

Methods S1

Program code related to STAR Methods

Analysis of Ca²⁺ imaging movies and immunohistochemical stainings was performed by using IGOR Pro 6.34 (Wavemetrics, Lake Oswego, USA) and the following, in house-written code. To use the code, you have to download IGOR Pro software (wavemetrics.com), copy it into the Procedure window and compile. Please note that there might be naming conflicts when compiling all packages are compiled together in one IGOR file. Please, thus, use the packages separately. There is currently no manual to the packages. Please contact the author of the code (Marcus Semtner: marcus.semtner@mdc-berlin.de) For questions.

Packages:

1. [Ca²⁺ imaging Primary analysis](#)

Monochrome movies (8bit or 16bit) are loaded into IGOR, and ROIs can be selected graphically. The package stores annotated Ca²⁺ traces within the IGOR-intrinsic folder structure.

2. [Ca²⁺ imaging secondary analysis](#)

Annotated experiments from the primary analysis are combined. Software distinguished responding from non-responding cells, generates merged views of Ca²⁺ responses from different experiments and calculates the analysis and summary files For export into Excel.

3. [IHC-GABABR in microglia](#)

Simple software to 3D-analyze GABABR signals in Iba1 volumes. Confocal stacks are loaded into IGOR. Single microglia are selected in a projection image by setting a ROI, and the software automatically generates a 3D, Iba-1-based mask under the ROI and calculates a GABABR signal threshold as well as the number of GABABR pixels within the Iba1 volume.

4. [IHC-GABA concentration in astrocytes](#)

Simple software to 3D-analyze GABA signals in Gfap volumes. Confocal stacks are loaded into IGOR. Single astrocytes are selected in a projection image by setting a ROI, and the software automatically generates a 3D, Gfap-based mask under the ROI and calculates the mean of GABA signals within the Gfap volume. Note that the nomenclature of the buttons has not been entirely updated to the specific task: astrocytes and GABA. Thus, “Iba” instead of “gfap” and “GABAB” instead of “GABA” are used.

Package 1: Ca2+ imaging Primary analysis (30.08.21)

```
#pragma rtGlobals=3           // Use modern global access method and strict wave access.

Menu "analysis"
  "_"
  "getROIs/0", f_val()
End

Function monitor(str)
  string str
  //this function is implemented to monitor the series of Function that is called after a user action
  //to activate, create Variable/G ismonitor=1
  NVAR ismonitor
    If (ismonitor)
      print str
    EndIf
End

Function ButtonProc_1killMC(ctrlName) : ButtonControl
  String ctrlName
  wave mCIMG
  killwaves mCIMG
End

Function globals()
  monitor("globals()")
  make_option()
  string/G picN=""
  string /G ROIMETH="Rectangle;Fit;Manual;Auto"
  string/G chan1,chan2,chan3,chan4
  wave/T wo=root:w_options
  variable ismonitor=0
  variable/G zmin,zmax
  Variable/G thr=str2num(wo[2][1])
  Variable/G sbsh=str2num(wo[0][1]), sbshv=str2num(wo[1][1])//boxsize
  Variable /G speed=100
  Variable/G numchan=1, smpl=1
  Variable/G cellsize=str2num(wo[3][1])
  Variable/G MaxCellsize=str2num(wo[4][1])
  make/O/N=(150,3)/T listW= ""
  make/O/N=(150,3) selW= 0
  variable/G imgFilter=1
  variable p1
  For(p1=0;p1<6;p1+=1)
    string/G root:$("s"+num2str(1+p1))=wo[23+3*p1][1]
    string/G root:$("e"+num2str(1+p1))=wo[24+3*p1][1]
    string/G root:$("w"+num2str(1+p1))=wo[22+3*p1][1]
    variable/G root:$("r"+num2str(1+p1))
    variable/G root:$("Perc"+num2str(0+p1))
  EndFor
End

Function fval_popOptions(lasch)
  string lasch
  make_option()
End

Function fval_popexport(lasch)
  string lasch
  exportDATA()
End
```

```

Function CheckProc_2showMc(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked
    If(checked)
wave mclMG
        If(!waveexists(mclMG))
            fQuest()
            pauseForuser w_quest
        EndIf
        appEndimage/T/W=fval#G0 mclMG
    else
        removeimage/W=fval#G0 mclMG
    EndIf
End

Function fQuest()
    dowindow/K w_quest
    PauseUpdate; Silent 1 // building window...
    NewPanel/N= w_quest /W=(764,163,1039,260) as "Define mclMG"
    PopupMenu popup0,pos={28,7},size={131,21},bodyWidth=100,title="select"
    PopupMenu popup0,mode=1,value= get2DIMGs()
    Button button0,pos={206,8},size={50,20},proc=ButtonProc,title="Load"
    Button button1,pos={113,41},size={50,50},title="DONE", proc=dupMC
End

Function/S get2DIMGs()
    return wavelist("",";",",","DIMS:2")
End

Function dupMC(lasch)
string lasch
    controlinfo/W=w_quest popup0
string wn=s_value
    duplicate/O root:$(wn), root:mclMG
    dowindow/K w_quest
End

Function F_val()
    monitor("F_val()")
    dowindow/K fval
    globals()

    dowindow/K fval

    PauseUpdate; Silent 1 // building window...
    NewPanel/W=(391,152,1671,934)/K=1/N=fval as "BUILT"
    modifypanel/W=fval fixedsize=1

    GroupBox group5,pos={0,0},size={1332,30},labelBack=(43520,43520,43520),frame=0
        Button button6,pos={31,6},size={50,20},proc=fval_popFILE,title="File"
        Button button7,pos={91,6},size={50,20},title="Edit"
        Button button8,pos={151,6},size={50,20},title="Options",proc=fval_popOptions
        Button button09,pos={211,6},size={50,20},title="Export",proc=fval_popExport
        Button button12 ,pos={1211,6},size={50,20},title="SAVE",proc=implementDATA
        Button button18,pos={519,5},size={80,20},proc=ButtonProc_1killMC,title="delete mclMG"
        CheckBox check15,pos={445,7},size={64,14},proc=CheckProc_2showMc,title="mC image"
        CheckBox check15,value= 1
//selection of Movies in file/analysed
    PopupMenu popup0,pos={15,38},size={426,21},bodyWidth=410,proc=SelPICT,title=" "
    PopupMenu popup0,mode=1,value=getpictlist()

    SetVariable setvar15,pos={595,41},size={53,16},bodyWidth=30,title="filter"
    SetVariable setvar15,value= imgFilter

    Button button0,pos={510,39},size={60,20},proc=PICnexas,title=">"

