AWD suitability mapping methodology*

The Alternate Wetting and Drying (AWD) practice has been proved to be an innovative practice in rice cultivation that significantly reduces 30% of total water use and 48% of total CH4 emissions without impacting rice yield (Richards and Sander, 2014). To support AWD out-scaling, the methodology for AWD suitability analysis developed has been developed by Nelson et al. (2015) and Sander et al. (2017). This suitability analysis mainly based on soil and climatic factors and specific conditions in rice production regions. The process of AWD suitability analysis is as follows:

1. Water balance analysis: developing a rice extent and calculating the dekadal water balance for each area of the rice region based on dekadal rainfall, potential evapotranspiration, and potential soil percolation rate;
2. Dekadal AWD suitability analysis: calculating AWD dekadal suitable score to determine when field water is deficit or surplus;
3. Seasonal AWD suitability analysis: calculating the AWD seasonal suitable score and categorized into three classes corresponding to three seasonal suitability levels: high suitability, moderate suitability, and low suitability.

In Vietnam, the government has prioritized AWD as an option for NDC implementation in the agricultural sector. To support national and local AWD implementation plan, the AWD suitability analysis methodology has been adapted for rice production condition of Vietnam.

The seasonal AWD suitability maps were first developed for national level and then down-scaled to provincial level. An Giang province was selected as a case study and the bio-physical AWD suitability maps were first developed for the province in 2018. Based on these maps, the staffs of An Giang province's Department of Agriculture and Rural Development were engaged to analyze local situation and adoption capacity, and identify (socio-economic) barriers that obstruct large scale adoption of AWD at its districts. Based on this participatory work, the bio-physical suitability and socio-economic adoption capacity have been integrated to develop the overall AWD suitability maps of the province. The outputs were validated by local officials, and considered as scientific reference for AWD scaling strategy in rice production of the province.

To support the bio-physical suitability analysis, IRRI has developed computer-based tool, namely Mapping suitability of the Alternate Wetting and Drying practice in rice production (MapAWD).

MapAWD tool development

The MapAWD is a Microsoft® Excel-based tool. The tool only requires simple input data (e.g. rice extent, cropping season, rainfall, potential evapotranspiration and soil percolation rates) and can in principle be applied to all rice growing areas. The original method has been improved by integrating climate-risks and unfavorable soil information in its analysis process.

The tool also integrates a GIS component to present the spatial distribution of suitability levels in the studied area. The tool uses the map of rice extent in raster format as a base map for all analysis steps.

*Description of the AWD suitability mapping methodology taken from a manuscript that is in the editing process.

System requirements

The MapAWD tool can operate in the following environment:

- Operation system: Windows XP, Windows 7 and later versions
- Microsoft Excel 2003 or later
- Disk space requirement: 7 Mb
- Memory requirement: 128 Mb RAM

Components of MapAWD

The core of MapAWD is an Excel workbook with programing codes written using Visual Basic for Application (VBA) language. MapAWD includes components for input and output data. The structure of MapAWD is described in the table below:

MapAWD components	Description
…\MapAWD \Inputs Crop.txt Land.txt PET.txt Rainfall.txt BaseMap.asc	Working folder Input folder contains required bio-physical data for suitability analysis and mapping. Input data is in tabular format (climate and land use data) and the base map is in ESRI ASCii format (*.asc)
\Outputs	Output folder
MapAWD v1.0.xls	MapAWD program and interface
MapAWD_Users_ manual.doc	Users' manual

VBA code for AWD suitability analysis

Workbook

Private Sub Workbook_Open()

```
ThisWorkbook.Sheets(1).Range("WorkDir").Value = ThisWorkbook.Path & "\"
```

End Sub

Sheet GeneralInfo

Private Sub Worksheet_Activate()

```
Me.Range("WorkDir").Value = ThisWorkbook.Path
```

