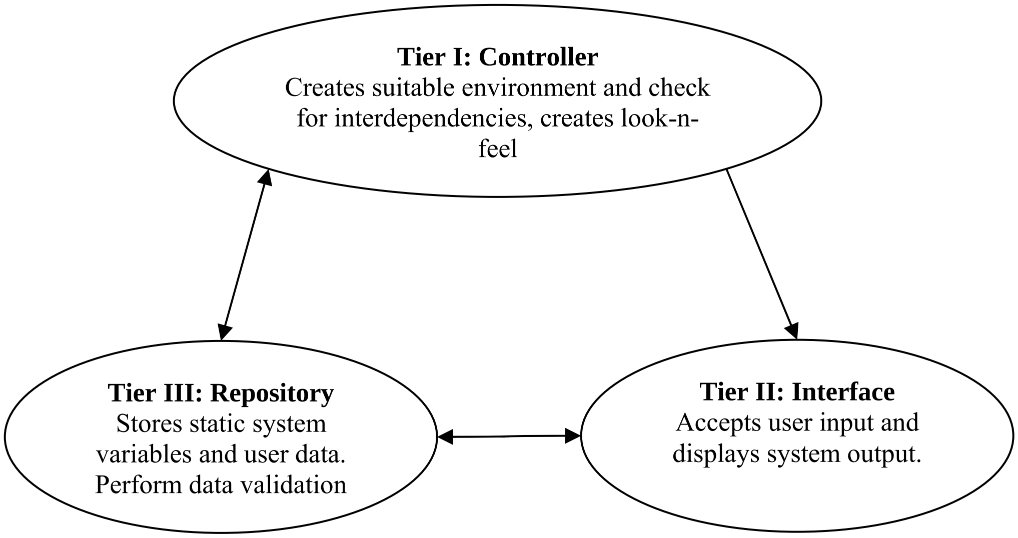

Tier I: Controller
Creates suitable environment and check for interdependencies, creates look-n-feel
Tier III: Repository
Stores static system variables and user data. Perform data validation
Tier II: Interface
Accepts user input and displays system output.

Figure 12: Three tier architecture designed in PRIMEGENSv2 GUI application. Such a partition helps in defining and classifying individual tasks or services and maintains OOP concepts of abstraction and encapsulation.

5.4.1. **Tier I: Controller:** This is the guardian section of application which acts as monitory and regulatory authority. Once the application is started, Controller adopts the **"Windows LookAndFeel"** loads its classes into memory.

```
setLNF();
javax.swing.UIManager.setLookAndFeel(javax.swing.UIMana
ger.getSystemLookAndFeelClassName());
```

It then instantiates Interface classes and Repository Static class names **Input**. It initializes system variables to default values through Repository and checks for environment variables. If an error is generated, it gives error and exits else it loads Interface and waits for user input.

```
try {
        Input.initSystemVar();
} catch (IOException ioe) {
        message = ioe.getMessage();
}
designPrimer = new DesignPrimer(message);
designPrimer.listenFrame();
```

Class **DesignPrimer(String message)** acts as the starting point. If an error message is generated, it is passed to Tier II interface to be displayed. This class controls which frame from the Interface tier should be displayed and also sets the current view of a particular frame based on current values of dependent variables. **ListenFrame()** method listens to frames and based on **FrameNo** variable value, decides which frame to display next. Based on the request source, it also decides what should be further navigation rule of the application e.g. based on the conditions, whether result frame is being viewed by immediately generated results or is being viewed by reloading previously stored results, it disables or enables option and execution frames.