```

```

Button button1,pos={447,39},size={60,20},proc=PlCnexlas,title="<"
Button button01 title="Use Update",pos={667,39},size={70,20},proc=useUpdate

wave/T listW
NVAR zmin,zmax

///Display options
GroupBox group0,pos={579,68},size={228,197},labelBack=(0,47872,47872)
GroupBox group2,pos={590,78},size={207,45},title="Contrast"
GroupBox group2,labelBack=(30464,30464,30464)
SetVariable setvar0,pos={598,100},size={85,16},bodyWidth=60,proc=SetPICT,title="zmin"
SetVariable setvar0,limits={0,inf,0},value= zmin
SetVariable setvar1,pos={695,100},size={88,16},bodyWidth=60,proc=SetPICT,title="zmax"
SetVariable setvar1,limits={0,inf,0},value= zmax

GroupBox group1,pos={592,137},size={103,109},title="Display"
GroupBox group1,labelBack=(30464,30464,30464)
Button button3,pos={602,222},size={80,20},proc=ROIleg,title="ROIs & legEnds"
Button button4,pos={602,199},size={80,20},proc=ROIleg,title="ROIs"
Button button5,pos={602,175},size={80,20},proc=ROIleg,title="no ROIs"
CheckBox check01,pos={616,157},size={54,14},title="backgr.",value= 0

GroupBox group3,pos={697,137},size={103,109},title="ROI action"
GroupBox group3,labelBack=(30464,30464,30464)
Button button2,pos={709,213},size={80,20},proc=deleteROI,title="Delete ROI"
Button button03,pos={709,189},size={80,20},proc=Delete_allROIs,title="Delete all ROIs"
CheckBox check0,pos={719,167},size={47,14},proc=ShowMaxZ,title="zProj. "
CheckBox check0,value= 0

ListBox list0,pos={582,277},size={225,499}, mode=2,listWave=listW
ListBox list0 proc=ListBoxProc

///Play Action
Slider slider0,pos={29,514},size={541,19},proc=fvalSliderProc
Slider slider0,limits={0,250,1},value= 0,vert= 0,ticks= 0

GroupBox group6,pos={27,536},size={543,53},labelBack=(43520,43520,43520)
GroupBox group7,pos={36,544},size={232,39},labelBack=(30464,30464,30464)
Button button9,pos={48,553},size={92,21},proc=fvalPlayMovie,title="Play"
SetVariable setvar2,pos={159,555},size={81,18},bodyWidth=41,title="Speed"
SetVariable setvar2,font="Arial",value= speed
Button buttonlast,pos={276,556},size={51,21},proc=framenexlas,title="<"
Button buttonnext,pos={330,556},size={51,21},proc=framenexlas,title=">"
CheckBox check13,pos={392,561},size={76,14},proc=CheckProc_2_setbackgr,title="Background"
CheckBox check13,value= 0
PopupMenu popup1,pos={481,554},size={21,21},bodyWidth=21,proc=fvalPopupMenuAction
PopupMenu popup1,mode=0,value= #"(fvalgetframes())"
TitleBox frame,pos={506,549},size={61,31},title=""
TitleBox frame,labelBack=(43520,43520,43520),font="Arial Black",fSize=24,frame=0
TitleBox frame,anchor= RC,fixSize=1

///ROI settings
TabControl tab0,pos={29,600},size={539,177},labelBack=(47872,47872,47872)
TabControl tab0,tabLabel(0)="Fit Cell",tabLabel(1)="Rectangle"
TabControl tab0,tabLabel(2)="Auto",tabLabel(3)="Manual",value= 0, proc=TabC
//Tab1
GroupBox group8,pos={72,635},size={194,129},title="scan area (box)"
GroupBox group8,labelBack=(26112,26112,26112)
GroupBox group9,pos={322,635},size={194,129},title="object recognition"
GroupBox group9,labelBack=(26112,26112,26112)
SetVariable setvar5,pos={99,684},size={119,16},bodyWidth=70,title="horizontal"
SetVariable setvar5,Format="%g pixels",value= sbsh, limits={0,inf,5}
SetVariable setvar6,pos={108,719},size={108,16},bodyWidth=70,title="vertical"

