**International Journal of Multidisciplinary and Current Research**

*Research Article*

ISSN: 2321-3124
Available at: http://ijmcr.com

# A Bioinformatics toolkit for rice (Oryza sativa Japonica) YUCCA genes (OS1-7) sequence comparative analysis

**Rabia Tabassum**

Department of Bioinformatics, Sir Syed University of Engineering & Technology , (SSUET) Karachi, Pakistan.

Accepted 15 Aug 2014, Available online 27 Aug 2014, **Vol.2** (**July/Aug 2014 issue)**

***Abstract***

*It has been estimated that almost half of the world's population relay wholly or partially on rice, Oryza sativa. The molecular mechanisms of IAA synthesis in rice (Oryza sativa), identified seven YUCCA-like genes (named OsYUCCA1-7) in the rice genome. Plants over expressing OsYUCCA genes exhibited increased IAA levels and characteristic auxin overproduction phenotypes. In this Project Visual basic is a computer language which is used to develop software (YUCCA IN) Visual Basic is a complete form of package for building user interface. The application program that was considered suitable for the program is Visual Basic version 6.Bioinformatics of biological data, using networks of computers and databases. involves the collection, organization and analysis of large amounts of data and on the basis of these tools can develop a software (YUCCA IN) that can check the mutant sequence like in this research the software will match the sequence of the entered gene with OSYUCCA (1-7).Software (YUCCA IN) is designed in Visual Basic. It will show the match and mismatch in the comparisons with the mutant genes and also it's phenotypic abnormality.*
*YUCCA IN compares different genes with over expressed genes and shows amino acid identity match and mismatch and also phenotypic abnormality of respective gene in Visual basic that greatly simplifies window application development. It saves time and reduces errors to its minimum level as compare with analytical approach. by using dynamic programming algorithm.*

***Keywords****: Oryza sativa Japonica, OsYUCCA1-7, Visual Basic, sequence comparative analysis, phenotypic abnormality*

## Introduction

There is a predominant hormone in rice known as a indole acetic acid(IAA).Indole-3-acetic acid (IAA), the predominant auxin in plants, its biosynthesis and regulation have not been clearly elucidated. Indole-3-acetic acid (IAA), the predominant auxin in plants, plays a critical role in many plant growth and developmental processes, including cell division, differentiation, and elongation; flower and vascular development; and tropism [1].Two major pathways for IAA biosynthesis have been proposed: the Trp (Tryptophan)-dependent and Trp-independent pathways[2].The molecular mechanisms of IAA synthesis in rice (Oryza sativa), identified seven YUCCA-like genes (named OsYUCCA1-7) in the rice genome [3].

In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Aligned sequences of nucleotide or amino acid residues are typically represented as rows within a matrix. Gaps are inserted between the residues so that identical or similar characters are aligned in successive columns [4].

There are two methods of pair wise sequence alignment, as in the YUCCA IN Software user enter its sequence to compares its homology with YUCCA group of genes so it can be done by Global as well as by local pair wise sequence alignment methods. Both sequence alignment methods based on dynamic programming

The Smith-Waterman algorithm is a general local alignment method also based on dynamic programming. With sufficiently similar sequences, there is no difference between local and global alignments [5].

To develop any software a programming language is used to run a program. Visual Basic is a programming language and environment developed by Microsoft. Based on the BASIC language, Visual Basic was one of the first *products to provide a graphical programming environment and a paint metaphor for developing user interfaces [6].*

## Existing System

The overepression of Yucca genes (Os-Yucca1-7) lead to phenotypic abnormality in Rice. There has been already a lot of work done on these genes by different technologies as plants grown on MS media, to compare the efficiency of the PCR primer sets for each OsYUCCA genes and