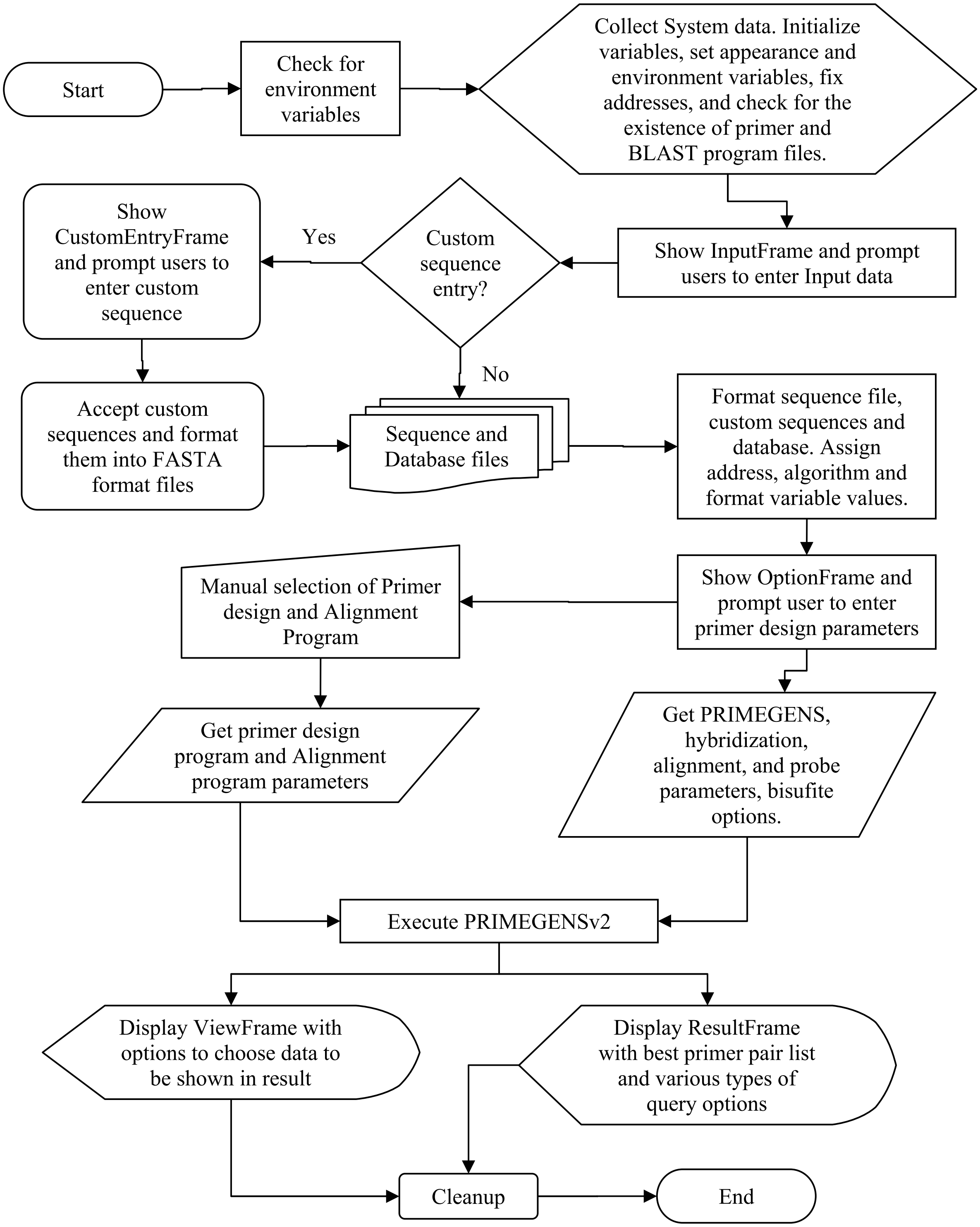

Start
Check for environment variables
Collect System data. Initialize variables, set appearance and environment variables, fix addresses, and check for the existence of primer and BLAST program files.
Show InputFrame and prompt users to enter Input data
Custom sequence entry?
Yes
Show CustomEntryFrame and prompt users to enter custom sequence
No
Accept custom sequences and format them into FASTA format files
Sequence and Database files
Format sequence file, custom sequences and database. Assign address, algorithm and format variable values.
Show OptionFrame and prompt user to enter primer design parameters
Manual selection of Primer design and Alignment Program
Get primer design program and Alignment program parameters
Get PRIMEGENS, hybridization, alignment, and probe parameters, bisufite options.
Execute PRIMEGENSv2
Display ViewFrame with options to choose data to be shown in result
Display ResultFrame with best primer pair list and various types of query options
Cleanup
End

Figure 13: Top level flowchart of algorithm followed in the application

Controller passes onto tier II: Interface which displays **InputFrame**. If query format selected in InputFrame is manual entry of custom sequence then it displays **ManualEntryFrame** and prompts user to enter nucleotide sequences. User supplied data is then passed to Tier III: Repository for FASTA format validation and storage after successful formatting. Controller then prompts tier II to display **OptionFrame** where user will enter program parameter values. The values are again supplied to tier III for validation and storage. Meanwhile if any unacceptable value is found, error messages generated from tier III are handled in a graceful manner. With user supplied and default values all stored, now is the time for execution to begin. Controller then creates three process threads namely, **execution thread**, **error thread** and **output thread**. Execution thread handles the commandline execution of the PRIMEGENSv2 program while output thread accepts runtime messages generated from PRIMEGENSv2 and passes them onto the **ExecFrame** of tier II i.e. Interface tier. Once execution is over, controller sets the system flags and informs result display modules of Interface about generated results. When user is done and tries to close the application, controller cleans up temporary files and output files from default location through **cleanup()** method and exits the application. This method can also be used to reset application to its start state. This method is useful in resetting application state but it may sometimes be harmful as user files generated in the default folder are also deleted. If users have not saved their data, they might lose it.

5.4.2. Tier 2: Repository: This module performs the role of model in the MVC design pattern. It stores all the system variables, data values, environment flag settings, user preferences and other requirement dependent variables. It checks every data value for its validity and generates error or warning messages if a value is not accepted.

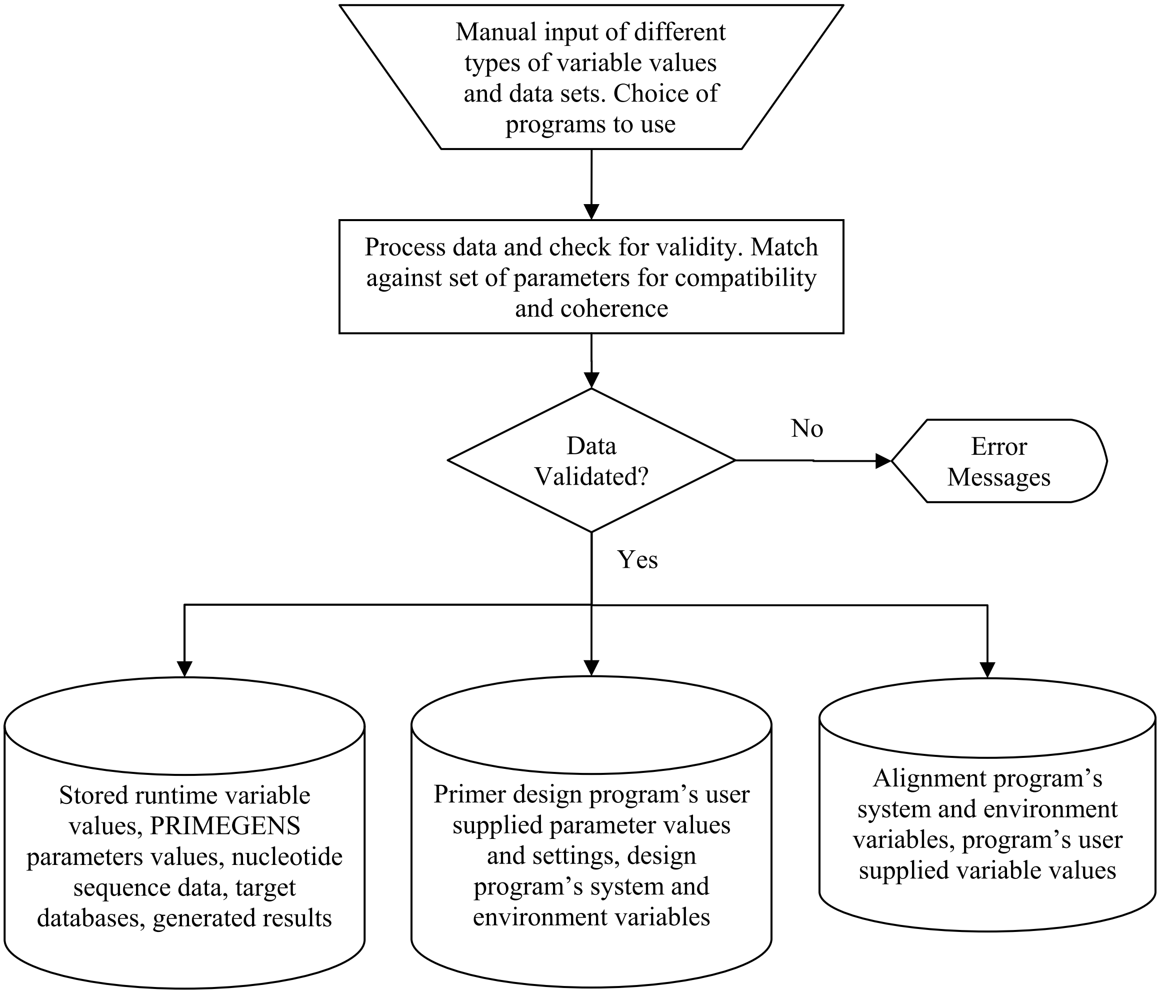

Manual input of different types of variable values and data sets. Choice of programs to use
Process data and check for validity. Match against set of parameters for compatibility and coherence
Data Validated?
No
Error Messages
Yes
Stored runtime variable values, PRIMEGENS parameters values, nucleotide sequence data, target databases, generated results
Primer design program's user supplied parameter values and settings, design program's system and environment variables
Alignment program's system and environment variables, program's user supplied variable values

Figure 14: This diagram shows functionality of Tier II: Repository in the application.