End Sub

Modul AWD

```
Option Explicit
Public StrVal As Variant, iTmp As Long
Public WorkDir As String
Public LandFile As String
Public CropFile As String
Public RainFile As String
Public PetFile As String
Public SuitFile As String
Public SuitMap As String

Public Type GridHeader
   nCols As Long ' number of columns in grid map
   nRows As Long ' number of rows in grid map
   xllcorner As Double
   yllcorner As Double
   CellSize As Single
   Nodata As Integer
   Count As Double
   'MinX As Double, MaxX As Double, MinY As Double, MaxY As Double
End Type
Public Type GridMap ' Store spatial variables
   Header As GridHeader
   Data() As Variant
End Type
Public DefaultGridHeader As GridHeader
Const BarWidth = 275 '.Shapes("StatusBar").Width
```

```
Sub SuitabilityAnalysis()
    Dim BaseMap As GridMap
    Dim AWDMap() As GridMap

    '---------------------------------------
    Dim ColNo As Long, RowNo As Long
    Dim nSeason As Integer 'number of crop seasons per year
    Dim RainData As Variant ' Rainfall data
    Dim Percolation As Variant ' Percolation data
    Dim PetData As Variant ' Potential evapo-transpiration data
    Dim WB As Single 'water balance = rainfall- PET -Percolation
    Dim AWDScore As Boolean 'AWD suitability
    Dim TotalScore As Integer ' Total score
    Dim ScoreThresholds(3) As Single
    Dim RiskDur(1, 1) As Integer '(Climate risk duration, dekad (0= saline, 1 = flood))
    Dim RiskSaline As Integer, RiskFlood As Integer, RiskSoil As Integer
    Dim PlantD() As Integer 'Planting dekad (season)
    Dim HarvestD(2) As Integer ' harvesting dekad (season)
    Dim nDekad(2) As Integer ' number of dekad in a season (season)
    Dim Irr As Single ' Irrigation water (mm)
    Dim StrLine As String
    Dim StrVal As Variant
    Dim StrOut As String
    Dim iCount As Integer
    Dim jCount As Double
    Dim s As Integer ' season
    Dim d As Integer 'dekad
    Dim BarRatio As Integer
    '-------------------------------------------------------------------------
    nSeason = 3
    ReDim PlantD(nSeason - 1)

    With ThisWorkbook.Sheets(1)
        WorkDir = .Range("WorkDir").Value
        CropFile = WorkDir & "\input\" & .Range("CropFile").Value
        LandFile = WorkDir & "\input\" & .Range("LandFile").Value
        RainFile = WorkDir & "\input\" & .Range("RainFile").Value
        PetFile = WorkDir & "\input\" & .Range("PETFile").Value

        RiskDur(0, 0) = .Range("SalineStart").Value: RiskDur(0, 1) = .Range("SalineEnd").Value
        RiskDur(1, 0) = .Range("FloodStart").Value: RiskDur(1, 1) = .Range("FloodEnd").Value

        SuitFile = WorkDir & "\output\" & ThisWorkbook.Sheets(1).Range("OutputFile").Value
        SuitMap = ThisWorkbook.Sheets(1).Range("OutputMap").Value
```

```
End With

With ThisWorkbook.Sheets(2)
   DefaultGridHeader.CellSize = .Range("CellSize").Value
   DefaultGridHeader.nCols = .Range("nCols").Value
   DefaultGridHeader.nRows = .Range("nRows").Value
   DefaultGridHeader.xllcorner = .Range("xllcorner").Value
   DefaultGridHeader.yllcorner = .Range("yllcorner").Value
   DefaultGridHeader.Nodata = .Range("NoData").Value
   DefaultGridHeader.Count = .Range("nCells").Value
End With

ReDim AWDMap(nSeason - 1)
For s = 0 To nSeason - 1
   AWDMap(s).Header = DefaultGridHeader
   ReDim AWDMap(s).Data(DefaultGridHeader.nCols, DefaultGridHeader.nRows)