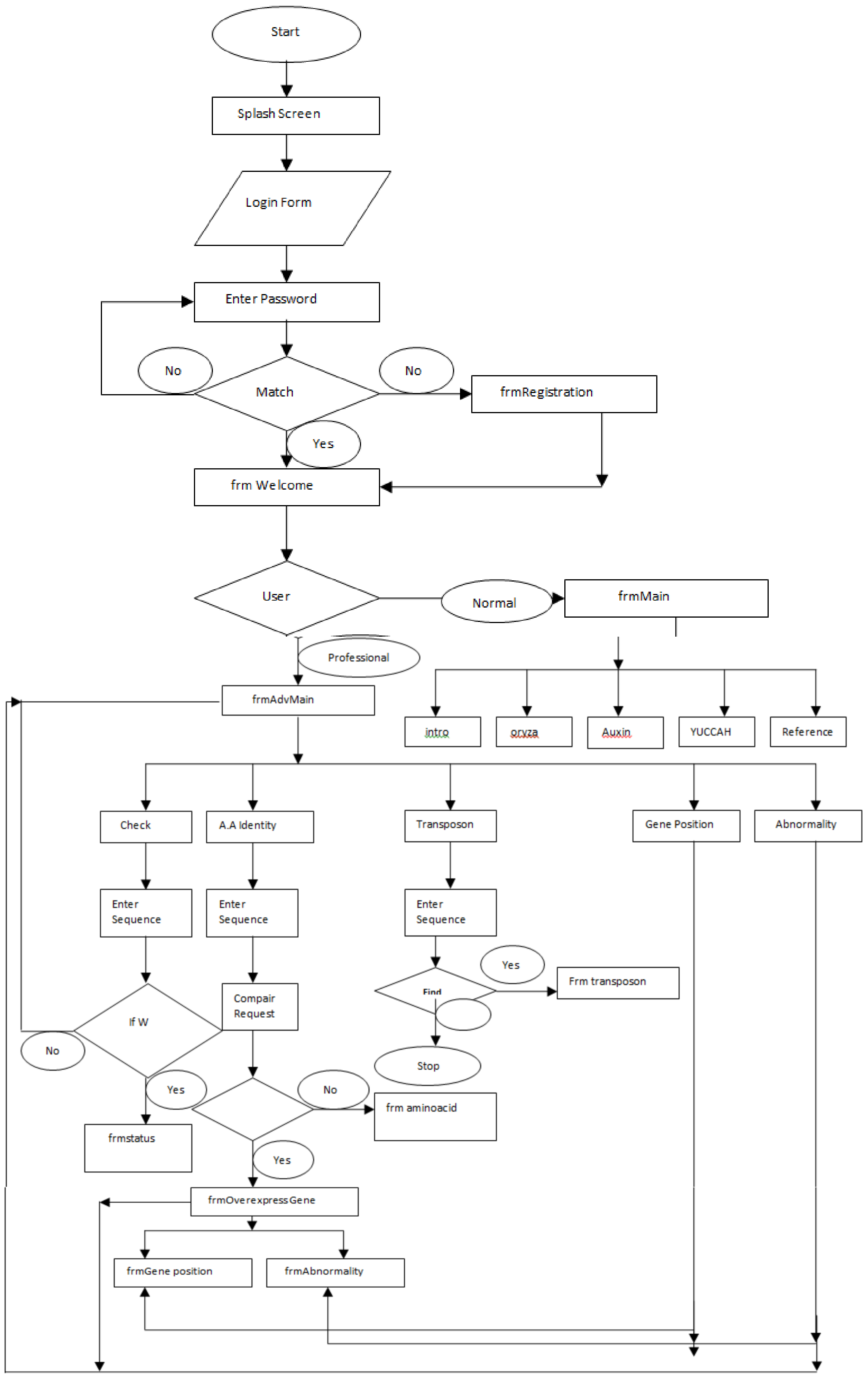


**Fig 1:** Data Flow Diagram

Phenotypes of transgenic rice plants expressing OsYUCCA under the control of a steroid hormone-inducible system.This Project was aimed to check the over-expression of Yucca genes (Os-Yucca1-7) lead to phenotypic abnormality in Rice by developing software using dynamic programming. No work is yet done related to gene comparison, Mutation detection and showing gene Phenotypic Abnormality of Rice. Thus in future, it may be used to compare normal and mutant genes show there amino acid identity and phenotypic abnormality.

## Objectives of project

Goal of the project is to develop software that should be efficient, authentic, simple to use, having following objectives.

- To compare rice DNA sequences of two rice cultivars at YUCCA genomic loci.
- To check the possible mutation in rice gene sequences.