This tier with the help of several subroutines preprocesses user supplied sequences and databases. It checks for the FASTA format and if user wants it can convert supplied sequence file into FASTA format.

5.4.3. Tier 3: Interface: This tier serves as the head of the application. It listens to user requirements and submits them to Repository. Again it receives response from repository tier and serves that back to user. This part takes care of the display portion of the application. It is not concerned with how user supplied data is treated or what kind of data is required for what values, it just accepts user values and passes them onto tier 2. Interface tier comprises of three packages **gui.frames**, **gui.dialog** and **gui.extra.** The **gui.frames** package contains six JFrame classes in which flow of control goes as shown in fig. 15 Input frame accepts transfer

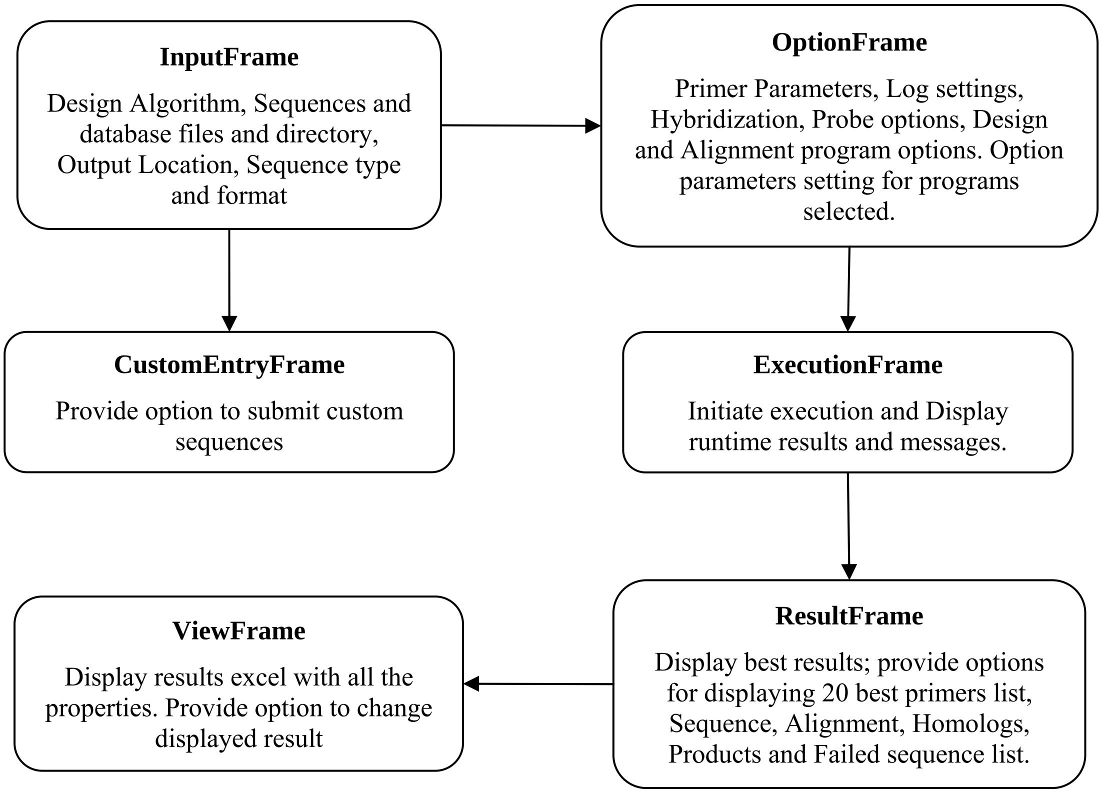

InputFrame
Design Algorithm, Sequences and database files and directory, Output Location, Sequence type and format
OptionFrame
Primer Parameters, Log settings, Hybridization, Probe options, Design and Alignment program options. Option parameters setting for programs selected.
CustomEntryFrame
Provide option to submit custom sequences
ExecutionFrame
Initiate execution and Display runtime results and messages.
ViewFrame
Display results excel with all the properties. Provide option to change displayed result
ResultFrame
Display best results; provide options for displaying 20 best primers list, Sequence, Alignment, Homologs, Products and Failed sequence list.

Figure 15: Flowchart for the PRIMEGENSv2 GUI. This diagram represents the functionality of Tier II of the architecture

Most of the functionalities of this tier have already been explained in previous chapter. **gui.frames** package uses **gui.doalog** package for intermediate help or to accept user input for third party program parameters. g**ui.extra** package contains icons and images.

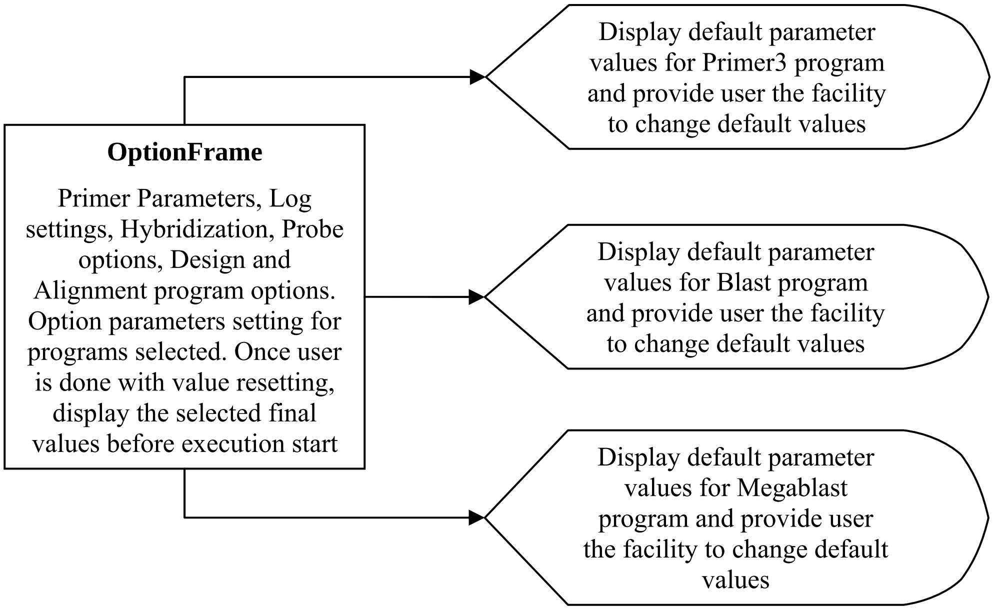

OptionFrame
Primer Parameters, Log settings, Hybridization, Probe options, Design and Alignment program options. Option parameters setting for programs selected. Once user is done with value resetting, display the selected final values before execution start
Display default parameter values for Primer3 program and provide user the facility to change default values
Display default parameter values for Blast program and provide user the facility to change default values
Display default parameter values for Megablast program and provide user the facility to change default values

Figure 16: Diagram showing interaction of OptionFrame and associated dialog boxes.