```

```

SetVariable setvar6,Format="%g pixels",value= sbsv, limits={0,inf,5}
SetVariable setvar3,pos={351,719},size={133,16},bodyWidth=70,title="min.cell size "
SetVariable setvar3,Format="%g pix^2",limits={0,100000,10},value= cellsize
SetVariable setvar4,pos={369,684},size={117,16},bodyWidth=70,title="threshold"
SetVariable setvar4,Format="%G AU",limits={0,4096,0.1},value= thr

//tab2
GroupBox group09,pos={72,635},size={194,129},title="rectangle size"
GroupBox group09,labelBack=(26112,26112,26112)
SetVariable setvar7,pos={99,684},size={119,16},bodyWidth=70,title="horizontal"
SetVariable setvar7,Format="%g pixels",limits={0,inf,10},value= sbsh
SetVariable setvar8,pos={108,719},size={108,16},bodyWidth=70,title="vertical"
SetVariable setvar8,Format="%g pixels",limits={0,inf,10},value= sbsv
CheckBox check44,pos={344,684},size={117,14},title="Soma and processes",value= 0

//tab3
Button button13,pos={44,632},size={80,50},title="Auto\rRecognition",proc=fW_autoROI
Button button15,pos={463,636},size={80,50},proc=defPNT,title="Define\rPoints"
Button button14,pos={463,705},size={80,50},proc=showarea,title="Area"
Button button16,pos={44,702},size={80,50},proc=importROIs,title="Import ROIs"

//tab4
GroupBox group14,pos={72,635},size={194,129},title="Thresholding"
GroupBox group14,labelBack=(26112,26112,26112)
GroupBox group15,pos={322,635},size={194,129},title="Auto object recognition"
GroupBox group15,labelBack=(26112,26112,26112)
SetVariable setvar06,pos={99,664},size={119,16},bodyWidth=70,title="horizontal"
SetVariable setvar06,Format="%g pixels",limits={0,500,5},value= sbsh
SetVariable setvar07,pos={108,699},size={108,16},bodyWidth=70,title="vertical"
SetVariable setvar07,Format="%g pixels",limits={0,500,5},value= sbsv
SetVariable setvar05,pos={99,734},size={117,16},bodyWidth=70,title="threshold"
SetVariable setvar05,Format="%G AU",limits={0,4096,0.1},value= thr
SetVariable setvar08,pos={348,699},size={136,16},bodyWidth=70,title="max.cell size "
SetVariable setvar08,Format="%g pix^2",limits={0,100000,10},value= Maxcellsize
SetVariable setvar04,pos={351,669},size={133,16},bodyWidth=70,title="min.cell size "
SetVariable setvar04,Format="%g pix^2",limits={0,100000,10},value=cellsize
Button button10,pos={369,733},size={100,20},proc=AutoRecognition,title="Start"

tabc("",0)

///application
Button button11,pos={1041,42},size={50,20},title="Analyze", proc=fval_analyse
CheckBox check12,pos={929,42},size={50,20}, proc=CheckProc_keep,title="keep "
GroupBox group4,pos={815,54},size={279,210},title="Application"
GroupBox group4,labelBack=(43520,43520,43520),fColor=(4352,4352,4352)
SetVariable setvarA1,pos={841,80},size={35,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA1,limits={-inf,inf,0},value= s1
SetVariable setvarA7,pos={885,80},size={35,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA7,limits={-inf,inf,0},value= e1
SetVariable setvarA13,pos={930,80},size={150,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA13,limits={-inf,inf,0},value= w1
SetVariable setvarA2,pos={841,110},size={35,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA2,limits={-inf,inf,0},value= s2
SetVariable setvarA8,pos={885,110},size={35,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA8,limits={-inf,inf,0},value= e2
SetVariable setvarA14,pos={930,110},size={150,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA14,limits={-inf,inf,0},value= w2
SetVariable setvarA3,pos={841,140},size={35,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA3,limits={-inf,inf,0},value= s3
SetVariable setvarA9,pos={885,140},size={35,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA9,limits={-inf,inf,0},value= e3
SetVariable setvarA15,pos={930,140},size={150,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA15,limits={-inf,inf,0},value= w3
SetVariable setvarA4,pos={841,170},size={35,16},proc=SetVarProc_Appli,title=" "