To predict the phenotypic abnormality in rice by using the gene expression data

## Materials and methods

Availability and requirements

Lists the following:

- Project name: YUCCA IN
- Project home page: Desktop Based
- Operating system(s): Platform independent
- Programming language: Visual Basic

## Collection of Data

All the Sequences of the Os-YUCCA genes and Transposon were gathered from European Molecular Biology laboratory (EMBL), Gene Bank and Rice Tos17 Insertion Mutant Database. Following sequences were collected:

1. Os-YUCCA1
2. Os-YUCCA2
3. Os-YUCCA3
4. Os-YUCCA4
5. Os-YUCCA5
6. Os-YUCCA6
7. Os-YUCCA7

## Collection of Literature

All the relevant information stated in introduction and literature review of the thesis was gathered from authentic research papers, scientific journals and Rice databases.

Detailed Design:
Algorithm of check sequence

```
Event Name: click on cmdNo
Arguments: None
Variables: None
Begin:
    IF
        Condition:cmdNo Is Pressed
        Load advMain
        Show advMain
        Hide Current
    End IF
End
```

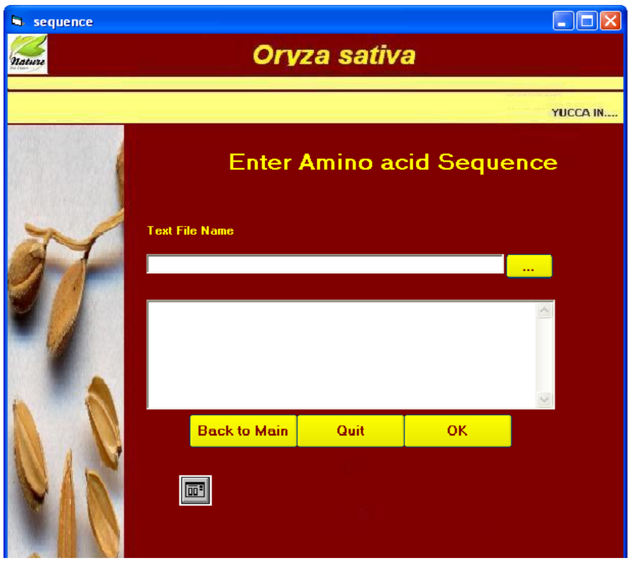


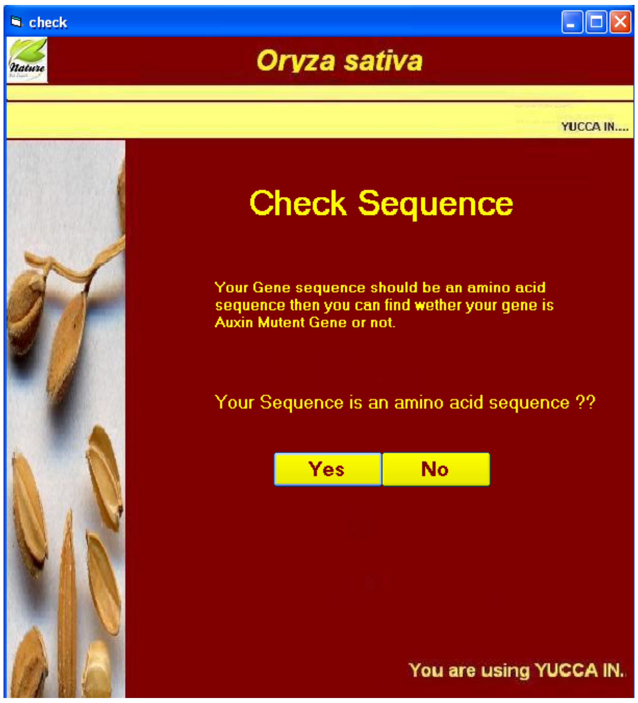
**Figure 2:** Check form

```
Coding frm check:
Private Sub cmdNo_Click()
If cmdNo.Value = True Then Load advmain
advmain.Show
Me.Hide
End Sub

Private Sub cmdyes_Click()
If cmdyes.Value = True Then Load sequence
sequence.Show
Me.Hide

End Sub
```

**Algorithm on Sequence form**

**Coding frm Sequence**

```
Private Sub LoadFile()
  Dim strText As String
  txt.Text = ""
  strText = Space(FileLen(txtFile.Text))
  Open txtFile.Text For Binary As #1
    Get #1, , strText
  Close #1
  txt.Text = strText
End Sub
--------------------------------------
Private Sub cmdBacktomain_Click()
If cmdBacktomain.Value = True Then Load advmain
advmain.Show
Unload Me
End Sub
--------------------------------------
Private Sub cmdBrowse_Click()
 Dim strFileName As String

  With dlg
    .Filter = "Data File|*.TXT|All Files|*.*"
    .ShowOpen
    strFileName = .FileName
   If strFileName <> "" Then
      txtFile.Text = strFileName
      LoadFile
    End If
  End With
End Sub
```



Check Sequence

Your Gene sequence should be an amino acid sequence then you can find wether your gene is Auxin Mutent Gene or not.

