GenVista: Bioinformatics Pipeline for Identification of Candidate Gene Anchor Markers in Plants

By-

Abhijeet Kapoor (041508)
kapoor.abhijeet@gmail.com

Ashish Ranjan Srivastava(041523)
ranjan.chitranshi@gmail.com

juit
विद्या तत्व ज्योतिसमः
JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

MAY-2008

Submitted in partial fulfillment of the Degree of Bachelor of Technology

DEPARTMENT OF BIOTECHNOLOGY & BIOINFORMATICS
JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT, SOLAN, H.P. , INDIA.

CERTIFICATE

This is to certify that the work entitled, **"GenVista: Bioinformatics Pipeline for Identification of Candidate Gene Anchor Markers in Plants"** submitted by **Abhijeet Kapoor** (041508) and **Ashish Ranjan Srivastava** (041523) in partial fulfillment for the award of Degree Of Bachelor Of Technology in Bioinformatics of Jaypee University of Information Technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other university or institute for the award of this or any other degree or diploma.

28/05/08

Dr. R.S.Chauhan

(Project Coordinator)

ACKNOWLEDGMENT

We wish to express our gratitude to **Dr. R. S. Chauhan** for providing us invaluable guidance and suggestions. He was always there to listen and advice. He taught us to ask questions and express our ideas. He showed us different ways to approach a research problem and provide us motivation to accomplish the goal.

Our sincere thanks to Krishna Sir and Arti Ma`am for their timely suggestions and co-operation during the period of our project.

Abhijeet Kapoor

Abhijeet Kapoor

(041508)

Ashish Ranjan.

Ashish Ranjan Srivastava

(041523)

TABLE OF CONTENTS:

List of Figures

List of Abbreviations

1) ESTs – Expressed Sequence Tags
2) CDS – Coding Sequences
3) BLAST- Basic Local Alignment Search Tool
4) PCR- Polymerase Chain Reaction

ABSTRACT

GenVista is an endeavor to design a bioinformatics pipeline for the best possible use of available genome information to gain insight into the unexplored area of plants for which complete genome information is not available. The level of gene conservation is an important criterion in determining the extent to which comparative genomics can be applied across species. GenVista utilizes conserved regions across species to design primers that can be used to amplify intronic region in a segment. The user input set of ESTs or Coding sequences in FASTA format which are compared against a database of homologous coding sequences. The corresponding coding sequences identified in the database are then used for primer design. Primers are designed from conserved regions as they are under less evolutionary constraints than introns.

INTRODUCTION

The sequencing of the genome of a model species is just the beginning of a new venture to unravel genetic information and to gain better insight into the genetics of other species under investigation. The complete sequencing of the *Arabidopsis thaliana* genome has been regarded as a landmark in plant sciences. Over the last two decades, comparative genetics has shown that the organization of genes within plant genomes has remained conserved over the evolutionary periods. The level of gene conservation i.e. synteny in structure, function and chromosomal location is an important criterion in determining the extent to which comparative genomics can be applied across species. The overall purpose and goal of understanding gene conservation across different genomes is to make use of model genome resources for practical applications.

Molecular markers permitting cross-species mapping (anchor markers) along co-linear genomic regions are central to comparative genomics. Anchor markers are landmarks selected from the developed marker map, selection based on even distribution across the genome and the ability to cross hybridize with related species. They are usually RFLPs but can also be ESTs. They are only rarely SSRs but never AFLP or RAPDs.

To identify potential anchor marker sequences in plants, we have established an **Automated Bioinformatics Pipeline – GenVista** that combines multi-species coding sequence data. For this purpose, we have used the available coding sequences information from three different organisms (*Arabidopsis thaliana, Ricinus communis and Glycine max*).

Earlier, our study focused mainly on sequence data of seed ESTs and seed CDS from oilseed plants but once the base was setup for the pipeline design we modified our study to include a database of coding sequences representing complete gene model in the respective plant species considered.

- Why oil seed plants

Oil seed plants are used for the production of biodiesel. Their seed contain tri-glycerides (oils) which are converted to biodiesel through transesterification reaction with low molecular weight alcohol, ex. Ethanol, methanol.

The use of Bio-diesel is being considered a sustainable way out for realizing foreseeable decline in the production of fossil fuel as well as the environment pollution concerns associated with the use of fossil fuels. The bio-diesel can be made after transesterification from virgin or used vegetable oils (both edible and non-edible). There are several vegetable oils which are being projected as alternatives for the production of biodiesel. Some of them include rapeseed, palm oil, castor oil, sunflower, safflower, mustard, soyabean, jatropha, corn oil.

For the development of the bioinformatics pipeline three plant species were considered viz. *Arabidopsis thaliana, Ricinus communis* and *Glycine max* as they are oil producing plants (*Table 1.1*) and complete genome information is available for them.

Plant Species	**Seed oil content (%)**
Arabidopsis thalliana	34.6-46
Ricinus communis	30-35
Glycine max	20

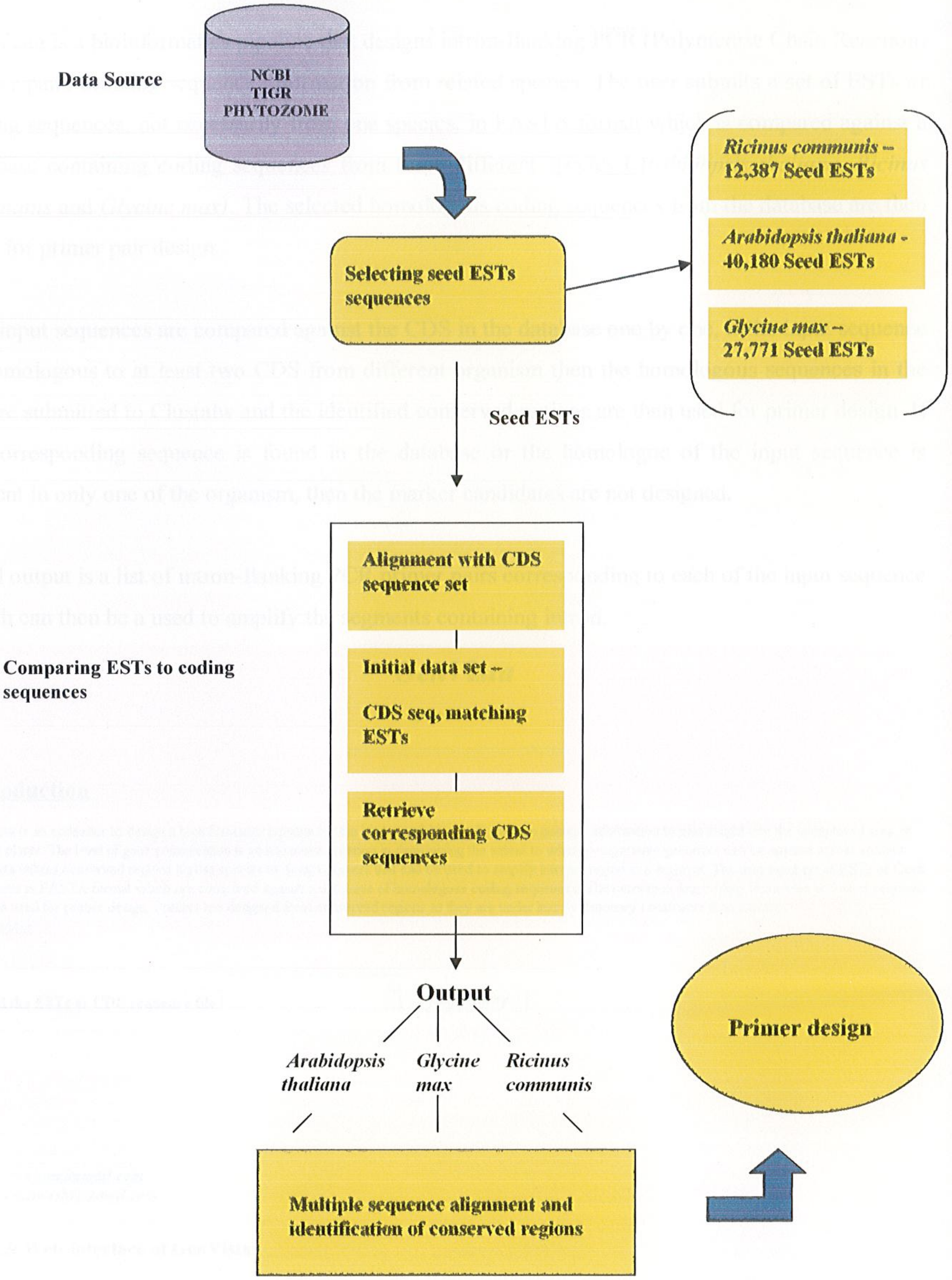
***Table 1.1*: Seed oil content of three different oil plants**

In India, concerted efforts are being made in projecting Jatropha (*Jatropha carcus*) as a suitable plant species for the commercial production of biofuel but the genome sequence information for Jatropha is unavailable.