```

```

SetVariable setvarA4,limits={-inf,inf,0},value= s4
SetVariable setvarA10,pos={885,170},size={35,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA10,limits={-inf,inf,0},value= e4
SetVariable setvarA16,pos={930,170},size={150,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA16,limits={-inf,inf,0},value= w4
SetVariable setvarA5,pos={841,200},size={35,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA5,limits={-inf,inf,0},value= s5
SetVariable setvarA11,pos={885,200},size={35,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA11,limits={-inf,inf,0},value= e5
SetVariable setvarA17,pos={930,200},size={150,16},proc=SetVarProc_Appli,title=" "
SetVariable setvarA17,limits={-inf,inf,0},value= w5
GroupBox group16,pos={834,226},size={252,31},labelBack=(39168,0,0)
    SetVariable setvarA6,pos={841,233},size={35,16},proc=SetVarProc_Appli,title=" "
    SetVariable setvarA6,limits={-inf,inf,0},value= s6
    SetVariable setvarA12,pos={885,233},size={35,16},proc=SetVarProc_Appli,title=" "
    SetVariable setvarA12,limits={-inf,inf,0},value= e6
    SetVariable setvarA18,pos={930,233},size={150,16},proc=SetVarProc_Appli,title=" "
    SetVariable setvarA18,limits={-inf,inf,0},value= w6

//results
GroupBox group05,pos={1102,54},size={170,210},title="Results"
    GroupBox group05,labelBack=(43520,43520,43520),fColor=(4352,4352,4352)
    SetVariable setvarR1,pos={1116,80},size={78,16},proc=SetVarProc_Appli,title=" "
    SetVariable setvarR1,valueBackColor=(65280,65280,48896)
    SetVariable setvarR1,limits={-inf,inf,0},value= r1,noedit= 2
    SetVariable setvarR2,pos={1116,110},size={78,16},proc=SetVarProc_Appli,title=" "
    SetVariable setvarR2,valueBackColor=(65280,65280,48896)
    SetVariable setvarR2,limits={-inf,inf,0},value= r2,styledText= 1
    SetVariable setvarR3,pos={1116,140},size={78,16},proc=SetVarProc_Appli,title=" "
    SetVariable setvarR3,valueBackColor=(65280,65280,48896)
    SetVariable setvarR3,limits={-inf,inf,0},value= r3,styledText= 1
    SetVariable setvarR4,pos={1116,170},size={78,16},proc=SetVarProc_Appli,title=" "
    SetVariable setvarR4,valueBackColor=(65280,65280,48896)
    SetVariable setvarR4,limits={-inf,inf,0},value= r4,styledText= 1
    SetVariable setvarR5,pos={1116,200},size={78,16},proc=SetVarProc_Appli,title=" "
    SetVariable setvarR5,valueBackColor=(65280,65280,48896)
    SetVariable setvarR5,limits={-inf,inf,0},value= r5,styledText= 1
    GroupBox group17,pos={1108,226},size={159,30},labelBack=(39168,0,0)
        SetVariable setvarR6,pos={1116,233},size={78,16},proc=SetVarProc_Appli,title=" "
        SetVariable setvarR6,valueBackColor=(65280,65280,48896)
        SetVariable setvarR6,limits={-inf,inf,0},value= r6,styledText= 1
    PopupMenu popup4,pos={1191,41},size={78,21},bodyWidth=78,proc=fval_popResults
    PopupMenu popup4,fColor=(21760,21760,21760)
    PopupMenu popup4,mode=4,popvalue="Mean_X",value= #"\Mean;Max; Min;Mean_X; SDvalue\"
    SetVariable POPPerc1,pos={1209,80},size={50,16},title=" ",value= PerC0,limits={-inf,inf,0}
    SetVariable POPPerc2,pos={1209,110},size={50,16},title=" ",value= PerC1,limits={-inf,inf,0}
    SetVariable POPPerc3,pos={1209,140},size={50,16},title=" ",value= PerC2,limits={-inf,inf,0}
    SetVariable POPPerc4,pos={1209,170},size={50,16},title=" ",value= PerC3,limits={-inf,inf,0}
    SetVariable POPPerc5,pos={1209,200},size={50,16},title=" ",value= PerC4,limits={-inf,inf,0}
    SetVariable POPPerc6,pos={1209,233},size={50,16},title=" ",value= PerC5,limits={-inf,inf,0}

TabControl tab1,pos={815,600},size={459,177},proc=TabC1
    TabControl tab1,labelBack=(47872,47872,47872),tabLabel(0)="Online Parameter"
    TabControl tab1,tabLabel(1)="Display",value= 1
    ///Tab1
    GroupBox group10,pos={833,625},size={280,140},title="Channels"
    SetVariable setvar11,pos={854,690},size={79,16},bodyWidth=30,disable=1,proc=setChanNum
    SetVariable setvar11,limits={1,4,1},value=numChan, title="channels"
    SetVariable setvar9,pos={950,647},size={150,16},bodyWidth=150,title=" "
    SetVariable setvar9,limits={-inf,inf,0},styledText= 1, value=chan1
    SetVariable setvar10,pos={950,677},size={150,16},bodyWidth=150,title=" "
    SetVariable setvar10,limits={-inf,inf,0},styledText= 1, value=chan2