CHAPTER 6. RESULTS AND FUTURE WORK

6.1. Profiling of the PRIMEGENS application

Complete application was run and tested using NetBeans IDE 6.1 and used Java platform JDK 1.6. on J2SE v 1.4.0. Java foundation classes JFC v 1.1 with Swing 1.1 was used to program the application. Memory usage profiling of whole application was done using NetBeans which include following steps: Open the application, Submit sequence file and database files, set output directory, select Primer design program, accept default parameter values for primer design program, select blast program and accept default parameters, execute, view results, select properties to display, view sequences, primers, 20 best primer list, print result and in the end cleanup. Fig 17 shows the result of total allocated and used heap memory for the application in the run.

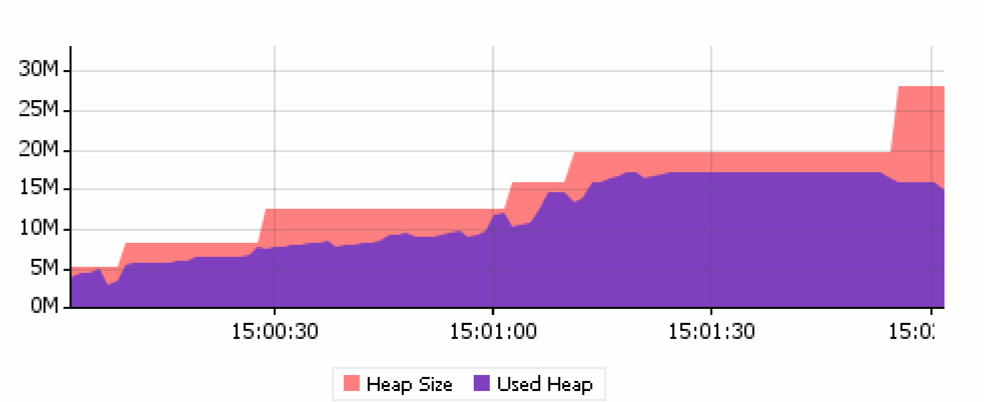


Figure 17: Snapshot of heap memory allocated and used in the application. Application used a maximum of 17MB of main memory

Maximum used memory at any time is 17,789,912 bytes which is approximately 17MB of main memory. Loaded classes in the memory count went to a maximum of 3769 and maximum number of threads at any point of time was 11 out of which 8 threads were system generated while 3 threads were created as Execution thread, Output thread and Error thread during execution process.

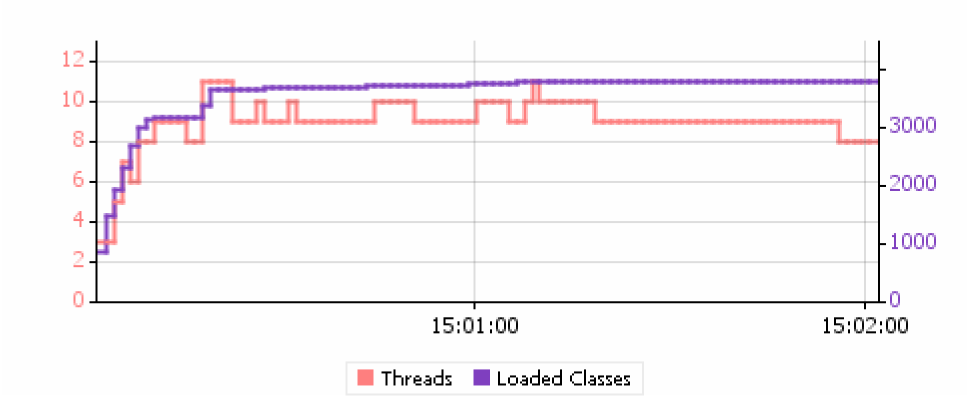
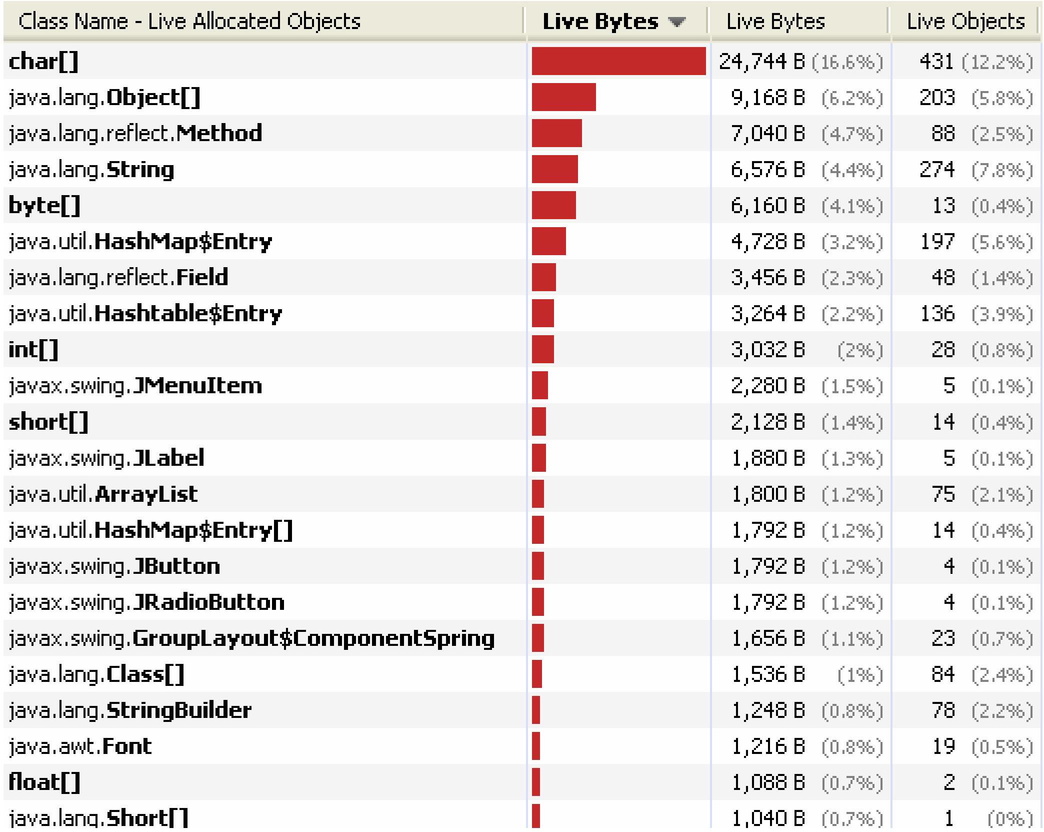


Figure 18: Graph showing number of threads created and classes loaded by the application. Out of maximum 11 threads 3 are helper threads created for execution. Application loads a maximum of 3769 classes in the memory.