The Castor bean (*Ricinus communis*) is a plant species, which is taxonomically and biochemically related to *Jatropha carcus* such as both of these belong to family Euphorbiaceae, both produce white sap, the seeds of both contain a similar toxic protein, and more importantly the seed oil from both these plant species is used in the production of biodiesel. The major fatty acid component in seed oil of Castor bean is ricinoleic acid (87%) and it is Linoleic & oleic acids (79%) in Jatropha. There could be similarities in the number and type of genes involved in the biosynthesis of fatty acids in the seed oil of both the Castor bean and Jatropha.

In the designed pipeline, we are utilizing genome resources of three oil plants (*Arabidopsis thaliana, Ricinus communis and Glycine max*) to gain insight into the unexplored areas of genomes of plants for which genome information is not available.

GenVista: Bioinformatics pipeline for identification of candidate gene anchor markers in plants *(Fig 1.1)*

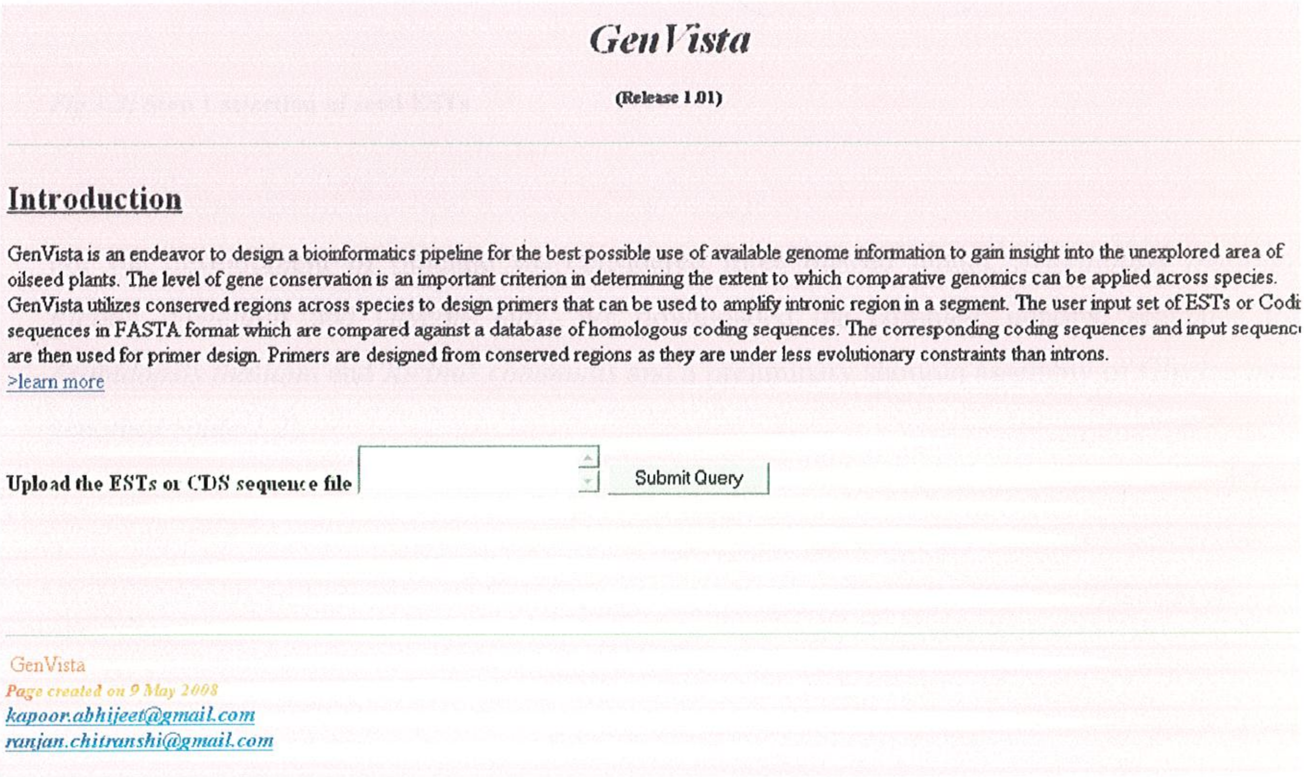

Data Source
NCBI
TIGR
PHYTOZOME
Selecting seed ESTs sequences
Ricinus communis – 12,387 Seed ESTs
Arabidopsis thaliana - 40,180 Seed ESTs
Glycine max – 27,771 Seed ESTs
Seed ESTs
Comparing ESTs to coding sequences
Alignment with CDS sequence set
Initial data set –
CDS seq, matching ESTs
Retrieve corresponding CDS sequences
Output
Arabidopsis thaliana
Glycine max
Ricinus communis
Multiple sequence alignment and identification of conserved regions
Primer design

FUNCTIONING OF GenVista

GenVista is a bioinformatics pipeline that designs intron-flanking PCR (Polymerase Chain Reaction) primer pairs utilizing sequence information from related species. The user submits a set of ESTs or coding sequences, not necessarily from one species, in FASTA format which is compared against a database containing coding sequences from three different species (*Arabidopsis thaliana, Ricinus communis* and *Glycine max)*. The selected homologous coding sequences from the database are then used for primer pair design.

The input sequences are compared against the CDS in the database one by one, if the input sequence is homologous to at least two CDS from different organism then the homologous sequences in the set are submitted to Clustalw and the identified conserved regions are then used for primer design. If no corresponding sequence is found in the database or the homologue of the input sequence is present in only one of the organism, then the marker candidates are not designed.

Final output is a list of intron-flanking PCR primer pairs corresponding to each of the input sequence which can then be a used to amplify the segments containing intron.



GenVista

(Release 1.01)

Introduction

GenVista is an endeavor to design a bioinformatics pipeline for the best possible use of available genome information to gain insight into the unexplored area of oilseed plants. The level of gene conservation is an important criterion in determining the extent to which comparative genomics can be applied across species. GenVista utilizes conserved regions across species to design primers that can be used to amplify intronic region in a segment. The user input set of ESTs or Codi sequences in FASTA format which are compared against a database of homologous coding sequences. The corresponding coding sequences and input sequenc are then used for primer design. Primers are designed from conserved regions as they are under less evolutionary constraints than introns.
>learn more

Upload the ESTs or CDS sequence file

GenVista
Page created on 9 May 2008
kapoor.abhijeet@gmail.com
ranjan.chitranshi@gmail.com

Fig 1.2: **Web-interface of GenVista**

MEHODOLOGY

The pipeline design comprised of four basic steps which are as following:

A) Selecting seed ESTs

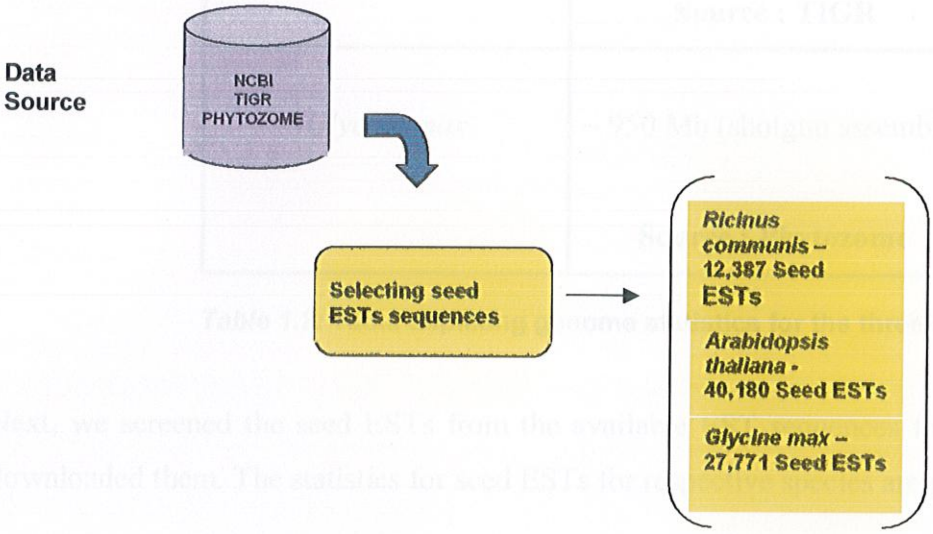

Data Source
NCBI
TIGR
PHYTOZOME
Selecting seed ESTs sequences
Ricinus communis – 12,387 Seed ESTs
Arabidopsis thaliana - 40,180 Seed ESTs
Glycine max – 27,771 Seed ESTs

***Fig 1.3:* Step 1 selection of seed ESTs**

For the development of pipeline we considered three oilseed plants, *Arabidopsis thaliana, Ricinus communis* and *Glycine max*. We downloaded the complete genome sequence for *Arabidopsis thaliana* and *Ricinus communis* and a preliminary shotgun assembly of *Glycine max* genome (*Table 1.2*).

Plant Species	Genome size
Arabidopsis thalliana	~ 125 Mb (consist of 5 chromosomes) **Source : NCBI**
Ricinus communis	~ 400 Mb (shotgun assembly consist of 28,518 contig sequences) **Source : TIGR**
Glycine max	~ 950 Mb (shotgun assembly) **Source : Phytozome**

Table 1.2: **Table depicting genome statistics for the three species.**

Next, we screened the seed ESTs from the available EST sequences for the three species and downloaded them. The statistics for seed ESTs for respective species are given below:

Plant Species	Seed ESTs
Arabidopsis thalliana	40,180
Ricinus communis	12,387
Glycine max	27,771

Table 1.3: **Table depicting seed EST statistics for the three species.**

Source: NCBI

As the oil production in these plants occur inside seed so, our analysis focused on ESTs from seed only.

B) ESTs alignment with coding sequences

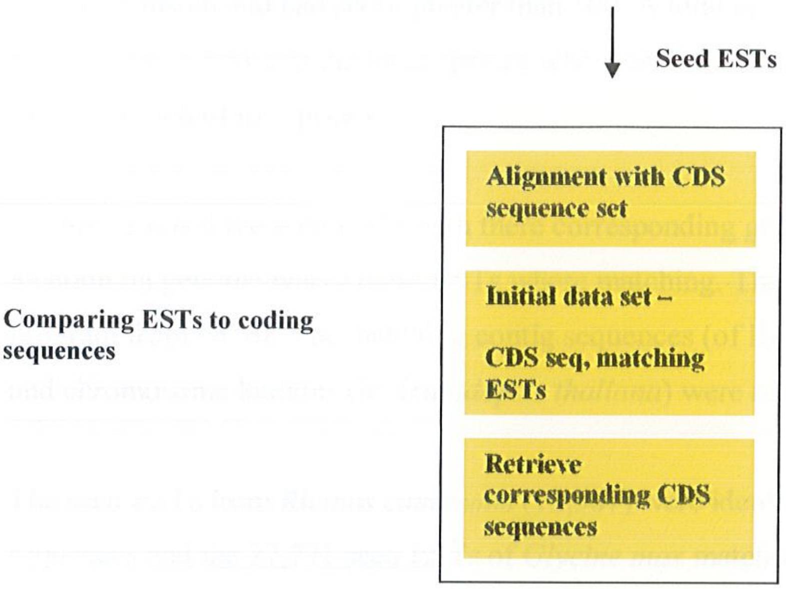

Seed ESTs
Comparing ESTs to coding sequences
Alignment with CDS sequence set
Initial data set – CDS seq, matching ESTs
Retrieve corresponding CDS sequences

Fig 1.4: **Generating homologous CDS set**

We aligned the seed ESTs with the coding sequences set for the respective species and retrieved the coding sequences that gave best match with the aligned ESTs. In this way, we screened coding sequences corresponding to seed from the set of coding sequences representing the complete gene model of plant species.

In *Ricinus communis*, of the complete coding sequence set, a total of 2,459 coding sequences were identified to be seed coding sequences. Similarly, in *Arabidopsis thaliana* and *Glycine max* this figure was found to be 9,965 and 13,333 respectively (*Table 1.4*).

Plant Species	Seed CDS
Arabidopsis thalliana	9,965
Ricinus communis	2,459
Glycine max	13,333

Table 1.4: **Seed CDS count in three species.**

In order to identify the corresponding seed coding sequences among the three species we compared these sequences using the BLAST program from NCBI and retrieved sequences that gave best match and had score greater than 100. A total of 776 set of sequences were identified to be common between the three species where each set contained three corresponding sequences one from each of the species.

We also aligned the seed ESTs with there corresponding genome sequences to identify the location on genome where these ESTs where matching. The analysis was done using the BLAST program from NCBI. The matching contig sequences (of R*icinus communis* and *Glycine max)* and chromosome loations (in *Arabidopsis thaliana*) were retrieved using perl codes.

The seed ESTs from *Ricinus communis* (12,387) were identified to be present in 630 contig sequences and the 27,771 seed ESTs of *Glycine max* matched 436 scaffolds. The identified contig and scaffold sequences were annotated using FGENESH to identify the genes in which these ESTs where matching.

C) Identification of conserved regions and primer design

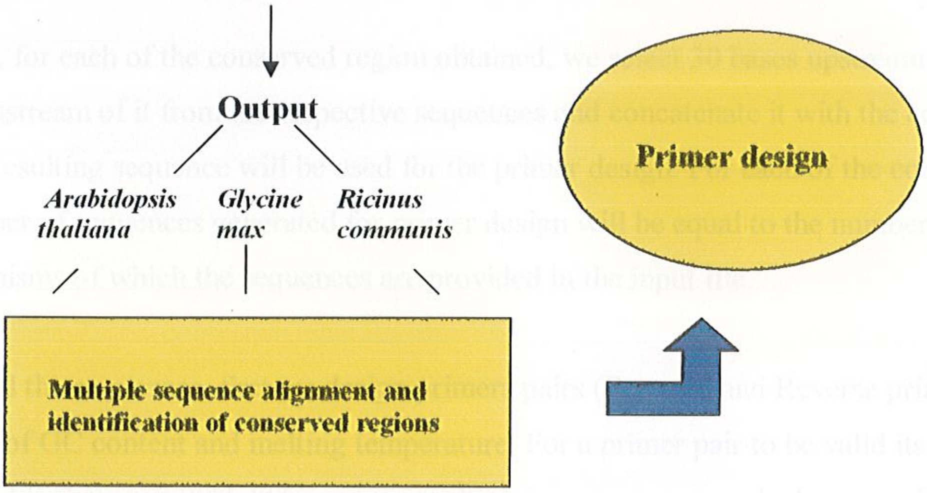

Output
Arabidopsis thaliana
Glycine max
Ricinus communis
Multiple sequence alignment and identification of conserved regions
Primer design

Fig 1.5: **Conserved region identification and primer pair design**

The set of homologous sequences are subject to multiple sequence alignment for identification of conserved regions. The multiple sequence alignment is performed using ClustalW program. The regions of conserved sequence identity in exons can be targeted to design primer pairs in order to amplify the intronic regions across various oilseed plant species.

For the design of primer pairs from the conserved regions across the species, we developed a perl code. We considered three parameters for the primer design:

1) Melting temperature of the primer which was kept between 50° C to 70° C.
2) GC content which varies between 20% - 80%.
3) Self-complementarity check.