   '"Season " & s + 1, GridHeader, FloatDataType, -9999, InRam
Next s

With ThisWorkbook
   ScoreThresholds(1) = .Sheets(1).Range("upper1").Value
   ScoreThresholds(2) = .Sheets(1).Range("upper2").Value
   ScoreThresholds(3) = .Sheets(1).Range("upper3").Value
   updateStatus "Processing... Please wait...", 0
End With
'-----------------------
Open LandFile For Input As #1
Open RainFile For Input As #2
Open PetFile For Input As #3
Open CropFile For Input As #4
Open SuitFile For Output As #5

'startTime = Timer()
jCount = 0
Do Until EOF(1) = True
        DoEvents
        StrVal = 0: ColNo = 0: RowNo = 0: Percolation = 0
        RainData = 0: PetData = 0

        Line Input #1, StrLine ' open land data
        StrVal = Split(StrLine, ",")
        ColNo = Val(StrVal(1))
        RowNo = Val(StrVal(2))
        Percolation = Val(StrVal(5))
```

```
RiskSoil = Val(StrVal(8)) ' Acid sulphate soil
RiskSaline = Val(StrVal(6)) ' Saline affected land
RiskFlood = Val(StrVal(7)) ' flood affected land
jCount = jCount + 1
If Int(jCount / DefaultGridHeader.Count * 100) > BarRatio Then
      BarRatio = Int(jCount / DefaultGridHeader.Count * 100)
      updateStatus "Processing..." & BarRatio & "%", BarRatio
End If
Line Input #2, StrLine ' open rainfall data
   RainData = Split(StrLine, ",")
Line Input #3, StrLine ' open PET data
   PetData = Split(StrLine, ",")
Line Input #4, StrLine ' open crop data
If IsNumeric(PetData(0)) = False Then GoTo SkipLine

StrVal = Split(StrLine, ",")
For s = 0 To nSeason - 1
      nDekad(s) = 0: TotalScore = 0: AWDMap(s).Data(ColNo, RowNo) = 0
      PlantD(s) = StrVal(s * 2 + 1)
      HarvestD(s) = StrVal(s * 2 + 2)
      'Calculate crop duration, ndekad
      If PlantD(s) = 0 Or HarvestD(s) = 0 Then
                  nDekad(s) = 0
                  ElseIf PlantD(s) > HarvestD(s) Then
                  nDekad(s) = 37 - PlantD(s) + HarvestD(s) + 1
                  ElseIf PlantD(s) < HarvestD(s) Then
                  nDekad(s) = HarvestD(s) - PlantD(s) + 1
        Else
                  nDekad(s) = 0
      End If

      'Calculate water balance (WB)
      If nDekad(s) = 0 Then AWDMap(s).Data(ColNo, RowNo) = -2: GoTo SkipSeason
      iCount = 0
      For d = 3 To nDekad(s) - 1 ' ignore 2 first dekads and the last dekad.
            'Check unapplication cases
            If RiskSoil = 1 Then 'Acid sulphate soil
                  AWDMap(s).Data(ColNo, RowNo) = -1:  GoTo SkipSeason
            ElseIf RiskSaline = 1 Then 'Saline affected land
                  If d >= RiskDur(0, 0) And d <= RiskDur(0, 1) Then
                              AWDScore = False: GoTo SkipDekad
                  End If
            ElseIf RiskFlood = 1 Then 'Flood affected land
                  If d >= RiskDur(1, 0) And d <= RiskDur(1, 1) Then
```

```
                        AWDScore = False: GoTo SkipDekad
                    End If
                End If

                WB = 0: AWDScore = 0
                if PlantD(s) + d <= 37 Then
                    If  Val(RainData(PlantD(s) + d)) >= 5 Then Irr = 0 Else Irr = 5 -
                        Val(RainData(PlantD(s) + d)) ' if rain water >=5mm then no need to
                        irrigate
                        WB = Val(RainData(PlantD(s) + d)) + Irr - Val(PetData(PlantD(s) + d))