Your Sequence is an amino acid sequence ??

You are using YUCCA IN.

**Figure 3:** Sequence form

```
Public Function GetFileText(strFileName As String) As
String
 Dim strText As String
  strText = Space(FileLen(App.Path & "\Files\" &
strFileName))
  Open App.Path & "\Files\" & strFileName For Binary As
#1
    Get #1, , strText
  Close #1
 GetFileText = strText
 End Function
--------------------------------------
Private Sub cmdOK_Click()
  Dim i As Integer
  Dim mCount As Integer
  For i = 1 To Len(txt.Text)
    If UCase(Mid(txt.Text, i, 1)) = "W" Then
      mCount = mCount + 1
    End If
  Next
    If mCount > 5 And mCount < 11 Then
    status.Show
    Me.Hide
```

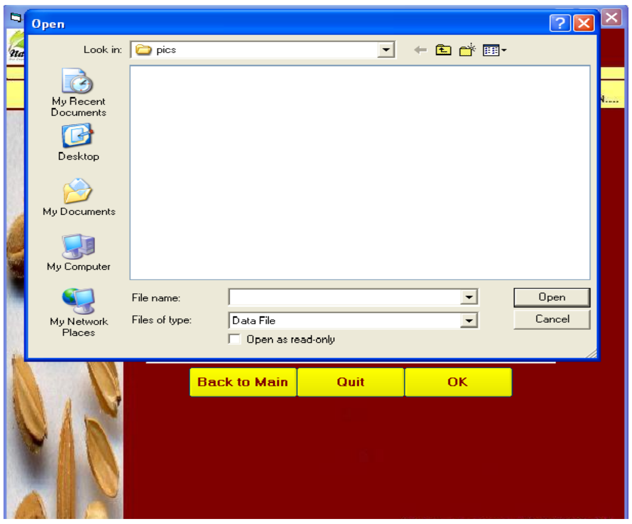


**Figure 4:** Browse

**Algorithm of compare Sequence**

```
Event Name: Click on CmdBrowse
Arguments: None
Variables:
      strFileName Type String
Begin:
      IF
      Condition:
      strFileName is Not Empty
      Set: txtFile.Text:=strFileName
      LoadFile
      End IF
End
--------------------------------------
Func Name:GetFileText
Arguments:
            strFileName
Variables:
            strText Type String
Begin:
      Set:=   strText:= Application Path
      Get: File Text in strText
      Close: File
```

```
    Set:=GetFileText=strText

End
---------------------------------------
Event Name: Click on cmdCompare
Arguments: None
Variables:
        mMatchNumber Type Intiger
        mTotalCharacterCount Type Long
        mMatchedCount(7) Type Long
        mMismatchedcount(7) Type Long
        mCount Type Long
        strText Type String
        strFileName Type String
        strFile Type String
        strMessage Type String

        i Type Intiger
        j Type Long
        intindex Type Intiger

Begin:
        Set: intIndex:=cboCompare.ListIndex
        IF intIndex:=7
            Goto Label: Compare ALL
        End IF

                SET: strFileName:="Yucca"& Index+1&".txt"
                SET: strText:=GetFileName(strFileName)
                SET: mTotalCharacterCount:=Len(strTex
                IF
                Condition:
                txt.Text:=strText
                SELECT CASE [intIndex]

                CASE 0
                SHOW: frmYUCCA1
                CASE 1
                SHOW: frmYUCCA2
                CASE 2
                SHOW: frmYUCCA3
                CASE 3
                SHOW: frmYUCCA4
                CASE 4
                SHOW: frmYUCCA5
                CASE 5
                SHOW: frmYUCCA6
                CASE 6
                SHOW: frmYUCCA7

                HIDE: Current FORM
            END SELECT
        ELSE
            LOAD: notMatch
            SET:    notmatch.mMisMatchFile:=strFileName
            SET:    notMatch.mstrUserText:=txt.Text
            SHOW: notMatch
                HIDE: Current Form
            End IF
            LABEL: CompareALL
        FOR LOOP
            i:=1 TO 7
            SET: strFileName:="Yucca"&i&".txt"
            SET: strFileName:=GetFileText(StrFileName)"
            SET: mTotalCharacterCount:=Len(strText)

                IF
                    Condition:txt.Text=strText
                    mMatchNumber:=i
                    EXIT FOR
                END IF
        LOOP BACK
        FOR
            I=1 TO 7
            SET: strFileName:="Yucca"&i&".txt"
            SET: strFileName:=GetFileText(StrFileName)"
            SET: mTotalCharacterCount:=Len(strText)
            SET: mCount(i)=mTotalCharactercount
            SET: mMatchCount(i)=0
            mMismatchedCount(i)=0
                LOOP FOR
                Condition: J:=1 TO Len(strText)
                    IF
                        Condition:
                        MID(strText,J,1):=MID(txt.Text,j,1)
                    SET:    mMatchCount=mMatchCount+1
                    Else
                    SET:
        mMismatchedCount(i)=mMismatchedCount+1
                    END IF
                LOOP BACK
            LOOP BACK

        LOOP FOR
        Condition: i=1 TO 7
            OUTPUT MSG: String Comparison Results
            IF
            Condition: i < 7
            strMessage=strMessage
        LOOP BACK
        Load: frmResult
        SET:=frmResult.lbl.Caption:=strMessage
        SHOW: frmResult
        HIDE: Current Form
        END
```