```

```

SetVariable setvar12,pos={950,707},size={150,16},bodyWidth=150,title=" "
SetVariable setvar12,limits={-inf,inf,0},styledText= 1, value=chan3
SetVariable setvar13,pos={950,737},size={150,16},bodyWidth=150,title=" "
SetVariable setvar13,limits={-inf,inf,0},styledText= 1, value=chan4

GroupBox group13,pos={1125,625},size={135,140},disable=1,title="scaling"
SetVariable setvar14,pos={1137,647},size={115,16},bodyWidth=90,disable=1,title="freq."
SetVariable setvar14,Format="%g frames/s",limits={0,inf,0.1},value= smpl
CheckBox check02,pos={1154,676},size={78,14},title="background "
CheckBox check02,value= 0,side= 1,proc=CheckProc_back
CheckBox check03,pos={1154,706},size={78,14}, title="norm bef. sub", proc=CheckProc_1
Button button06,pos={1155,733},size={80,20},proc=updateTrace,title="update"

```

```

///Tab2

```

```

GroupBox group11,pos={831,632},size={419,58},title="Graph1"
CheckBox check1,pos={843,659},size={21,14},title=" ",value= 0,proc=DisableGraph
PopupMenu popup2,pos={876,657},size={49,21},proc=setdisp
PopupMenu popup2,mode=2,popvalue="ratio",value= #""\single;ratio\""
CheckBox check3,pos={946,659},size={51,14},title="chan 1",value= 0,proc=CheckProc
CheckBox check4,pos={1006,659},size={51,14},title="chan 2",value= 0,proc=CheckProc
CheckBox check5,pos={1066,659},size={51,14},title="chan 3",value= 0,proc=CheckProc
CheckBox check6,pos={1126,659},size={51,14},title="chan 4",value= 0,proc=CheckProc

PopupMenu popup1_1,pos={947,659},size={100,21},bodyWidth=100,proc=PopupMenuProc
PopupMenu popup1_1,mode=1,value=getNameChan()
PopupMenu popup1_2,pos={1077,659},size={100,21},bodyWidth=100,proc=PopupMenuProc
PopupMenu popup1_2,mode=2,value=getNameChan()
TitleBox title0,pos={1057,650},size={15,35},title="/",font="Arial",fSize=24
CheckBox check10,pos={1193,660},size={44,14},title="norm.",value= 1,proc=NormProc

```

```

GroupBox group12,pos={831,702},size={419,58},title="Graph2"
CheckBox check2,pos={845,732},size={21,14},title=" ",value= 0,proc=DisableGraph
PopupMenu popup3,pos={876,728},size={56,21},proc=setdisp, disable=1
PopupMenu popup3,mode=1,popvalue="single",value= #""\single;ratio\""
CheckBox check7,pos={946,729},size={51,14},title="chan 1",value= 0
CheckBox check8,pos={1006,729},size={51,14},title="chan 2",value= 0
CheckBox check9,pos={1066,729},size={51,14},title="chan 3",value= 0
CheckBox check07,pos={1126,729},size={51,14},title="chan 4",value= 0
PopupMenu popup1_3,pos={935,727},size={100,21},bodyWidth=120
PopupMenu popup1_3,mode=1,value=getNameChan()
PopupMenu popup1_4,pos={1077,727},size={100,21},bodyWidth=120
PopupMenu popup1_4,mode=2,value=getNameChan()
TitleBox title1,pos={1047,720},size={15,35},title="/",font="Arial",fSize=24
CheckBox check11,pos={1193,730},size={44,14},title="norm.",value= 1