Top 20 classes which used the most memory are sorted in order of decreasing memory usage in fig 19.



Class Name - Live Allocated Objects	Live Bytes ▼	Live Bytes	Live Objects
char[]		24,744 B (16.6%)	431 (12.2%)
java.lang.**Object[]**		9,168 B (6.2%)	203 (5.8%)
java.lang.reflect.**Method**		7,040 B (4.7%)	88 (2.5%)
java.lang.**String**		6,576 B (4.4%)	274 (7.8%)
byte[]		6,160 B (4.1%)	13 (0.4%)
java.util.**HashMap$Entry**		4,728 B (3.2%)	197 (5.6%)
java.lang.reflect.**Field**		3,456 B (2.3%)	48 (1.4%)
java.util.**Hashtable$Entry**		3,264 B (2.2%)	136 (3.9%)
int[]		3,032 B (2%)	28 (0.8%)
javax.swing.**JMenuItem**		2,280 B (1.5%)	5 (0.1%)
short[]		2,128 B (1.4%)	14 (0.4%)
javax.swing.**JLabel**		1,880 B (1.3%)	5 (0.1%)
java.util.**ArrayList**		1,800 B (1.2%)	75 (2.1%)
java.util.**HashMap$Entry[]**		1,792 B (1.2%)	14 (0.4%)
javax.swing.**JButton**		1,792 B (1.2%)	4 (0.1%)
javax.swing.**JRadioButton**		1,792 B (1.2%)	4 (0.1%)
javax.swing.**GroupLayout$ComponentSpring**		1,656 B (1.1%)	23 (0.7%)
java.lang.**Class[]**		1,536 B (1%)	84 (2.4%)
java.lang.**StringBuilder**		1,248 B (0.8%)	78 (2.2%)
java.awt.**Font**		1,216 B (0.8%)	19 (0.5%)
float[]		1,088 B (0.7%)	2 (0.1%)
java.lang.**Short[]**		1,040 B (0.7%)	1 (0%)

Figure 19: Table shows amount of main memory occupied by top 22 classes in application at some point of time.

6.2. Result and data generated by application

I did a comparison between the results generated by PRIMEGENS commandline program and PRIMEGENSv2 application. The results were same but in different formats. As ViewFrame screen of this system allows user to select parameters they would like to see in result, so this application displays and stores only selected parameters for the results excel. Users have the option to revise their selection as many number of times they want. The **config.txt** in Appendix B shows, all values are accepted and written by this application. This file is passed to the PRIMEGENS commandline executable. Generated results were also tested for their similarity in results and format. It was found to be 100% similar. Results displayed over screen were also matched with files and they were same.

6.3. Overview of the application

To support and run computationally intensive programs like Primer3, Megablast/Blast and PRIMEGENS commandline, it is expected that the platform on which these programs run should be very fast and should consume as less resources as possible. As we have seen in the profiling of this application, it takes very little resources and is extremely fast. As Java runtime automatically does garbage collection, memory management is efficiently handled in this program. Through its single point entry interface of configuration file, the developed structure supports any number of tools for a specific purpose to be plugged in the system. While maintaining services based functionality, it still retains abstraction in different modules. Every module has its own properties and methods and least interaction in terms of functioning with any other module. Java language specifications are easy to use and very developer friendly. Designer of third party tools can easily develop plugins for this system and can use it to compare results generated from other products.

The associated help bar on almost every panel helps user to properly understand meaning of technical terms and parameters usage for the application. There are tool tip text provided for every text box and labels which let user recognize the variable. Any returning user also has the facility to revise its stored result data and save it for future use. This way the target set for the current work has been achieved.

6.4. Future work in this area

There are tremendous opportunities associated with this application and needs to be developed further. The foremost need is to plug in few more third party tools so that extensive use of this system can be observed. As more and more users would use this system, different types of queries associated with results would generate which should be implemented into the system.

Our laboratory is working towards enhancing the result section in which users would be able see results in more elaborate ways. We are working to implement a facility in the system so that generated primers can again be blasted against a database. This would help a user check if instead of best primer chosen by this application they can find another primer suitable for their use.

Another improvement in this system was suggested by the research degree committee member Dr. Michael Wang. He suggested retrieving gene related information from internationally acclaimed databases so that a user should be able to fetch and see the latest developments on the gene he/she is working with. This can be done using SOAP and sending a relevant query string to the concerned web resource.

A simpler web implementation of this application is under process which uses similar design pattern. The initial design is to allow user to submit their job on the server maintained by Digital Biology Laboratory. The process of data submission would be carried out using a very similar interface, but with little fewer parameters. This website will provide a list of organism gene databases to select from. Users would have to upload their sequence files and submit their job for execution. When finished, PRIMEGENS web system will send an email to the user's account informing them of the generated results. This result will be kept on the server for a limited period of time after which it can be deleted. If users provide consent, their results and data can be uploaded to the website's database. It would soon be implemented and then users from all over the world would be able to access its features. This way more requirements will arise and based on the users' feedback, this system can be made more effective.

Appendix A: Specifications of FASTA format

FASTA format description

A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol in the first column. It is recommended that all lines of text be shorter than 80 characters in length. An example sequence in FASTA format is:

```
>gi|532319|pir|TVFV2E|TVFV2E envelope protein
ELRLRYCAPAGFALLKCNDADYDGFKTNCSNVSVVHCTNLMNTTVTTGLLLNG
SYSENRT
QIWQKHRTSNDSALILLNKHYNLTVTCKRPGNKTVLPVTIMAGLVFHSQKYNLR
LRQAWC
HFPSNWKGAWKEVKEEIVNLPKERYRGTNDPKRIFFQRQWGDPETANLWFNCH
GEFFYCK
MDWFLNYLNNLTVDADHNECKNTSGTKSGNKRAPGPCVQRTYVACHIRSVIIW
LETISKK
TYAPPREGHLECTSTVTGMTVELNYIPKNRTNVTLSPQIESIWAAELDRYKLVEIT
PIGF
APTEVRRYTGGHERQKRVPFVXXXXXXXXXXXXXXXXXXXXXXVQSQHLLAG
ILQQQKNL
LAAVEAQQQMLKLTIWGVK
```

Sequences are expected to be represented in the standard IUB/IUPAC nucleic acid codes, with these exceptions: lower-case letters are accepted and are mapped into upper-case; a single hyphen or dash can be used to represent a gap of indeterminate length; and in amino acid sequences, U and * are acceptable letters (see below). Before submitting a request, any numerical digits in the query sequence should either be removed or replaced by appropriate letter codes (e.g., N for unknown nucleic acid residue).