C.1) Working of Primer Design Algorithm

For developing primer pairs we rely on the conserved regions across the species. Working of primer design algorithm is divided in to following steps:

1) First, from the multiple sequence alignment obtained through ClustalW, we select only those conserved regions across species which are 10 or more bases in length. Primers designed from conserved regions smaller than 10, say 5 or 7, are less specific and may bind to regions in genome from which they are not coming.

2) Next, for each of the conserved region obtained, we select 30 bases upstream and downstream of it from the respective sequences and concatenate it with the conserved region. The resulting sequence will be used for the primer design. For each of the conserved region, number of sequences generated for primer design will be equal to the number of different organisms of which the sequences are provided in the input file.

3) For all the sequences, first we design primers pairs (Forward and Reverse primers) on the basis of GC content and melting temperature. For a primer pair to be valid its GC content should lie between 20%-80% and the melting temperature must be between 50° C to 70° C provided the length of primer is under 30 bases. Thus, the maximum length of primer designed will never exceed 30 bases .Any primer pair having GC content and melting temperature not in the specified range will be rejected.

4) Finally, the program performs a self-complementarity check on the non-rejected primer pair sequences and provides a list of valid consensus primers.

Before designing the primer pairs (forward and reverse primers) for the 776 set of CDS obtained, we validated the working of primer design algorithm on a small dataset of candidate genes involved in the fatty acid metabolism of oilseed plants.

Testing primer design algorithm on candidate genes of fatty acid pathway

For validating the working of primer design algorithm we selected 14 different genes involved in the fatty acid metabolism pathway which were found to be common among the three species considered. Following are the 14 genes which we took:

1) Acetyl-CoA carboxylase

2) Ketoacyl-ACP synthase

3) Ketoacyl-ACP reductase

4) Ketoacyl-ACP dehydrogenase

5) Enoyl-ACP reductase

6) FATA (Thioesterase)

7) FATB (Thioesterase)

8) Stearoyl desautrase

9) Omega-6 desaturase (FAD2)

10) Palmitoyl desaturase

11) Linoleate desaturase

12) Oleoyl desaturase

13) Malonyl transferase

14) Oleoyl-12 hydroxylase

We collected the gene sequences for these 14 genes from the available public domain databases such as TAIR for *Arabidopsis thaliana* genes, TIGR castor genome database for *Ricinus communis* genes and Phytozome for *Glycien max* genes.

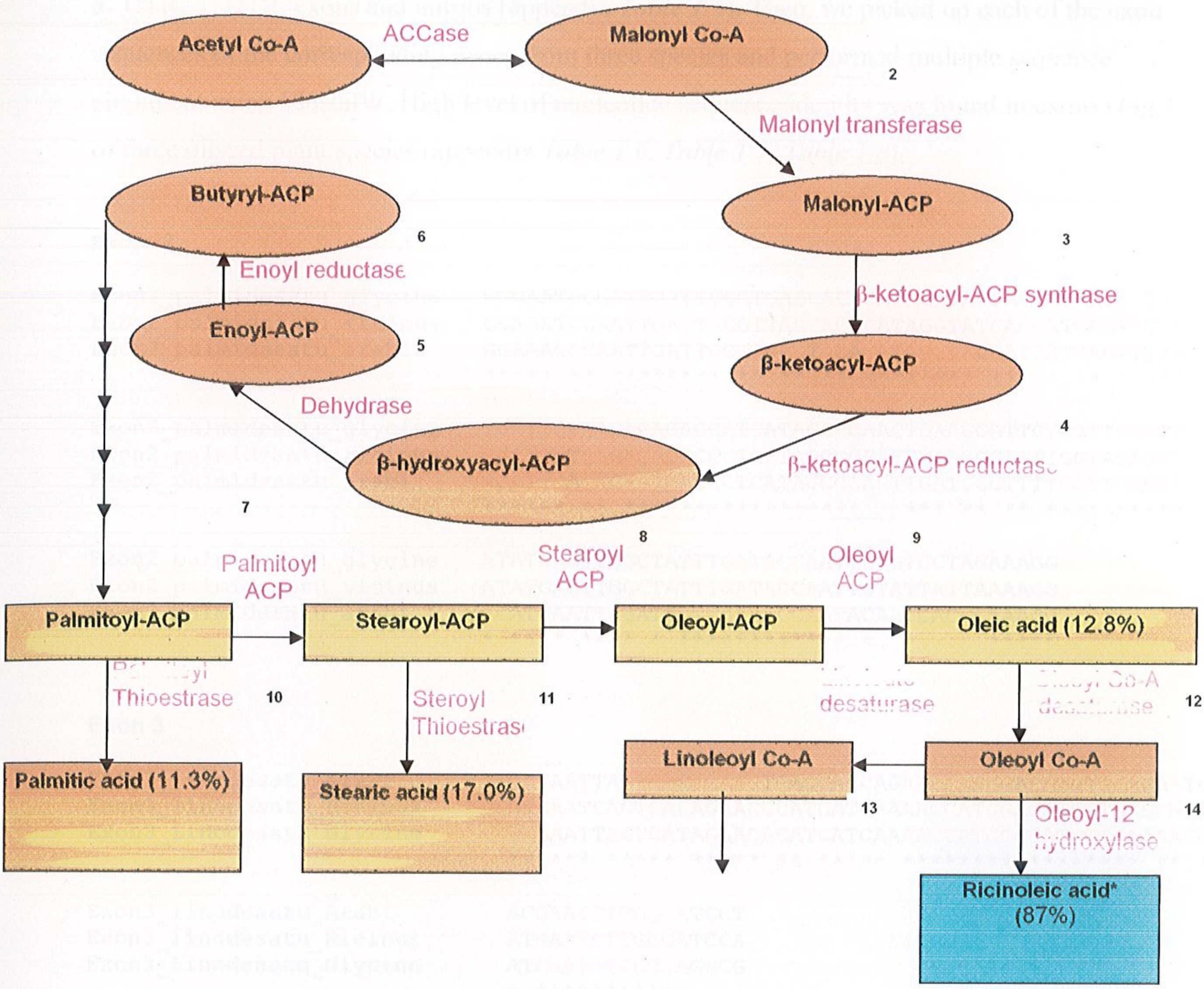


Fig 1.6: **Fatty acid biosynthetic pathway in oilseed plants depicting 14 common genes in the three species.**

Further, we annotate these sequences using FGENESH and generated a table showing the position of 5` UTR, 3` UTR, exons and introns (appendix *Table 1.5*). Then, we picked up each of the exon sequences of the corresponding genes from three species and performed multiple sequence alignment using ClustalW. High level of nucleotide sequence identity was found in exons (*Fig.1.7*) of three oilseed plant species (appendix *Table 1.6, Table 1.7, Table 1.8*).

```
Exon 2

Exon2_palmidesatu_glycine    GGGAATCCGATTGATTGGGTGAGCACACATAGGTACCATCACCAGTTTTG
Exon2_palmidesatu_ricinus    GGAAATCCAATTGATTGGGTTAGCACGCATAGGTATCACCATCAGTTTTG
Exon2_palmidesatu_arabi      GGAAACCCAATTGATTGGGTGAGTACACATAGGTACCATCATCAGTTTTG
                             ** ** ** *********** ** ** ******** ** ** ********

Exon2_palmidesatu_glycine    TGATTCTGAGAGAGACCCTCATAGCCCAACTGAAGGATTCTGGTTCAGTC
Exon2_palmidesatu_ricinus    TGATTCTGAGAGAGACCCTCATAGCCCCATTGAAGGGTTTTGGTACAGTC
Exon2_palmidesatu_arabi      TGATTCAGACAGAGACCCTCATAGCCCACTTGATGGGTTTTGGTTCAGTC
                             ****** ** *****************   *** ** ** **** *****

Exon2_palmidesatu_glycine    ATATGAGTTGGCTATTTGATACCAATTCTGTCCTAGAAAGG
Exon2_palmidesatu_ricinus    ATATGAGTTGGCTATTTGATACCAATTCTATTATTAAAAGG
Exon2_palmidesatu_arabi      ACATGAATTGGATGTTTGATACCAATACAATCACCCAAAGG
                             * **** **** * ************ *  *      *****
```

```
Exon 3

Exon3_linodesatu_Arabi      GAGAATTAGTCACAGAACTCACCACCAGAACCATGGACATGTTGAGAATG
Exon3_linodesatu_Ricinus    GAGAATCAGTCACAGAACCCATCATCAAAACCATGGACATGTTGAGAATG
Exon3_Linodesatu_Glycine    GAGAATTAGTCATAGAACACATCATCAAAACCATGGCCATGTTGAAACG
                            ****** ***** ***** ** ** ** ******** ******** ** *

Exon3_linodesatu_Arabi      ACGAATCTTGGCATCCT
Exon3_linodesatu_Ricinus    ATGAATCTTGGCATCCA
Exon3_Linodesatu_Glycine    ATGAATCTTGGCACCCG
                            * ***********
```

