Supplementary data for

"Image analysis of stress-induced lignin deposition in *Arabidopsis thaliana* using the macro program LigninJ for ImageJ software"

Masato Nakamura, Tomoaki Kamehama, Yasushi Sato

This file includes

Supplementary Figure S1

Supplementary Figure S2.

```
macro "LigninJ" {
requires("1.52q");

d = getDirectory("Choose a Directory");
Dialog.create("Parameters");
Dialog.addNumber("Known distance:", 100);
Dialog.addString("Unit:", "um");
Dialog.addNumber("Pixels of known distance:", 73.9);
Dialog.addNumber("Lower threshold of a* :", 20);
Dialog.addNumber("Lower size limits of lignified area (Unit^2) :", 20);
Dialog.show();
D=Dialog.getNumber();
Unit=Dialog.getString();
P=Dialog.getNumber();
LT=Dialog.getNumber();
LS=Dialog.getNumber();

files = getFileList(d);
    for (i = 0; i < files.length; i++) {
    f = d + files[i];
    doLignin(f);
    }

function doLignin(f) {
setBatchMode("hide");
open(f);

run("Set Scale...", "distance=P known=D pixel=1 unit=Unit global");
rename("copy");
origBit = bitDepth;
if (bitDepth() != 24) exit;

//Reduce picture to 1/50 for background correction
run("Averaging Reducer", "x=50 y=50");
    max = 0;
    width = getWidth(); height = getHeight();
    for (y=0; y<height; y++) {
        if (y%20==0) showProgress(y, height);
        for (x=0; x<width; x++) {
            value = getPixel(x,y);
            if (value>max) {
                max = value;
                xmax = x;
                ymax = y;
            }
        }
    }
selectWindow("Reduced copy");
run("RGB Stack");
val = newArray(3);
for (s=1;s<=3;s++) {
    setSlice(s);
    val[s-1] = getPixel(xmax,ymax);
    }
run("Close");

selectWindow("copy");
```

```
run("RGB Stack");
for (s=1; s<=3; s++) {
    setSlice(s);
    dR = 255/val[s-1];
    run("Multiply...","value=dR slice");
    }
run("Convert Stack to RGB");
rename("correct");
newpath = File.directory + File.separator + File.nameWithoutExtension+"_1.jpg";
saveAs(".jpg", newpath);
run("Lab Stack");
run("Next Slice [>]");
run("Threshold...");
setThreshold(LT, 128.0000);
run("Set Measurements...", "area redirect=None decimal=3");
run("Analyze Particles...", "size=LS-infenity summarize add slice");

selectWindow("correct");
newpath = File.directory + File.separator + File.nameWithoutExtension+"_2.jpg";
saveAs(".jpg", newpath);
run("Set Measurements...", "area mean redirect=None decimal=3");
if (roiManager("count") > 0) {
    roiManager("Multi Measure append");
    roiManager("Delete");
    } else {
    makeRectangle(1,1,1,1);
    roiManager("add");
    run("Set Measurements...", "area redirect=None decimal=3");
    roiManager("Multi Measure append");
    roiManager("Delete");
    }
run("Close All");
selectWindow("Threshold");
run("Close");
}

selectWindow("Summary of correct");
newpath = File.directory + File.separator + "Summary.csv";
saveAs("results", newpath);
run("Close");
selectWindow("Results");
newpath = File.directory + File.separator + "Results.csv";
saveAs("results", newpath);
run("Close");
}
```

Supplementary Figure S1. Contents of the LigninJ macro program. The values and unit highlighted in yellow are default and can be adjusted to the respective projects. The notation "//" indicates explanatory comments.

```
Sub Results_calculation()

    Sheets("Results").Select
    Sheets("Results").Copy After:=Sheets(1)
    Sheets("Results (2)").Select
    Sheets("Results (2)").Name = "Results Calculation"

    Range("B:E").Insert
    Range("B1") = "No"
    Range("C1") = "Lab"
    Range("D1") = "Total Area"
    Range("E1") = "Index"
MaxRow = Cells(Rows.Count, 1).End(xlUp).Row
MaxCol = Cells(1, Columns.Count).End(xlToLeft).Column
    Range("C2") = "L"
    Range("C3") = "a"
    Range("C4") = "b"

Range("C2:C4").Copy Destination:=Range("C2:C" & MaxRow)

    Dim No As Long
    No = 0
    Do While No < MaxRow - 1
    Cells(No + 2, 2) = No / 3 + 1
    Cells(No + 3, 2) = No / 3 + 1
    Cells(No + 4, 2) = No / 3 + 1
    No = No + 3
    Loop

    Dim Rnt As Long
    Rnt = 2
    Do While Rnt <= MaxRow

     If Cells(Rnt, 6) < 10 Then
  Cells(Rnt, 6) = 0
    End If

     Dim Cnt As Long
     Cnt = 6
     M2 = 0
     T2 = 0
     Do While Cnt < MaxCol
     T1 = Cells(Rnt, Cnt)
     T2 = T2 + T1

      If Cells(Rnt, 3) = "L" Then
      M1 = Cells(Rnt, Cnt) * (100 - Cells(Rnt, Cnt + 1))
      M2 = M2 + M1
      Cnt = Cnt + 2
      Cells(Rnt, 4) = T2
      Cells(Rnt, 5) = M2
      Else
      M1 = Cells(Rnt, Cnt) * Cells(Rnt, Cnt + 1)
      M2 = M2 + M1
      Cnt = Cnt + 2
      Cells(Rnt, 4) = T2
      Cells(Rnt, 5) = M2
```

```
      End If
    Loop

Rnt = Rnt + 1
  Loop

    With ActiveSheet
        .Sort.SortFields.Clear
        .Sort.SortFields.Add Key:=.Range("C1"), Order:=xlAscending
        .Sort.SetRange .Range(Cells(1, 1), Cells(MaxRow, MaxCol))
        .Sort.Header = xlYes
        .Sort.Apply
End With

  End Sub
```

Supplementary Figure S2. Contents of the "Results_caculation_excel_macro" for Microsoft Excel.