**Coding Form compare Sequence**

```
Private Sub LoadFile()
  Dim strText As String
  txt.Text = ""
  strText = Space(FileLen(txtFile.Text))
  Open txtFile.Text For Binary As #1
   Get #1, , strText
```

```
 Close #1
 txt.Text = strText
End Sub
---------------------------------------
Private Sub cmdBrowse_Click()
 Dim strFileName As String
With dlg
   .Filter = "Data File|*.TXT|All Files|*.*"
   .ShowOpen
   strFileName = .FileName

   If strFileName <> "" Then
     txtFile.Text = strFileName
     LoadFile
   End If
 End With
  End Sub
---------------------------------------
```

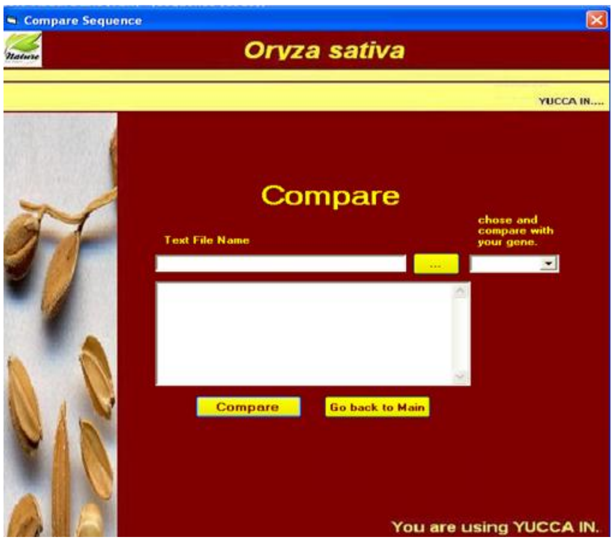


**Figure 5:** Compare form

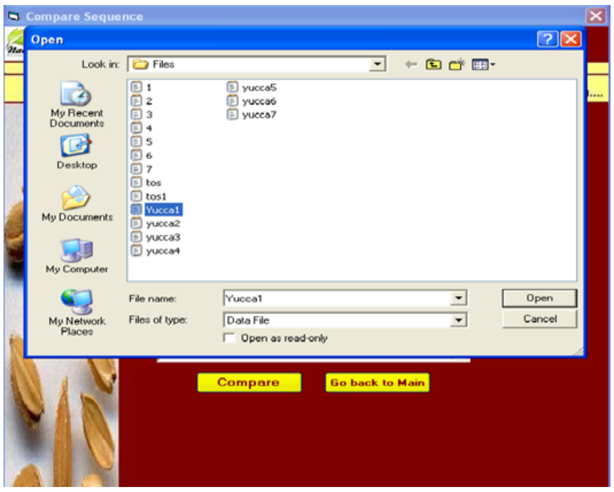


**Figure 6:** Compare browse

```
Public Function GetFileText(strFileName As String) As String
 Dim strText As String
 strText = Space(FileLen(App.Path & "\Files\" & strFileName))
 Open App.Path & "\Files\" & strFileName For Binary As #1
   Get #1, , strText
 Close #1
 GetFileText = strText
 End Function
```