The nucleic acid codes supported are:

A --> adenosine
C --> cytidine
G --> guanine
T --> thymidine
U --> uridine
R --> G A (purine)
Y --> T C (pyrimidine)
K --> G T (keto)

M --> A C (amino)
S --> G C (strong)
W --> A T (weak)
B --> G T C
D --> G A T
H --> A C T
V --> G C A
N --> A G C T (any)
- gap of indeterminate length

Appendix B: Configuration File sample format

```
#
#                                   PRIMEGENSv2.0.5
#
#Copyright/Availability:
#The program PRIMEGENS is copyrighted@Digital Biology Laboratory, University of
#Missouri and is NOT in the public domain. PRIMEGENS is free of charge to academic
#users and can be downloaded from http://www.digbio.missouri.edu/primegens.
#
#PRIMEGENS uses some results from two third-party programs, i.e., Primer3
#developed by the Whitehead Institute for Biomedical Research and BLAST developed
#by National Center for Biotechnology Information (NCBI).
#
#The authors of PRIMEGENS make no representations about the suitability of the
#software for any purpose. It is provided "as is" without express or implied
#warranty. The authors shall not be liable for any damages suffered by Licensee
#from the use of this software.
#
#Acknowledgment:
#The development of PRIMEGENS was sponsored by the Office of Biological and
#Environmental Research, U.S. Department of Energy, under Contract
#DE-AC05-00OR22725, managed by UT-Battelle, LLC. Dong Xu, Gary Li, Liyou Wu,
#Jizhong Zhou, and Ying Xu at Oak Ridge National Laboratory are involved in this
#project. We would like to thank our colleagues Manesh Shah, Joel A. Klappenbach,
#and Julia Stair for their help and insightful comments.
#
#Citation:
#If you use PRIMEGENS in any publication, please refer to:
#Gyan Prakash Srivastava and Dong Xu "Genome-Scale Probe and Primer Design with
#PRIMEGENS" Methods Mol Biol. 2007; 402:159-7617951795.
#
#
#copyright@Digital Biology Laboratory
#Modified@ Sepetember, 2008
#version: 2.0.5
#
#standard tags used in primer design
#it will allow PRIMEGENS to recognize various input formats and parameters
#
#
#
#select algorithm type for primer design. currenly four types of algorithm
#design is available that is PSPD, FSPD, GSPD and SSPD.
```

```
#SSPD(Sequence-Specific Primer Design) (default)
#       The algorithm is used to design primer for user input query
#       sequences in FASTA format. It will design primer for each sequenc and will
#       check for hybridization for each pair.
#
#FSPD(Fragment-Specific Primer Design)
#       The algorithm would generate the fragment for each of the long query
#       sequences based on fragment length (FSPD_LENGTH) and inter-distance
#       between two fragments(FSPD_PERIOD).
#
#GSPD(Gene-Specific Primer Design)
#       The algorithm would use a list of input gene symbols to find respective
#       nucleotide sequences from the database and generate primers. Algorithm
#       assumes every gene symbols is present in the database.
#
#PSPD(Probe-Specific Primer Design)
#       The algorithm would first find the input unique fragment (probe) for a
#       sequence amongst its homologs using multiple sequence alignment technique
#       and generate primer for each sequence.
#
#ALGORITHM_TYPE=1                 #Sequnce-specific primer design
#ALGORITHM_TYPE=2                 #Fragment-specific primer design
#ALGORITHM_TYPE=3                 #Gene-specific primer design
#ALGORITHM_TYPE=4                 #Probe-specific primer design

ALGORITHM_TYPE=1                  #Sequnce-specific primer design

###########QUERY SEQUENCE INPUT FORMAT TYPE####################
#This tag represents title of the input query sequence. Program would parse
#this title to fetch the coordinate of the sequence in case of chromosome type
#database. Title follows the following format
#chr22:10900-109500 "description" Incase parsing the title.
#
#QUERY_FORMAT=1                          #FASTA:gid
#QUERY_FORMAT=2                          #FASTA:gid,description
#QUERY_FORMAT=3                          #FASTA:chr,start,end
#QUERY_FORMAT=4                          #FASTA:chr,start,end,desciption
#
QUERY_FORMAT=1                           #FASTA:gid
#
#This tag represents the presence/absence of nucleotide sequence in a query
#and must follow same rule for each of them. If nucleotide sequence is not
#present, program would either fetch the sequence from database or genome by
#parsing the sequence title.
#
```

```
#QUERY_SEQUENCE=0                #only query name present in query input
#QUERY_SEQUENCE=1                #nucleotide sequence present in query input
#
QUERY_SEQUENCE=1                 #nucleotide sequence present in query input
#
#
##############DATABASE
TYPE##############################################
#if database is in a single file, which contains multiple sequences,
#whether small or large, the database is classified as a single database.
#
#if database is genome of any species such that, each chromosome sequence
#is contained in one file like downloaded human genome then database type
#is classified as chromosome type.
#
#if there are multiple database, each file contains multiple sequences in
#FASTA format, then these database types are classified as multiple type.
#To Use this option, user  to generate a file containg list of names for
#each database file.
#
#DATABASE_TYPE=1                 #single database file
#DATABASE_TYPE=2                 #multiple databases
#DATABASE_TYPE=3                 #chromosomal sequences of organism
#
DATABASE_TYPE=1                       #single database file
#
###################PRIMEGENS
CONFIGURATION###############################
#Bisulfite mode primer design is for methylated type sequence primer design.
#Value of 1 indicates bisulfite mode primer design. If a value of 1 is
#entered then PRIMER_CG_MAX tag should have some value (default=0).