***Fig 1.7:* Conserevd sequences in exon 2 and exon 3 of palmitoyl desaturase and linoleate desaturase genes respectively.**

The regions of conserved sequence identity in exons can be targeted to design primer pairs for the amplification of intronic regions. So, we took the coding sequences corresponding to these 14 genes and designed forward and reverse primers for them using our primer design algorithm. These intron-flanking PCR primers were then used to amplify the intronic regions in *Ricinus communis.*

Designing primers for seed coding sequences

Next, we designed forward and reverse primers for the 776 sets of coding sequences obtained. These intron-flanking PCR primers were then used to amplify the intronic regions in *Ricinus communis.* Table 1.9 provides the complete list of designed primers for two sets of homologus sequences in *Ricinus communis* from the 776 sets identified. These primers have been ordered and will be used for PCR amplification of the intronic segments in *Ricinus communis.*

Gene set #	Castor bean	
	Left Primer	**Right primer**
1	GATCTTCAAAAATAGAGTGACAAATGT	GTTTGTTCTCCGTACCTGCGATC
	TGTAAGTGTACGTCAAACAAGAGG	CGAACACCAGTCGATTCTTTACTTA
3	AAAAGGATTTCTTGTGTCCGATTTG	TTCGTTTTTCAACTACTTAATTTACTC
	AAAAGGATTTCTTGTGTCCGATTTG	ATACTTCTCGTACTTTTTGCACGTA
	GGGCTGTGAAATTTCAATTAAGGAG	TCTTTTTACCTCGTCCTTCTTCGA
	TTTCCGAGAAGAAGAGAAAAATGGA	CAAAAACTGAAGAGGTGTCAAGACT
	ATGGAGCAGGAAGAAGCTGAGAG	CAAAAACTGAAGAGGTGTCAAGACT
	AAATGAGGTACGGACTGATCTCCA	ATACTTCTCGTACTTTTTGCACGTA
	GCTCAAGTAAGCTCTCATGGGC	CAAAAACTGAAGAGGTGTCAAGACT
	GCTCAAGTAAGCTCTCATGGGC	ATACTTCTCGTACTTTTTGCACGTA
	GCTCAAGTAAGCTCTCATGGGC	TACGAACATAGCCCGTCACTACTA
	CAGCCAATATAGTGTCAAGCATAGA	CAAAAACTGAAGAGGTGTCAAGACT
	CAGCCAATATAGTGTCAAGCATAGA	ACGACCTTCTCACTATCAGGCTG
	TCTAAACTTAGTTGCTTGAGTTGGAA	GTGTTCCTTTAAAGTTCTGGACACT
	AGAGTGTGATGGAATATGAAGAGCA	ACAACACAGTTCATATTAGGACCTA
	ACAGTACTTTACGATTATGGGATGT	ACGACCTTCTCACTATCAGGCTG

Table 1.9: **List of primers designed in *Ricinus communis* for the 2 sets of homologus sequences.**

Since, the base for the working of pipeline has been established and the working of different steps has been validated, next we extended the pipeline to include a database comprising coding sequences set representing complete gene model of the three plant species considered, so that the comparison of the user`s input multi-species sequence data with the coding sequences in database will make it possible to design candidate genes anchor markers across species.

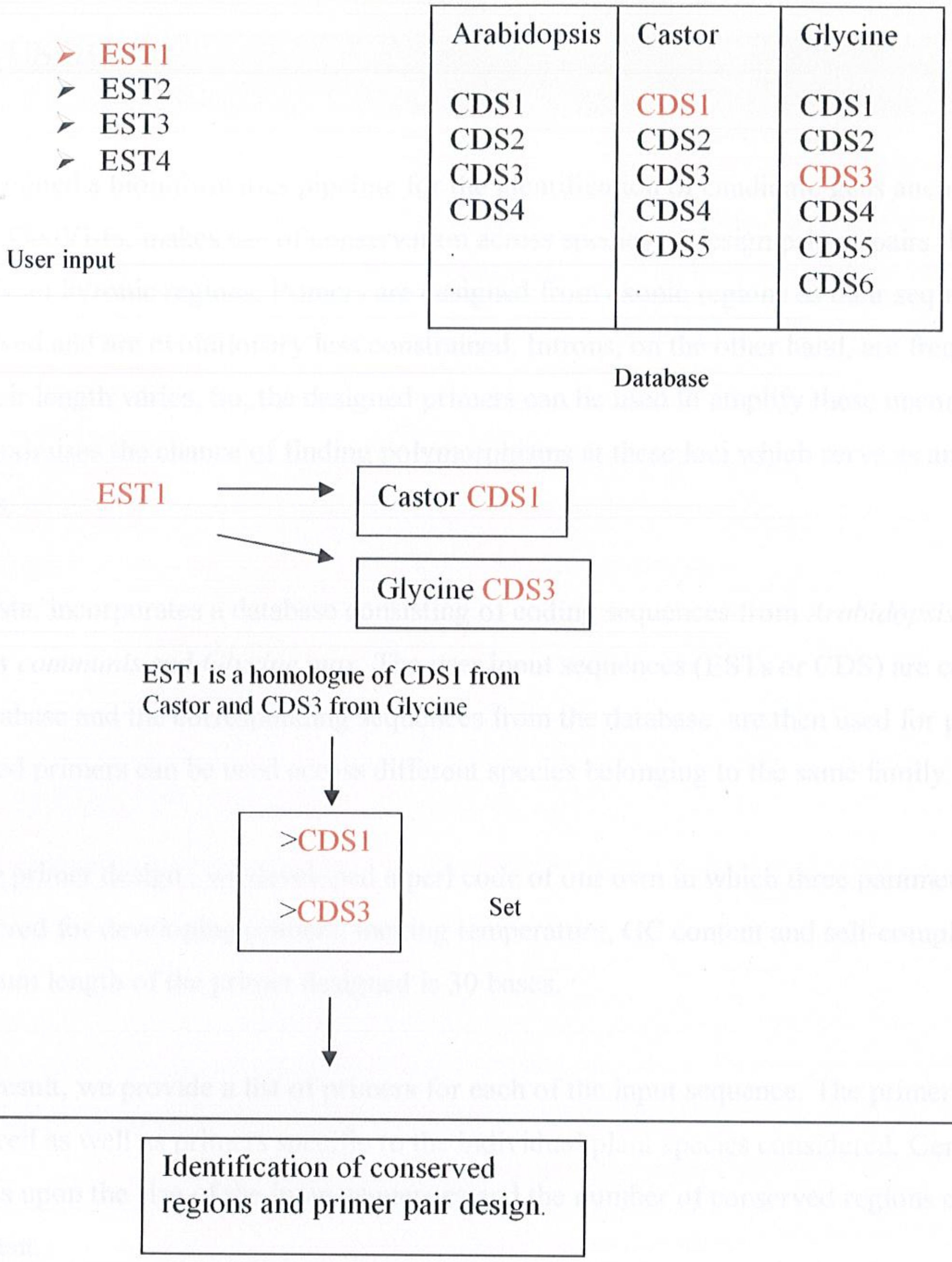
DATABASE DESIGN AND WORKING

We constructed a database of coding sequences from the three plant species viz. *Arabidopsis thaliana, Ricinus communis* and *Glycine max*.