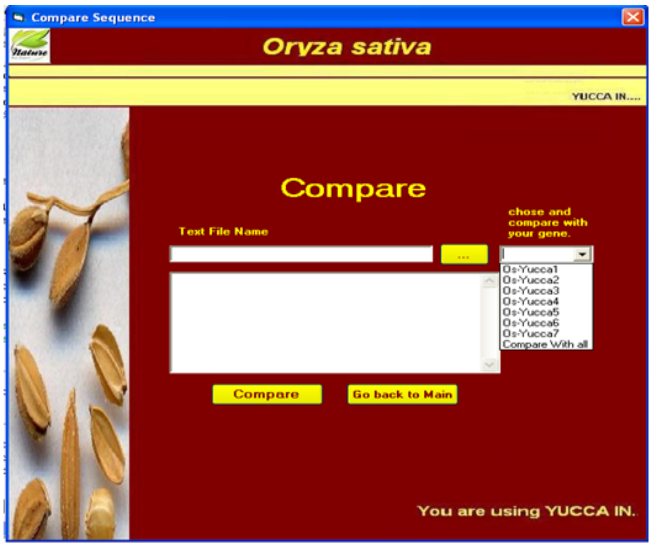
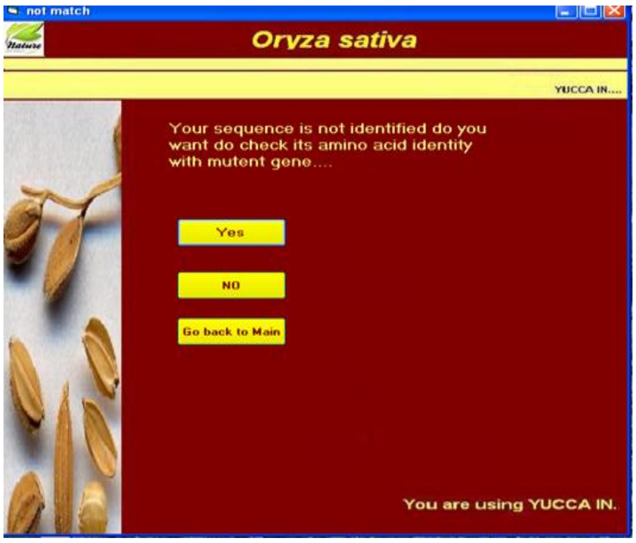


**Figure 7**: Compare with Os-Yucca (1-7)

```
Private Sub cmdCompare_Click()
 Dim mMatchNumber As Integer
  Dim mTotalCharacterCount As Long
 Dim mMatchCount(7) As Long, mMisMatchedCount(7) As Long, mCCount(7) As Long

 Dim strText As String
 Dim strFileName As String
 Dim i As Integer, j As Long

 Dim intIndex As Integer
 intIndex = cboCompare.ListIndex
 Dim strFile As String

 If intIndex = 7 Then
   GoTo CompareALL
 End If

 strFileName = "yucca" & intIndex + 1 & ".txt"
 strText = GetFileText(strFileName)
 mTotalCharacterCount = Len(strText)

 If txt.Text = strText Then
   Select Case intIndex

     Case 0
       frmYUCCA1.Show
     Case 1
       frmYUCCA2.Show
     Case 2
```

```
      frmYUCCA3.Show
     Case 3
      frmYUCCA4.Show
     Case 4
      frmYUCCA5.Show
     Case 5
      frmYUCCA6.Show
     Case 6
      frmYUCCA7.Show
      Me.Hide
    End Select
    Exit Sub

   Else
    Load notmatch
    notmatch.mMisMatchFile = strFileName
    notmatch.mstrUserText = txt.Text
    notmatch.Show
    Me.Hide
    End If
    Exit Sub
```



Your sequence is not identified do you want do check its amino acid identity with mutent gene....