                        - Val(Percolation)
                    Else
                      If Val(RainData(PlantD(s) + d - 37 + 1)) >= 5 Then Irr = 0 Else Irr = 5 -
                        Val(RainData(PlantD(s) + d - 37 + 1)) ' if rain water >=5mm then no
                        need to irrigate
                        WB = Val(RainData(PlantD(s) + d - 37 + 1)) + Irr -
                        Val(PetData(PlantD(s) + d - 37 + 1)) - Val(Percolation)
                      End If
                    If WB < 0 Then AWDScore = True Else AWDScore = False
                SkipDekad:
                    TotalScore = TotalScore + Abs(AWDScore)
                    iCount = iCount + 1
            Next d
            'Calculate AWD suitability
            Select Case TotalScore / iCount ' ignore 2 first dekads and the last dekad.
              Case 0: AWDMap(s).Data(ColNo, RowNo) = 0 ' no suitable
              Case Is <= ScoreThresholds(1): AWDMap(s).Data(ColNo, RowNo) = 1 ' low suitable
              Case Is <= ScoreThresholds(2): AWDMap(s).Data(ColNo, RowNo) = 2 ' medium
        suitable
              Case Else: AWDMap(s).Data(ColNo, RowNo) = 3 ' high suitable
            End Select

        SkipSeason:
            If s = 0 Then
              StrOut = AWDMap(s).Data(ColNo, RowNo)
            Else
              StrOut = StrOut & "," & AWDMap(s).Data(ColNo, RowNo)
            End If
          Next s
          Print #5, Val(PetData(0)) & "," & ColNo & "," & RowNo & "," & StrOut
        SkipLine:
Loop
Close #1: Close #2: Close #3: Close #4: Close #5
If ThisWorkbook.Sheets(1).Range("opt_OutputMap") = "Yes" Then
```

```
        If ThisWorkbook.Sheets(1).Range("outputMap").Value <> "" Then
          For s = 0 To nSeason - 1
            updateStatus "Writing outputs for season " & s + 1, 100
            ExportAsc s, AWDMap(s) ' Write asc format of AWD map
          Next s
        End If
      End If
      'Erase AWDMap
      MsgBox "Process completed", vbOKOnly + vbInformation, "Done!"
      updateStatus "Ready!", 0
End Sub

Sub ExportAsc(s As Integer, ascMap As GridMap)
      Dim iTmp As Long
      Dim i As Integer, j As Integer
      Dim iFile As Integer
      Dim OutMapFile As String
      iFile = FreeFile

      OutMapFile = WorkDir & "\output\" & "Season_" & (s + 1) & "_" & SuitMap

      Open OutMapFile For Output As #iFile
      With ascMap.Header
        Print #1, "ncols" & Space(11) & .nCols
        Print #1, "nrows" & Space(11) & .nRows
        Print #1, "xllcorner" & Space(11) & .xllcorner
        Print #1, "yllcorner" & Space(11) & .yllcorner
        Print #1, "CellSize" & Space(11) & .CellSize
        Print #1, "NODATA_value" & Space(11) & .Nodata

      iTmp = 0
      DoEvents
      For j = 1 To .nRows  ' Y
        DoEvents
        For i = 1 To .nCols ' X
          If IsEmpty(ascMap.Data(i, j)) Then ascMap.Data(i, j) = .Nodata
          If i < .nCols Then
              Print #1, ascMap.Data(i, j);
          Else
              Print #1, ascMap.Data(i, j)
          End If
        Next
      Next
      End With
```

```
EndProcess:
Close #iFile

End Sub
Sub OpenBaseMap()
Dim PreviousName As String
Dim InMapFile As String
Dim iFile As Integer
Dim iCount As Double

PreviousName = ThisWorkbook.Sheets(1).Range("BaseMap").Value

ChDir ThisWorkbook.Path