```

```

TabC1("",0)
setChanNum("setvar11",1,"1", "numChan")

```

```

Display/W=(29,69,570,506)/HOST=#
MAKE/O/N=(2,2) curlIMG
AppEndImage/T curlIMG
ModfilyImage curlIMG ctab= {*,*,Grays,0}
ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

```

```

        ModifyGraph mirror=0
        ModifyGraph nticks=0
        ModifyGraph noLabel=2
        ModifyGraph standoff=0
        ModifyGraph axThick=0
        SetAxis/A/R left
        RenameWindow #,G0
        SetActiveSubwindow ##
SetWindow fval hook=getR, hookevents=1

Display/W=(815,278,1272,591)/HOST=#/L=IA
MAKE/O/N=(1) traces=nan, traces_X=nan
AppEndToGraph traces vs traces_X
//AppEndToGraph curTrace
//AppEndToGraph/L=IA appli vs appli_x
//ModifyGraph marker(traces)=19
//ModifyGraph lSize(appli)=2
//ModifyGraph rgb(traces)=(26112,52224,0),rgb(appli)=(0,0,0)
//ModifyGraph noLabel(IA)=2
//ModifyGraph axThick(IA)=0
ModifyGraph lblPos(left)=56
//ModifyGraph freePos(IA)=-50
ModifyGraph axisEnab(left)={0,0.8}
//ModifyGraph axisEnab(IA)={0.9,1}

RenameWindow #,G1
SetActiveSubwindow ##

//      SelPICT("",0,stringfromlist(0,GetpictList(),";"))
End //F_val()

Function loadsinglechan(str1)
string str1//pathtofile
NVAR f1=root:f1
NVAR f2=root:f2
NVAR redfac
NVAR numchan
NVAR whichchan
        f1=(floor(f1/numchan))*numchan+whichchan

wave curFSP0
killwaves/Z curca
        make/O/W/U/N=(dimsize(curFSP0,0)/redfac,dimsize(curFSP0,1)/redfac,ceil((f2-f1)/numchan)) curca=0

wave curca
        If(!waveexists(curca))//The "out of memory" problem...
                print dimsize(curFSP0,0)/redfac,dimsize(curFSP0,1)/redfac,ceil((f2-f1)/numchan)
                return 0
        EndIf
variable p1,q1

        For(p1=f1;p1<f2;p1+=numchan)
                ImageLoad/N=test/O/Q/LR3D/S=(p1)/C=(1) str1
wave test=${stringfromlist(0,S_waveNames,";")}
                If(redfac>1)//reduce size

                        ImageInterpolate/PXSZ={redfac,redfac} Pixelate      test

wave M_PixelatedImage
                        imagetransForm/P=(q1)/D=M_PixelatedImage setplane curca
                else
                        imagetransForm/P=(q1)/D=test setplane curca

```



```

        EndIf
        killwaves/Z test
        q1+=1
        slider/Z slider0, win=FSP, value=p1
    EndFor
End

Function TabC(ctrlName,tabNum) : TabControl
    String ctrlName
    Variable tabNum
    //monitor("TabC("+ctrlName+", "+num2str(tabNum)+")")

    GroupBox group8,win=fval,disable=(tabnum!=0)
    GroupBox group9,win=fval,disable=(tabnum!=0)
    SetVariable setvar5,win=fval,disable=(tabnum!=0)
    SetVariable setvar6,win=fval,disable=(tabnum!=0)
    SetVariable setvar3,win=fval,disable=(tabnum!=0)
    SetVariable setvar4,win=fval,disable=(tabnum!=0)

    GroupBox group09,win=fval,disable=(tabnum!=1)
    SetVariable setvar7,win=fval,disable=(tabnum!=1)
    SetVariable setvar8,win=fval,disable=(tabnum!=1)
    CheckBox check44,win=fval,disable=(tabnum!=1)

    Button button13,win=fval,disable=(tabnum!=2)
    Button button14,win=fval,disable=(tabnum!=2)
    Button button15,win=fval,disable=(tabnum!=2)
    Button button16,win=fval,disable=(tabnum!=2)

    GroupBox group14,win=fval,disable=(tabnum!=3)
    GroupBox group15,win=fval,disable=(tabnum!=3)
    SetVariable setvar06,win=fval,disable=(tabnum!=3)
    SetVariable setvar07,win=fval,disable=(tabnum!=3)
    SetVariable setvar05,win=fval,disable=(tabnum!=3)
    SetVariable setvar08,win=fval,disable=(tabnum!=3)
    SetVariable setvar04,win=fval,disable=(tabnum!=3)
    Button button10,win=fval,disable=(tabnum!=3)

    return 0
End

Function fval_popFILE(ctrlName) : ButtonControl
    String ctrlName
    monitor("fval_popFILE("+ctrlName+")")

    popupContextualMenu/C=(31, 21) "Load;Close all Movies;Save all Data"

    strswitch(S_selection)
        case "Load":
            ButtonProc("") ///FILELOADER.ips
            break
        case "Close all Movies":
            CloseMovies()
            break
        case "Save all Data":
            implementDATA("")
            break
    Endswitch
    checkbox check0, win=fval, value=0
End //fval_popFILE

Function CloseMovies()
    monitor("CloseMovies()")

string wl=GETpictlist()