The database consists of three relations:

1) **Organism:** In this relation, information about the plant species is stored. It has two attributes Organism_id and Organism_name. Organism_id is the primary key as well as foreign key which are used to reference other relations.
2) **Sequence:** In this relation, information regarding the coding sequences is stored. It has three attributes, Organism_id, Sequence_id and Sequence. (Organism_id + Sequence_id) are the primary key as well as foreign key which are used to reference other relations.
3) **Common_Sequences:** In this relation, we keep information regarding which sequences of the complete set are homologous to each other. It has four attributes n, common_id, sequence_id and organism_id. Here n is the primary key and it keeps track of number of record in the relation. It is incremented automatically, whenever a new record in the database is added.

The submitted EST sequences or coding sequences are first compared with database CDS one by one to generate matching EST-CDS (or CDS-CDS) set. For comparison of the sequences, we have used BLAST program from NCBI and we select the best matching sequence with a cut-off E-value of $10^{-6.}$ Likewise, when all the sets are designed (here a set consist of all the homologous CDS from database identified for the input sequences) they are submitted to ClustalW for multiple sequence alignment and further the identified conserved regions are used for primer design. (*Fig 1.5*)


EST1
EST2
EST3
EST4
User input
Database
EST1
Castor CDS1
Glycine CDS3
EST1 is a homologue of CDS1 from Castor and CDS3 from Glycine
>CDS1
>CDS3
Set
Identification of conserved regions and primer pair design.

Arabidopsis	Castor	Glycine
CDS1	CDS1	CDS1
CDS2	CDS2	CDS2
CDS3	CDS3	CDS3
CDS4	CDS4	CDS4
	CDS5	CDS5
		CDS6

Fig 1.8: **Generation of homologues set for primer design.**

DISCUSSION

We designed a bioinformatics pipeline for the identification of candidate gens anchor markers in plants. GenVista, makes use of conservation across species to design primer pairs that can be used to amplify an intronic regions. Primers are designed from exonic regions as their sequences are conserved and are evolutionary less constrained. Introns, on the other hand, are frequently altered and their length varies. So, the designed primers can be used to amplify these unconserved regions and maximises the chance of finding polymorphisms at these loci which serve as anchors across species.

GenVista, incorporates a database consisting of coding sequences from *Arabidopsis thaliana, Ricinus communis* and *Glycine max.* The user input sequences (ESTs or CDS) are compared against the database and the corresponding sequences from the database are then used for primer design.The designed primers can be used across different species belonging to the same family.

For the primer design , we developed a perl code of our own in which three parameters were mainly considered for developing primers, melting temperature, GC content and self-complemetarity. The maximum length of the primer designed is 30 bases.

In the result, we provide a list of primers for each of the input sequence. The primer list comprises conserved as well as primers specific to the individual plant species considered. GenVista, runtime depends upon the size of the input sequences and the number of conserved regions determined in the alignment.

BIBLIOGRAPHY:

Research papers

1. Jakob Fredslund, Lene H Madsen, Birgit K Hougaard, Anna Marie Nielsen, David Bertioli, Niels Sandal, Jens Stougaard, and Leif Schauser *A general pipeline for the development of anchor markers for comparative genomics in plants.* BMC Genomics. 2006
2. Jakob Fredslund, Lene H. Madsen, Birgit K. Hougaard, Niels Sandal, Jens Stougaard, David Bertioli, and Leif Schauser *GeMprospector—online design of cross-species genetic marker candidates in legumes and grasses.* Nucleic Acids Res. 2006

Web-pages

3. www.ncbi.nlm.nih.gov/
4. www.phytozome.com
5. www.tair.com
6. www.tigr.org/tdb/at/at.html
7. www.castorbean.tigr.org/