**Figure 8:** Not match Form

```
CompareALL:
     For i = 1 To 7
    strFileName = "yucca" & i & ".txt"
    strText = GetFileText(strFileName)
    mTotalCharacterCount = Len(strText)
    If txt.Text = strText Then
   'MsgBox "Matched with " & strFileName
     mMatchNumber = i
     Exit For
     End If
     Next
      For i = 1 To 7
     strFileName = "yucca" & i & ".txt"
     strText = GetFileText(strFileName)
     mTotalCharacterCount = Len(strText)
     mCCount(i) = mTotalCharacterCount
       mMatchCount(i) = 0
       mMisMatchedCount(i) = 0

       For j = 1 To Len(strText)
        If Mid(strText, j, 1) = Mid(txt.Text, j, 1) Then
         mMatchCount(i) = mMatchCount(i) + 1
        Else
         mMisMatchedCount(i) = mMisMatchedCount(i) + 1
        End If
       Next

      Next

  Dim strMessage As String

  For i = 1 To 7
   strMessage = strMessage & "Os-Yucca " & i & " Matched = " & Format(mMatchCount(i) / mCCount(i) * 100, "0.00") & " %  -  Mismatched = " & Format(100 - (mMatchCount(i) / mCCount(i) * 100), "0.00") & " %"
   If i < 7 Then strMessage = strMessage & vbCrLf
  Next

  Load frmResult
  frmResult.lbl.Caption = strMessage
  frmResult.Show , Me
  Me.Hide
  ' strMessage
   'txt.Text = ""
  End Sub
Private Sub cmdgobacktomain_Click()
If cmdgobacktomain.Value = True Then Load advmain
advmain.Show
Me.Hide
End Sub
```

**Algorithm of amino acid identity**

```
EVENT NAME: Change of Dir1
Argument:NOne
Variable:None
Begin
        SET: Label1.Visible = True
        SET: Label1.Caption = Dir1.Path

End
--------------------------------------
EVENT NAME: Change of Dir1
Argument:NOne
Variable:None
Begin
        SET: Label2.Visible = True
        SET: Label2.Caption = Dir2.Path
End
--------------------------------------
EVENT NAME: Change of Dir1
Argument:NOne
```

```
Variable:None
Begin
        SET: Label3.Visible = True
        SET: Label3.Caption = Dir3.Path

End
-------------------------------------
Event Name: Change of Combo
Argument: None
Variable:
        mMatchNumber Type Intiger
        mTotal CharacterCount Type Long
        strText Type String
        strFileName Type String
        i Type Intiger
        j type Long
Begin:

        CALL: CheckFile()
        SET:strText := GetFileText(mMisMatchFile)
        SET:mTotalCharacterCount := Len(strText)
        SET:lblCharacterLength.Caption := "Total
Characters:"& mTotalCharacterCount

    LOOP FOR j:=1 To Len(strText)
        IF
        Condition:
                SET:Mid(strText, j, 1) = Mid(mstrUserText, j,
1)
        ElSE
                SET:txtMisMatch.Text = txtMisMatch.Text
& Mid(strText, j, 1)
        EndIF
    LOOP BACK

END
-------------------------------------
Func Name: GetFileText
Argument: strFileName
Variables:
        strText Type String
Begin:
    SET:    strText = Application Path
    GET:    strText in FILE address
    CLOSE:    FILE
    SET:    GetFileText = strText
END
```