```

```

variable p1
  For (p1=0; p1<itemsinlist(wl); p1+=1)
    killwaves/Z root:$(stringfromlist(p1, wl, ";"))
  EndFor
End

Function /S GETpictlist()
  monitor("GETpictlist()")
  string expo=removefromlist("curlMG",wavelist(";",",", "DIMs:3"),";")
  expo=removefromlist("listW",expo,";")

  return expo
End

Function fvalSliderProc(ctrlName,sliderValue,event) : SliderControl
  String ctrlName
  Variable sliderValue
  Variable event // bit field: bit 0: value set, 1: mouse down, 2: mouse up, 3: mouse moved
  monitor("fvalSliderProc"+"ctrlName+", "+num2str(sliderValue)+", "+num2str(event)+")"
  fvaldisplayFrame(sliderValue)

  return 0
End

Function/S fvalgetframes()
  monitor("fvalgetframes()")
  string expo=""
  SVAR picn
  wave curca=:(picN)
  variable numFrames=dimsize(curca,2)-1

  variable p1
  For(p1=0;p1<numFrames;p1+=1)
    expo+="Frame_" +num2str(p1)+";"
  EndFor

  return expo
End

Function PICnexlas(Diana) : ButtonControl
  String Diana
  monitor("PICnexlas"+"Diana +")"
  string PICList=getpictlist()
  SVAR picN
  variable curlTEM= whichlistitem(picn,getpictlist(),";")
  If(curlTEM<0)
    SelPICT("",0,stringfromlist(0,piclist,";"))
    return 0
  EndIf

  If(stringmatch(Diana,"*0")) ///Forward
    curitem+=1
    If(curlitem>itemsinlist(piclist,";")-1)
      curitem=0
    EndIf
  EndIf

  If(stringmatch(Diana,"*1")) ///back
    curitem-=1
    If(curlitem<0)
      curitem=itemsinlist(piclist,";")-1
    EndIf
  EndIf
End

```

```

        SelPICT("",curlItem,stringfromlist(curlITEM,piclist,";"))
        //SelPICT("",popNum,"")
End

Function  setAPPLiRes()
    monitor("setAPPLiRes()")
SVAR pn=root:picN
//      print datafolderexists(pn),datafolderdir(1)

variable p1=0
    For(p1=0;p1<6;p1+=1)

        SetVariable $("setvarA"+num2str(1+p1)), win=fval, value=root:$(pn):$("s"+num2str(1+p1))
        SetVariable $("setvarA"+num2str(7+p1)), win=fval, value=root:$(pn):$("e"+num2str(1+p1))
        SetVariable $("setvarA"+num2str(13+p1)), win=fval, value=root:$(pn):$("w"+num2str(1+p1))
        SetVariable $("setvarR"+num2str(1+p1)), win=fval, value=root:$(pn):$("r"+num2str(1+p1))
        SetVariable $("POPPerc"+num2str(1+p1)), win=fval, value=root:$(pn):$("Perc"+num2str(0+p1))

    EndFor
End

Function framenexas(lasch):ButtonControl
string lasch
    monitor("framenexas("+lasch+")")
NVAR numchan
wave CurIMG
variable CurFrameInMovie=str2num(note(curlmg))
variable change
    If(stringmatch(lasch, "*last"))
        change=-numchan
    else
        change=numchan
    EndIf
    fvalDisplayframe(CurFrameInMovie+change)
End

Function fvalPlayMovie(Vicky) : ButtonControl
    String Vicky
    monitor("fvalPlayMovie("+Vicky+")")

NVAR speed //in frames/sec
    If(speed==0)
        return 0
    EndIf
wave curlIMG
NVAR numchan
variable change=numchan///maybe 2 For ratiometric purposes
do
variable t0=ticks ///1 tick= 1/60s
do
    while(ticks<t0+(60/speed))
variable CurFrameInMovie=str2num(note(curlmg))
        fvalDisplayframe(CurFrameInMovie+change)
        doupdate/W=fval#0

        If(GetKeyState(0)==32)//Esc
            break
        EndIf
    while(1)
End

Function fvalupdateInfo(var)
variable var
    monitor("fvalupdateInfo("+num2str(var)+")")
SVAR picN
NVAR numchan