APPENDIX

A.1) Primer Design Algorithm – Perl code

```
#!C:/Perl/bin/perl.exe
##
## printenv -- demo CGI program which just prints its environment
##
print "Content-type: text/html; charset=iso-8859-1\n\n";

open(F1,"file\.aln");
open(F2,">msa1.txt");
$ref=0;
$ref2=1;
$i=0;
While (<F1>)
{
  If (/CLUSTAL/)
   {
   }
  Else
   {
    chomp ($_);
    If ($_ eq “)
    {
     If ($refl eq 1)
     {
      $m=$i;
      $i=0;
     }
    }
   Else
   {
    $refl=1;
    @w=split(/\s+/,$_);
    $names[$i]=$w[0];
    $name[$i]=$name[$i].$w[1];
    $i=$i+1;
   }
  }
}


$k=0;
For ($i1=0; $i1< ($m); $i1++)
{
    $_=$name[$i1];
     If (/\*/)
    {
    }
    Else
    {
      print F2 $name[$i1]."\n";
    }
  }

 open(F1,"msa1.txt");
 open(F2,">result.txt");
 $i=0;
 While (<F1>)
 {
   chomp($_);
   $name[$i]=$_;

   $i=$i+1;
 }
   $re=$i;

 @w=split('',$name[0]);
 $m=scalar(@w);
 For ($i1=0;$i1<$i;$i1++)
 {
   @w=split('',$name[$i1]);
    For ($i2=0;$i2<scalar(@w);$i2++)
   {
     If ($w[$i2] eq 'A'|| $w[$i2] eq 'a' )
     {
       $c[$i2][0]=$c[$i2][0]+1;
     }

    If ($w[$i2] eq 'T' || $w[$i2] eq 't')
    {
      $c[$i2][1]=$c[$i2][1]+1;
    }

   If ($w[$i2] eq 'C'  || $w[$i2] eq 'c')
   {
     $c[$i2][2]=$c[$i2][2]+1;
   }

  If ($w[$i2] eq 'G'|| $w[$i2] eq 'g')
  {
    $c[$i2][3]=$c[$i2][3]+1;
  }
 }
```

```
   }

   For ($i2=0;$i2<$m;$i2++)
   {
      $ref=0;
      For ($i3=0;$i3<4;$i3++)
      {
        If (($c[$i2][$i3]/$i) > 0.7 || ($c[$i2][$i3]/$i) eq 0.7)
        {
          $ref=1;
          $r1=$i3;
        }
      }
     If ($ref eq 1)
     {
       If ($r1 eq 0)
     {
       $x[$i2]='A';
     }
     If ($r1 eq 1)
     {
       $x[$i2]='T';
     }
     If ($r1 eq 3)
     {
       $x[$i2]='G';
     }
    If ($r1 eq 2)
    {
      $x[$i2]='C';
    }
  }
  Else
  {
  $x[$i2]=' ';
  }
  }

  For ($i3=0;$i3<$m;$i3++)
   {
    print F2 $x[$i3];
   }
 open(F1,"result.txt");
 open(F2,">result1.txt");
 While (<F1>)
 {
 chomp($_);
 $seq1=$_;
```

```
}
@w=split(",$seq1);
print  "<h2>Conserved Regions list</h2>";
print  "<table border=1>";
print “<tr><td>Align position</td><td>length</td><td>conserved sequence</td><td>melting
temperature</td></tr>";
For ($i=0;$i<scalar (@w);$i++)
{
 $c=0;
 For ($i1=$i;$i1<scalar(@w);$i1++)
 {
  If ($w[$i1] ne '' )
 } {
  $c=$c+1;
 }
 Else
 {
    $i1=scalar(@w);
 }
 }
If ($c > 9)
{
 print F2 ($i+1)."xxx".$c."yyyyy";
 print "<tr><td>". ($i+1)."</td>";
 print " <td>".$c."</td> ";
 $pos[$k]=$i+1;
 $c[$k]=$c;
 $c_C=0;
 $c_G=0;
 $c_A=0;
 $c_T=0;
 $seq[$k]=";
 print "<td>";
 For ($i1=$i;$i1<($i+$c);$i1++)
 {
  If($w[$i1] eq 'A' )
  {
   $c_A=$c_A+1;
  }
 If ($w[$i1] eq 'T' )
 {
   $c_T=$c_T+1;
 }
 If ($w[$i1] eq 'G' )
{
  $c_G=$c_G+1;
}
If ($w[$i1] eq 'C' )
{
  $c_C=$c_C+1;
 }
 print F2 $w[$i1];
 print  $w[$i1];
 $seq[$k]=$seq[$k].$w[$i1];
}
print  "</td>";
print F2 " ".((4*($c_G+$c_C))+(2*($c_A+$c_T)))."<BR>";
print  "<td> ".((4*($c_G+$c_C))+(2*($c_A+$c_T)))."</td></tr>";
$temp[$k]=((4*($c_G+$c_C))+(2*($c_A+$c_T)));
$k=$k+1;
$i=$i+$c-1;
}
}
print  "</table>";
print  "<br>";
print  "number of organisms ".$re."\n";
For ($k1=0;$k1<$re;$k1++)
{
   print  "<td>".$names[$k1]."</td>";
   print  Fl "<td>".$names[$k1]."</td>";
}
print "</tr>";
print Fl "</tr>";
For ($k1=0;$k1<$k;$k1++)
{
  $ref_pos[$k1]=0;
  print  "<tr><td>".$seq[$k1]."</td>";
  print Fl  "<tr><td>".$seq[$k1]."</td>";
  $num=30-$c[$k1];
  For ($k3=0;$k3<$re;$k3++)
  {
    $sequence[$k1][$k3]=";
    print  "<td>";
    print Fl "<td>";
    $k2=0;
    @w=split(",$name[$k3]);
    $e=1;
    $as=";
    If ($num > $e)
    {
     While ($k2 eq 0 )
     {
       If ($w[[$pos[$k1]-1]-$e] ne "-")
       {
         print  $w[$pos[$k1]-1-$e];
         print Fl $w[$pos[$k1]-1-$e];
         $sequence[$k1][$k3]=$w[$pos[$k1]-1-$e].$sequence[$k1][$k3];
         $e=$e+1;
       }
     If ($e > $num)
```

```
      {
        $k2=1;
      }
    }
  }
print   "&nbsp;&nbsp; ";
print   $seq[$k1];
If ($num < 1)
{
   @wxz=split(",$seq[$k1]);
   For ($rw=(scalar(@wxz)-1);$rw>(scalar(@wxz)-30);$rw--)
   {
     $sequence[$k1][$k3]=$sequence[$k1][$k3].$wxz[$rw];
   }
}
Else
{
  $sequence[$k1][$k3]=$sequence[$k1][$k3].$seq[$k1];
}
print Fl $seq[$k1];
$e=1;
$k2=0;
If ($num > $e )
{
   While ($k2 eq 0)
   {
     If ($w[[$pos[$k1]+$c[$k1]-2]+$e ] ne "-" )
     {
       print  $w[$pos[$k1]+$c[$k1]-2+$e];
       $sequence[$k1][$k3]=$sequence[$k1][$k3].$w[$pos[$k1]+$c[$k1]-2+$e];
       print Fl $w[$pos[$k1]+$c[$k1]-2+$e];
       $e=$e+1;
     }
     If ($e > $num)
     {
       $k2=1;
     }
   }
}
}
}

For ($i1=0;$i1<$k;$i1++)
{
  For ($i2=0;$i2<$re;$i2++)
  {
     print    "<td>".$i1." ".$i2." ".$sequence[$i1][$i2]."</td>";
  }
 }
```

```
For ($i1=0;$i1<$k;$i1++)
{
   For ($i2=0;$i2<($re);$i2++)
   {
     ($left1,$right1)=primer($sequence[$i1][$i2],$c[$i1]);
     $left[$i1][$i2]=$left1;
     $right[$i1][$i2]=$right1;
    }
}
For ($i1=0;$i1<$k;$i1++)
{
  For ($i2=0;$i2<($re);$i2++)
  {
     print  $left[$i1][$i2]."&nbsp;&nbsp;".$right[$i1][$i2];
}
}
#validation of left primer and rigth primer
#most probale left primer

For ($i2=0;$i2<($re);$i2++)
{
   print  "<td>".$names[$i2]."</td>";
   For ($i1=0;$i1<$k;$i1++)
   {
     For ($i3=0;$i3<$k;$i3++)
     {
       @sx1=split(",$left[$i1][$i2]);
       @sx2=split(",$right[$i3][$i2]);
        If ( (scalar(@sx1) > 0 ) && (scalar(@sx2) > 0) )
        {
           $qw=complimentarity($left[$i1][$i2],$right[$i3][$i2]);
           If ($qw eq 0)
           {
             @wq=split(",$right[$i3][$i2]);
             $seqq=";
             foreach $x(@wq)
             {
               If ($x eq ‘A’)
               {
                  $seqq=$seqq.'T';
               }
               If ($x eq ‘T’)
               {
                  $seqq=$seqq.'A';
               }
              If ($x eq ‘G’)
              {
                  $seqq=$seqq.'C';
              }
              If ($x eq ‘C’)
```

```
          {
          $seqq=$seqq.'G';
           }
         }
     print  "<td>".$left[$i1][$i2]."<br>".$seqq."<br>".($pos[$i3]-$pos[$i1])."<br>";
     }
   }
  }
}
}
For ($i2=0;$i2<($re);$i2++)
{

   For ($i1=0;$i1<$k;$i1++)
     {
       for($i3=0;$i3<$k;$i3++)
         {
          }
     }
 }
sub primer()
{
($seq,$length)=@_;
$length1=$length;
$left='';
$right='';
@sequence1=split('',$seq);
#minmum primer size is 10
$ref=0;
For ($i=0;$i<20;$i++)
{  $seq1='';
   For ($id=$i;$id<30;$id++)
   {
     $seq1=$seq1.$sequence1[$id];
   }