InMapFile = Application.GetOpenFileName("ESRI ASCII raster format(*.asc), *.asc", 1, "Open
base map for reference")
If InMapFile <> "False" Then
   ThisWorkbook.Sheets(1).Range("BaseMap").Value = InMapFile

'--------ReadAscMap InMapFile
   iFile = FreeFile
   Open InMapFile For Input As #iFile
   updateStatus "Reading basemap....please wait!", 0
   With ThisWorkbook.Sheets(2)
      For iCount = 1 To 6
      Line Input #iFile, StrVal ' Read Arc Map description
      Select Case iCount
         Case 1: .Range("nCols") = Val(Mid(StrVal, InStr(1, StrVal, " "), 100))
         Case 2: .Range("nRows") = Val(Mid(StrVal, InStr(1, StrVal, " "), 100))
         Case 3: .Range("xllcorner") = Val(Mid(StrVal, InStr(1, StrVal, " "), 100))
         Case 4: .Range("yllcorner") = Val(Mid(StrVal, InStr(1, StrVal, " "), 100))
         Case 5: .Range("CellSize") = Val(Mid(StrVal, InStr(1, StrVal, " "), 100))
         Case 6: .Range("NoData") = Val(Mid(StrVal, InStr(1, StrVal, " "), 100))
      End Select
      Next
   iCount = 0
   Do Until EOF(iFile) = True
   DoEvents
      Line Input #iFile, StrVal
      iCount = iCount + .Range("nCols") - (Len(StrVal) - Len(Replace(StrVal, .Range("NoData"),
""))) / Len(.Range("NoData"))
      updateStatus "Reading base map..." & iCount, 100
   Loop
      .Range("nCells") = iCount
```

```
        End With
        Close #iFile
    '--------
    Else
        ThisWorkbook.Sheets(1).Range("BaseMap").Value = PreviousName
    End If

    InMapFile = Empty

    updateStatus "Ready!", 0
End Sub
Sub GetWorkingDirectory()
    Dim PreviousName As String
    Dim fldr As FileDialog
    Dim sItem As String

    PreviousName = ThisWorkbook.Sheets(1).Range("WorkDir").Value
        Set fldr = Application.FileDialog(msoFileDialogFolderPicker)
        With fldr
            .Title = "Select a Folder"
            .AllowMultiSelect = False
            .InitialFileName = ThisWorkbook.Path 'Application.DefaultFilePath
            If .Show <> -1 Then
                sItem = PreviousName
                GoTo NextCode
            Else
                sItem = .SelectedItems(1)
            End If
        End With
    NextCode:
        ThisWorkbook.Sheets(1).Range("WorkDir").Value = sItem
        Set fldr = Nothing
    End Sub
    Public Function GetFileList(ByVal FileExt As String, ByVal iFolder As String) As String()
    Dim FSO As Object, iFile As Object
    Dim FSOSource As Object
    Dim iArray() As String
    Dim i As Integer

    If Dir(LCase(iFolder)) = "" Then GoTo ReadingError
        Set FSO = CreateObject("Scripting.FileSystemObject")
        Set FSOSource = FSO.GetFolder(iFolder)
    i = 0
    For Each iFile In FSOSource.Files
```

```
    If LCase(Mid(iFile.Name, InStr(1, iFile.Name, ".", vbTextCompare) + 1, Len(iFile.Name))) =
LCase(FileExt) Then
        ReDim Preserve iArray(i)
        iArray(i) = Mid(iFile.Name, 1, InStr(1, iFile.Name, ".", vbTextCompare) - 1)
        i = i + 1
    End If
Next
GetFileList = iArray
Erase iArray
Exit Function
ReadingError:
    MsgBox "The folder " & iFolder & " does not exist. Check for structure of your installation",
vbExclamation + vbOKOnly, "WBTools"
End Function
Sub updateStatus(Text As String, Ratio As Integer)
    ThisWorkbook.Sheets(1).Range("Status").Value = Text
    ThisWorkbook.Sheets(1).Shapes("StatusBar").Width = Ratio / 100 * BarWidth
End Sub
```