#BISULFITE_MODE=0                #normal primer design
#BISULFITE_MODE=1                #bisulfite mode primer design
#
BISULFITE_MODE=0                 #normal primer design
#
#bisulfite primer design parameters
#max no. of CG sites in either primer (ignored if BISULFITE_MODE=0)
#PRIMER_CG_MAX=?                 #valid only for bisulfite mode
#PRIMER_CG_MAX=0                 #no CG allowed in the primer pair
#PRIMER_CG_MAX=2                 #max no. of CG allowed in primer pair
#
PRIMER_CG_MAX=?                       #valid only for bisulfite mode
#
#probe design
```

```
#PROBE_DESIGN=0                          #usual primer design
#PROBE_DESIGN=1                          #probe design only
#
PROBE_DESIGN=0
#
#GC content of the probes
#PROBE_MAX_GC=70.0                       #max. GC% in probe should be 70.0%
#PROBE_MIN_GC=50.0                       #min. GC% in probe should be 50.0%
#
PROBE_MAX_GC=70.0                        #max. GC% in probe should be 70.0%
PROBE_MIN_GC=50.0                        #min. GC% in probe should be 50.0%
#
#max probe blast hit into consideration. This will control the specificity of
#designed probes.
#PROBE_BLAST_HIT=2                       #max. blast hit for any probe
#
PROBE_BLAST_HIT=0                        #max. blast hit for any probe
#
#These parameters are used while generating probes for any partitioned query
#sequence. Using small value of PROBE_PERIOD would increase the runtime of the
#algorithm. Do not confuse it with the length of fragment of query sequence
#while partitioning it.
#PROBE_LENGTH=160                        #probe length should be 160 bases
#PROBE_PERIOD=20                         #overlap between any two probe should be
20 bases
#
PROBE_LENGTH=160                         #probe length should be 160 bases
PROBE_PERIOD=20                          #overlap between any two probe should be
20 bases
#
#if the program does not find probe in a region, it will jump to next region
#PROBE_REGION=1000                       #length of the fragment of partioned query
sequence
#
PROBE_REGION=500                         #length of the fragment of partioned query
sequence
#
#find only single probe per query sequence. if to find unique probes, then
#probe region is not considered.
#PROBE_UNIQUE=0                                 #fragment-wise sequence probe
design
#PROBE_UNIQUE=1                                 #find a unique probe for each input
query sequence
#
PROBE_UNIQUE=0                           #fragment-wise sequence probe design
```

```
#
#The size/description during display can be controled from following values.
#DISPLAY_LEVEL=1                         #used in JAVA based GUI execution
window
#DISPLAY_LEVEL=2                         #used in command line execution
#DISPLAY_LEVEL=3                         #detailed description of primer design
#DISPLAY_LEVEL=4                         #used for debugging problems during primer
design
#
DISPLAY_LEVEL=2                          #used in command line execution
#
#additional parameters specifically used for fragment-specific primer design
#FSPD_LENGTH=160                         #query fragment to design associated primer
pair
#FSPD_PERIOD=20                          #overlap between any two query fragment
#
FSPD_LENGTH=160                          #query fragment to design associated primer
pair
FSPD_PERIOD=20                           #overlap between any two query fragment
#
#selecte query sequence fragment to include max. number of cut sites.
#CUT_SITE_COUNT=3                        #count of cut sites to consider
#CUT_SITE=CCGC,GCGG,ACGT  #list of cut site nucleotides
#
CUT_SITE_COUNT=3                         #count of cut sites to consider
CUT_SITE=CCGC,GCGG,ACGT                  #list of cut site nucleotides
#
#total blast records
CHRCOUNT=48

#total blast record
BLASTRECORDS=48

#blast will run 24 times
BLASTLOOP=12

#database count in each blast loop
CHRDBINLOOP=4

#smallest possible amplicon
AMPSB=500

#oligo size start from right
MINBIND=18
```

```
#type of blast program used for hybridization check
FORMATDB_PROGRAM=formatdb
#BLAST_PROGRAM=blastall
#BLAST_PROGRAM=bl2seq
#BLAST_PROGRAM=megablast
BLAST_PROGRAM=megablast.exe

#right size oligo length to check hybridization
WORDSIZE=18

#type of primer design program used for primer design
#PRIMERDESIGN_PROGRAM=primer3_core
PRIMERDESIGN_PROGRAM=primer3.exe
#PRIMERDESIGN_PROGRAM=muprimer.exe

#type of alignment program used for multiple sequence alignment
#ALIGNMENT_PROGRAM=clustalW
#ALGINMENT_PROGRAM=bl2seq
#ALIGNMENT_PROGRAM=blast
#ALIGNMENT_PROGRAM=megablast
#ALGINMENT_PROGRAM=self

#LOGS=0
LOGS=1

#valid only for pspd primer design
#DESIGN_PRIMER=1
DESIGN_PRIMER=1

BLAST_OGP=4
BLAST_OIP=2
PSPD_EXPECT="1e-15"
PSPD_MINFILTERLENGTH=50
RUN_SIM_FLAG=1
MAX_SIMILARITY=0.75
MINSEGLENGTH=100

#probe design
#MIN_PROBE_LENGTH=150
#MAX_PROBE_LENGTH=170

#################PRIMER DESIGN
PARAMETERS############################
#primer design parameters used specifically by primer3
#primer3-parameter
```

```
PRIMER_PRODUCT_SIZE_RANGE=80-150

#primer3-parameter
PRIMER_EXPLAIN_FLAG=0

#primer3-parameter
PRIMER_FILE_FLAG=0

#primer3-parameter
#number of primer pairs design in one run by primer3
PRIMER_NUM_RETURN=25

#primer3-parameter
PRIMER_MAX_POLY_X=0

#primer3-parameter
PRIMER_INTERNAL_OLIGO_MAX_POLY_X=0

#primer3-parameter
PRIMER_MIN_TM=56.0

#primer3-parameter
PRIMER_OPT_TM=60.0

#primer3-parameter
PRIMER_MAX_TM=66.0

#primer3-parameter
PRIMER_MIN_SIZE=19

#primer3-parameter
PRIMER_OPT_SIZE=20

#primer3-parameter
PRIMER_MAX_SIZE=23

#primer3-parameter
PRIMER_MAX_GC=60

#primer3-parameter
PRIMER_MIN_GC=40

#primer3-parameter
#PRIMER_SELF_END=100
PRIMER_SELF_END=3
```

```
#primer3-parameter
#PRIMER_SELF_ANY=100
PRIMER_SELF_ANY=8

#primer3-parameter
PRIMER_MAX_END_STABILITY=100.0
```