**Coding frm amino acid identity**

```
Public mMisMatchFile As String
Public mstrUserText As String
Private Sub Check1_Click()
Dir1.Visible = True
End Sub
-------------------------------------
Private Sub Check2_Click()
Dir2.Visible = True
End Sub
-------------------------------------
Private Sub Check3_Click()
Dir3.Visible = True
End Sub
-------------------------------------
Private Sub Command1_Click()
Unload Me
Form3.Show
Form3.Show
End Sub
-------------------------------------
Private Sub Dir1_Change()
Label1.Visible = True
Label1.Caption = Dir1.Path
End Sub
-------------------------------------
Private Sub Dir2_Change()
Label2.Visible = True
Label2.Caption = Dir2.Path
End Sub
-------------------------------------
Private Sub Dir3_Change()
Label3.Visible = True
Label3.Caption = Dir3.Path
End Sub
-------------------------------------
Private Sub ImageCombo1_Change()
End Sub
 Public Sub CheckFile()
  Dim mMatchNumber As Integer
  Dim mTotalCharacterCount As Long
  Dim strText As String
  Dim strFileName As String
  Dim i As Integer, j As Long
  strText = GetFileText(mMisMatchFile)
   mTotalCharacterCount = Len(strText)
  lblCharacterLength.Caption = "Total Characters : " &
mTotalCharacterCount
   For j = 1 To Len(strText)
      If Mid(strText, j, 1) = Mid(mstrUserText, j, 1) Then
        txtMatch.Text = txtMatch.Text & Mid(strText, j, 1)
      Else
        txtMisMatch.Text = txtMisMatch.Text &
Mid(strText, j, 1)
      End If
    Next
  lblPercent.Caption = "Match " &
Round(Len(txtMatch.Text) / (Len(txtMisMatch.Text) +
Len(txtMatch.Text)) * 100, 2) & " %"
    lblMisMatchPercent.Caption = "Mis-match Percent " &
Round(100 - (Len(txtMatch.Text) /
(Len(txtMisMatch.Text) + Len(txtMatch.Text)) * 100), 2) &
" %"
End Sub
```

```
--------------------------------------
Public Function GetFileText(strFileName As String) As
String
 Dim strText As String
 strText = Space(FileLen(App.Path & "\Files\" &
strFileName))
 Open App.Path & "\Files\" & strFileName For Binary As
#1
 Get #1, , strText
 Close #1
 GetFileText = strText
 End Function
 Private Sub cmdback_Click()
If cmdback.Value = True Then Load frmCompare
frmCompare.Show
End Sub
```

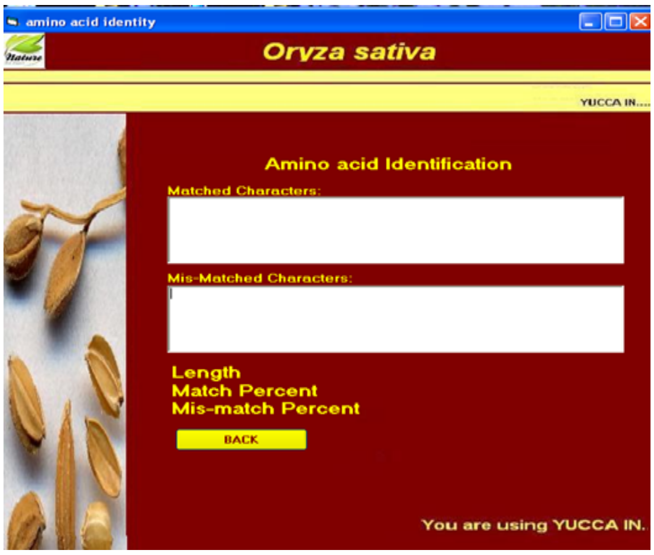


**Figure 9:** Amino acid identity

## Results

Results are taken from the YUCCA-IN by comparing YUCCA genes with the others present in database. YUCCA IN compares different genes with over expressed genes and shows amino acid identity match and mismatch.

## Conclusions

YUCCA IN compares different genes with over expressed genes and shows amino acid identity match and mismatch and also phenotypic abnormality of respective gene in Visual basic that greatly simplifies window application development. It saves time and reduces errors to its minimum level as compare with analytical approach by using dynamic programming algorithm.

## References

[1]. Teale W.D, Papono, Palme K (2006) Auxin in action: signalling, transport and the control of plant growth and development. Nat Rev Mol Cell Biol 7: 847–859.
[2]. Zazimalova E, Napier RM (2003). Points of regulation for auxin action. J Plant Cell Rep 21: 625–634.
[3]. Cheng Y, Dai X, Zhao Y (2006) Auxin biosynthesis by the YUCCA flavin monooxygenases controls the formation of floral organs and vascular tissues in Arabidopsis. J Genes and Dev 20: 1790-1799.
[4]. Brudno M, Malde P, Poliakov, Couronne O, Dubchak I, Batzoglou S (2003) Glocal alignment: finding rearrangements during alignment. J Bioinformatics 19: 54–62.
[5]. Wang L, Jiang T (1994) On the complexity of multiple sequence alignment. J Comput Biol 1: 337-348.
[6]. Lien DA (1986) The Basic Handbook: Encyclopedia of the BASIC Computer Language (3rd ed.).(Compusoft Publishing) pp 21-24