  @seq2=split('',$seq1);

  #gc content
  $gc=0;
  $at=0;
  $length=0;
  $score=0;
  foreach $x(@seq2)
  {
    If ($x eq 'G' ||  $x eq 'g' || $x eq 'C' || $x eq 'c' )
    {
      $gc=$gc+1;
    }
    If ($x eq 'A' ||  $x eq 'a' || $x eq 'T' || $x eq 't' )
```

```
        {
        $at=$at+1;
        }
       $length=$length+1;
    }
If ($length ne 0 )
{
  If (($gc/$length) > 0.2)
  {
     $score=$score+1;
  }
}
$tm=(4*($gc)+2*($at) );
If ( $tm > 49 && $tm < 71 )
{
   $score=$score+1;
}

If ($score eq 2 )
{
    $left=$seq1;
    $i=21;
}
}

#rigth primer
For ($i=scalar(@sequence1);$i>(30-$length+10);$i--)
{
  $seq1='';
 For ($id=(30-$length1);$id<$i;$id++)
  {
   $seq1=$seq1.$sequence1[$id];
  }
  @seq2=split('',$seq1);
#gc content
$gc=0;
$at=0;
$length=0;
$score=0;
foreach $x(@seq2)
{
   If ($x eq 'G' ||  $x eq 'g' || $x eq 'C' || $x eq 'c' )
   {
      $gc=$gc+1;
   }
   If ($x eq 'A' ||  $x eq 'a' || $x eq 'T' || $x eq 't' )
   {
    $at=$at+1;
```

```
     }
     $length=$length+1;
 }
If (($gc/$length) > 0.2 )
{
    $score=$score+1;
}
$tm=(4*($gc)+2*($at) );
If ( $tm > 49 && $tm < 71 )
{
   $score=$score+1;
}

If ($score eq 2 )
{
   $right=$seq1;
   $i=30-$length;
  }

 }
  return($left,$right);

}
sub complimentarity
{
($seq11,$seq22)=@_;
@s11=split('',$seq11);
@s22=split('',$seq22);
$m=1;
$mis=-1;
$g=-2;

For ($ix=0;$ix<(scalar(@s11)+1);$ix++)
{
    $m[$ix][0]{score}=0;
    $m[$ix][0]{p}=2;
}

For ($ix=0;$ix<(scalar(@s22)+1);$ix++)
{
     $m[0][$ix]{score}=0;
     $m[0][$ix]{p}=3;
}

$m[0][0]{p}=0;

For ($ix1=1;$ix1<(scalar(@s11)+1);$ix1++)
{
    For ($ix2=1;$ix2<(scalar(@s22)+1);$ix2++)
   {
     If ($s11[$ix1-1] eq  $s22[$ix2-1] )
     {
        $sc1=$m[$ix1-1][$ix2-1]{score}+$m;
     }
    Else
     {
       $sc1=$m[$ix1-1][$ix2-1]{score}+$mis;
     }

   $sc2=$m[$ix1-1][$ix2]{score}+$g;
   $sc3=$m[$ix1][$ix2-1]{score}+$g;

   $p1=max($sc1,$sc2,$sc3);
   If ($p1 eq 1 || $p1 eq 2 || $p1 eq 3 )
   {
   }
  Else
  {
   $p1=1;
  }
  If ($p1 eq 1 )
  {
   If ($sc1 < 0)
  {
   $sc1=0;
  }
   $m[$ix1][$ix2]{score}=$sc1;
   $m[$ix1][$ix2]{p}=1;
  }

 If ($p1 eq 2 )
 {
   If ($sc2< 0)
   {
     $sc2=0;
   }
  $m[$ix1][$ix2]{score}=$sc2;
  $m[$ix1][$ix2]{p}=2;
 }

 If ($p1 eq 3 )
 {
   If ($sc3 < 0 )
   {
     $sc3=0;
    }
   $m[$ix1][$ix2]{score}=$sc3;
   $m[$ix1][$ix2]{p}=3;
 }
 }
```

```
}
open(F11,">matrix.txt");
print F11 "  ";
For ($ix2=0;$ix2<scalar(@s22);$ix2++)
{
  print  F11 $s22[$ix2]." ";
}
print F11 "\n";

For ($ix1=0;$ix1<(scalar(@s11)+1);$ix1++)
{
  print F11  $s11[$ix1-1]." ";
  For ($ix2=0;$ix2<(scalar(@s22));$ix2++)
  {
    print F11 $m[$ix1][$ix2]{score}." ";
  }
 print F11 "\n";
}
close(F11);
$r=0;
$counts=0;
$row[$counts]=-1;
$col[$counts]=0;
$counts=$counts+1;
While ($r < (scalar(@s11)+1) )
{
  $max=$m[$r][0]{score};
  $r1=$r;
  $c1=0;
  For ($t1=$r;$t1<(scalar(@s11)+1);$t1++)
  {
    For ($t2=0;$t2<(scalar(@s22)+1);$t2++)
    {
      If ($max < $m[$t1][$t2]{score} )
        {
        $max=$m[$t1][$t2]{score};
         $r1=$t1;
         $c1=$t2;
        }
      }
   }
 $row[$counts]=$r1;
 $col[$counts]=$c1;
 $counts=$counts+1;
 $r=$r1+1;

}

$m=0;
$r=0;
```

```
$c=1;
$r1=0;$c1=1;
$data1=";
$data2=";
$pos11=";
$pos22=";

$data3=";
$data4=";
For ($counts1=1;$counts1<$counts;$counts1++)
{  $data3=";
   $data4=";
   $pos1=";
  $pos2=";
   $r1=$row[$counts1];
   $c1=$col[$counts1];
   $r2=$row[$counts1-1];
   $c2=$col[$counts1-1];

   While ($r1 > $r2  )
   {
     If ($m[$r1][$c1]{p} eq 1 )
     {
         print  F11  $r1." ".$c1." ";
         $data3=$data3.$s11[$r1-1]." ";
         $pos1=$pos1.($r1-1)." ";
         $data4=$data4.$s22[$c1-1]." ";
         $pos2=$pos2.($c1-1)." ";
         $r1=$r1-1;
         $c1=$c1-1;
      }
    Else
    {
      If ($m[$r1][$c1]{p} eq 2 )
      {
        print  F11  $r1." ".$c1." ";
        $data3=$data3.$s11[$r1-1]." ";
        $pos1=$pos1.($r1-1)." ";
        $data4=$data4.'--'." ";
        $pos2=$pos2."--"." ";
        $r1=$r1-1;
       print  F11  $r1." ".$c1."\n";
       }
      Else
      {
         if($m[$r1][$c1]{p} eq 3 )
         {
           print  F11  $r1." ".$c1." ";
```

```
                $data3=$data3.'--'." ";
                $data4=$data4.$s22[$c1-1]." ";
                $pos1=$pos1."--"." ";
                $pos2=$pos2.($c1-1)." ";
                $c1=$c1-1;
                print F11 $r1." ".$c1."\n";
            }
        }
    }
If ($r1 eq 0)
{
  last;
}
}
If ($r2 eq -1 )
{
  For ($c2=$c1;$c2>0;$c2--)
    {
      $data3=$data3.'--'." ";
      $pos1=$pos1."--"." ";
      $data4=$data4.$s22[$c2-1]." ";
      $pos2=$pos2.($c1-1)." ";
      $c2=$c2-1;
    }
 }
@d3=split(",$data3);
@d4=split(",$data4);
@pos1=split(/[ ]/,$pos1);
@pos2=split(/[ ]/,$pos2);
print F11 "\n".$data3."\n".$data4."\n";
For ($i=(scalar(@d3)-1);$i > -1 ;$i--)
{
  $data1=$data1.$d3[$i];
}
For ($i=(scalar(@d4)-1);$i > -1 ;$i--)
{
  $data2=$data2.$d4[$i];
}
For ($i=(scalar(@pos1)-1);$i > -1 ;$i--)
{
  $pos11=$pos11.$pos1[$i]." ";
}
For ($i=(scalar(@pos2)-1);$i > -1 ;$i--)
{
  $pos22=$pos22.$pos2[$i]." ";
}
$data3=";
$data4=";
```

```
}
print F11 "\n".$data1."\n".$data2;
close(F11);
open(F12,">>sd.txt");
print F12 "\n".$data1."\n".$data2."\n";
print F12 "\n".$pos11."\n".$pos22."\n";
close(F12);
@d1=split(/[ ]/,$data1);
@d2=split(/[ ]/,$data2);
@pos11=split(/[ ]/,$pos11);
@pos22=split(/[ ]/,$pos22);
$ref=0;
$count=0;
$score=0;
$prev_pos1=$pos11[0];
$prev_pos2=$pos22[0];
For ($i=0;$i<scalar(@d1);$i++)
{
  If ($d1[$i] eq $d2[$i] )
  {
     If ( (($pos11[$i]-$prev_pos1) eq 1 ) && (($pos22[$i]-$prev_pos2) eq 1 ) )
     {
       $count=0;
       $score=$score+1;
     }
   Else
   {
     $count=$count+1;
     If ($count > 2 )
     {
       $score=0;
     }
   }
  }
  Else
  {
   $count=$count+1;
    If ($count > 2 )
    {
      $score=0;
    }
  }
  If ($score > 4 )
  {
   $ref=1;
  }
  $prev_pos1=$pos11[$i];
  $prev_pos2=$pos22[$i];
}
return ($ref);
```

```
}
sub max()
{
($x1,$x2,$x3)=@_;
If ($x1 > $x2 && $x1 > $x3 )
{
  return 1;
}
If ($x2 > $x1 && $x2 > $x3 )
{
  return 2;
}

If ($x3 > $x1 && $x3 > $x2 )
{
   return 3;
}

If ($x1 eq $x2 && $x1 > $x3 )
{
  return 1;
}

If ($x1 > $x2 && $x1 eq $x3 )
{
  return 1;
}
If ($x1 eq $x2 && $x1 > $x3 )
{
  return 1;
}

If ($x2 > $x1 && $x2 eq $x3 )
{
   return 2;
}
If ($x2 eq $x3 && $x2 < $x1 )
{
  return 1;
}
If ($x2 eq $x1 && $x1 < $x3 )
{
  return 3;
}

if ($x1 eq $x2 && $x3 > $x1 )
{
  return 3;
}
```

```
Else
{
  return 1;
}

}
```