REFERENCES

1. Arvidsson, S., et al., QuantPrime - a flexible tool for reliable high-throughput primer design for quantitative PCR. BMC Bioinformatics, 2008. 9: p. 465.

2. Morgenstern, B., DIALIGN: multiple DNA and protein sequence alignment at BiBiServ. Nucleic Acids Res, 2004. 32(Web Server issue): p. W33-6.

3. Thompson, J.D., T.J. Gibson, and D.G. Higgins, Multiple sequence alignment using ClustalW and ClustalX. Curr Protoc Bioinformatics, 2002. Chapter 2: p. Unit 2 3.

4. Xu, D., et al., PRIMEGENS: robust and efficient design of gene-specific probes for microarray analysis. Bioinformatics, 2002. 18(11): p. 1432-7.

5. Srivastava, G.P., et al., PRIMEGENS-v2: genome-wide primer design for analyzing DNA methylation patterns of CpG islands. Bioinformatics, 2008. 24(17): p. 1837-42.

6. Srivastava, G.P. and X. Dong, Genome-scale probe and primer design with PRIMEGENS. Methods in Molecular Biology, 2007. 402: p. 159-76.

7. Altschul, S.F., et al., Basic local alignment search tool. J Mol Biol., 1990. 215(3): p. 403-10.

8. Rozen, S. and H.J. Skaletsky, Primer3 on the WWW for general users and for biologist programmers. Krawetz S, Misener S (eds) Bioinformatics Methods and Protocols: Methods in Molecular Biology. Humana Press, Totowa, NJ, 2000: p. 365-386.

9. Bartlett, J.M.S. and D. Stirling, A Short History of the Polymerase Chain Reaction. Methods in Molecular Biology, 2003. 226: p. 3-6.

10. Burpo, F.J., A critical review of PCR primer design algorithms and crosshybridization case study. Biochemistry, 2001: p. 218.

11. Polymerase Chain Reaction (PCR). Laboratory for Microbial Ecology, Department of Earth, Ecological and Environmental Sciences, University of Toledo, 2004.

12. Q, H., et al., Primers are decisive for sensitivity of PCR. Biotechniques, 1994. 17(1): p. 82-87.

13. Dieffenbach, C.W., T.M. Lowe, and G.S. Dveksler, General concepts for PCR primer design. Genome Research, 1993. 3: p. S30-S37.

14. Wu, D., et al., The effect of temperature and oligonucleotide primer length on specificity and efficiency of amplification by the polymerase chain reaction. DNA Cell Biol., 1991. 10: p. 233-238.

15. Wallace, R., et al., Hybridization of synthetic oligodeoxyribonucleotides to phi chi 174 DNA: the effect of single base pair mismatch. Nucleic Acids Research, 1979. 6(11): p. 3543-57.

16. SantaLucia, J.J., H. Allawi, and P. Seneviratne, Improved nearest-neighbor parameters for predicting DNA duplex stability. Biochemistry. 1996 Mar 19;(11):, 1996. 35(11): p. 3555-62.

17. Untergasser, A., et al., Primer3Plus, an enhanced web interface to Primer3 Nucleic Acids Research, 2007. 35(Web Server Issue): p. W71-W74.

18. Wrobel, G., AutoPrime: selecting primers for expressed sequences. Genome Biology, 2004. 5: p. 11.

19. You, F.M., et al., BatchPrimer3: A high throughput web application for PCR and sequencing primer design. BMC Bioinformatics, 2008. 9: p. 253.

20. Pattyn, F., et al., RTPrimerDB: the real-time PCR primer and probe database, major update 2006. Nucleic Acids Research, 2006. 34(Database issue): p. D684-D688.

21. Wang, X. and B. Seed, A PCR primer bank for quantitative gene expression analysis. Nucleic Acids Research, 2003. 31(24): p. e154.

22. Cui, W., D.D. Taub, and K. Gardner, qPrimerDepot: a primer database for quantitative real time PCR. Nucleic Acids Research, 2006. 35(Database issue): p. D805–D809.

23. Han, S., AtRTPrimer: database for Arabidopsis genome-wide homogeneous and specific RT-PCR primer-pairs. BMC Bioinformatics, 2006. 7: p. 179.

24. Fredslund, J., DATFAP: A Database of Primers and Homology Alignments for Transcription Factors from 13 Plant Species. BMC Genomics, 2008. 9: p. 140.

25. Abd-Elsalam, K.A., Bioinformatic tools and guideline for PCR primer design. African Journal of Biotechnology, 2003. 2(5): p. 91-95.

26. Rose, T.M., J.G. Henikoff, and S. Henikoff, CODEHOP (COnsensus-DEgenerate Hybrid Oligonucleotide Primer) PCR primer design. Nucleic Acids Research, 2003. 31(13): p. 3763-6.

27. Giegerich, R., F. Meyer, and C. Schleiermacher, GeneFisher--software support for the detection of postulated genes. Proceedings of International Conference on Intelligent Systems for Molecular Biology., 1996. 4: p. 68-77.

28. Lamprecht, A.-L., et al., GeneFisher-P: variations of GeneFisher as processes in Bio-jETI. BMC Bioinformatics, 2008. 9(Suppl 4): p. S13.

29. Aranyi, T., et al., The BiSearch web server. BMC Bioinformatics, 2006. 7: p. 431.

30. Aranyi, T. and G.E. Tusnady, BiSearch: ePCR tool for native or bisulfite-treated genomic template. Methods Mol Biol, 2007. 402: p. 385-402.

31. Tusnady, G.E., et al., BiSearch: primer-design and search tool for PCR on bisulfite-treated genomes. Nucleic Acids Res, 2005. 33(1): p. e9.

32. Turchin, A. and J.J. Lawler, The Primer Generator: A Program that Facilitates the Selection of Oligonucleotides for Site-Directed Mutagenesis. Biotechniques., 1999. 26(4): p. 672-6.

33. Li, L.-C. and R. Dahiya, MethPrimer: designing primers for methylation PCRs. Bioinformatics, 2002. 18(11): p. 1427-1431.

34. Nielsen, H.B. and S. Knudsen, Avoiding cross hybridization by choosing nonredundant targets on cDNA arrays Bioinformatics, 2002. 18(2): p. 321-322.

35. Croce, O., F. Chevenet, and R. Christen, OligoHeatMap (OHM): an online tool to estimate and display hybridizations of oligonucleotides onto DNA sequences. Nucleic Acids Res, 2008. 36(Web Server issue): p. W154-6.

36. Fredslund, J. and M. Lange, Primique: automatic design of specific PCR primers for each sequence in a family. BMC Bioinformatics, 2007. 8: p. 369.

37. Plasterer, T.N., PRIMERSELECT. Primer and probe design. Methods Mol Biol, 1997. 70: p. 291-302.

38. Gordon, P.M. and C.W. Sensen, Osprey: a comprehensive tool employing novel methods for the design of oligonucleotides for DNA sequencing and microarrays. Nucleic Acids Res, 2004. 32(17): p. e133.

39. Lefever, S., et al., RTPrimerDB: the portal for real-time PCR primers and probes. Nucleic Acids Res, 2008.

40. Pattyn, F., et al., RTPrimerDB: the real-time PCR primer and probe database. Nucleic Acids Res, 2003. 31(1): p. 122-3.

41. Sullivan, R.T., C.B. Morehouse, and J.W. Thomas, Uprobe 2008: an online resource for universal overgo hybridization-based probe retrieval and design. Nucleic Acids Res, 2008. 36(Web Server issue): p. W149-53.

42. Kellner, W.A., et al., Uprobe: a genome-wide universal probe resource for comparative physical mapping in vertebrates. Genome Res, 2005. 15(1): p. 166-73.

43. Rouillard, J.M. and E. Gulari, OligoArrayDb: pangenomic oligonucleotide microarray probe sets database. Nucleic Acids Res, 2009. 37(Database issue): p. D938-41.