```

```

wave curca=${picN}
    slider slider0, win=fval,limits={0,dimsz(curca,2),1}, value=var
    TitleBox frame, win=fval, title=num2str(var)
        //var/=chan
    If(strlen(wavelist("*","","WIN:fval#g1")))
        setdrawlayer/K/W=fval#g1 userfront
        setdrawenv/W=fval#g1 linethick=2,linefgc=(0,0,0),xcoord=bottom, ycoord=prel,save
        drawline/W=fval#g1 var/numchan,0,var/numchan,1
    EndIf

    //checkgROI()
End

```

```

Function fvalDisplayframe(Clara)
variable Clara
    monitor("fvalDisplayframe("+num2str(Clara)+")")
    Clara =round(Clara)
    ///never zMax...
    checkbox check0, win=fval, value=0

```

```

SVAR picN
wave CurCa=root:${picN}
wave curlmg
variable numF=dimsz(CurCa,2)-1
    If(Clara <0)
        Clara =numF
    EndIf
    If(Clara >numF)
        Clara =0
    EndIf

    Imagetranform/P=(Clara) getplane curca
wave M_ImagePlane
    duplicate/O M_ImagePlane, curlmg
wave curimg
    note/K curlmg, num2str(Clara)
    killwaves/Z M_ImagePlane

    controlinfo/W=fval check13
    If(V_value)//checked
wave bas
        redimension/S curlMG
        curimg-=bas
        curimg*=(curimg>0)
    EndIf
NVAR If1=root:imgFilter
    If(If1>1)
variable p1
        For(p1=0;p1<If1;p1+=1)
            imagefilter/N=(3) gauss curlMG
        EndFor
    EndIf

    fvalupdateInfo(Clara)
End

```

```

///Contrast
Function SetPICT(ctrlName,varNum,varStr,varName) : SetVariableControl

    String ctrlName
    Variable varNum
    String varStr
    String varName
    monitor("SetPICT("+ctrlName+", "+num2str(varnum)+", "+varStr+", "+varName+")")

```

```
SVAR picN=root:picn
```

```

NVAR zmin=root:$(picN):zmin,zmax=root:$(picN):zmax
NVAR smin=root:zmin,smax=root:zmax
    smin=zmin
    smax=zmax
    ModifyImage/W=fval#g0 curlMG ctab= {zmin,zmax,Grays,0}

///store variable
SVAR picN=root:picN
    If(datafolderexists(picN))
        EndIf
End

//Display
Function ROIleg(ctrlName) : ButtonControl
    String ctrlName
    monitor("ROIleg("+ctrlName+")")
    setdrawlayer/K/W=fval#g0 progfront
    setdrawlayer/K/W=fval#g0 userfront
    If(stringmatch(ctrlName,"*5"))
        return 0
    EndIf
    controlinfo/W=fval check01
variable checked=v_value
    If(checked)
        drawallbROIs()
    EndIf
    drawallROIs()

    If(stringmatch(ctrlName,"*4"))
        return 0
    EndIf

    If(checked)
        drawallbLEG()
    EndIf
    drawallLEG()
End

Function drawallLEG()
    monitor("drawallLEG()")
SVAR picN
wave/T wC=root:$(picN):$("roiCOORD")
variable p1

    For (p1=0; p1<numpts(wC);p1+=1)
        makeCoordWfromStr(p1)
wave xw,yw
        setdrawlayer/W=fval#g0 progfront
        setdrawenv/W=fval#g0 xcoord=top,ycoord=left,txtrgb=(65000,65000,0), fsize=10, fstyle=1,save
        wavestats/Q xw; variable meanX=V_avg
        wavestats/Q yw; variable meanY=V_avg
        drawtext/W=fval#g0 meanX,meanY, "ROI_" + num2str(p1+1)//10,10,"ROI_" + num2str(p1+1)//
    EndFor
End

Function drawallbLEG()
    monitor("drawallbLEG()")
SVAR picN
wave/T wC=root:$(picN):$("roibCOORD")
variable p1

    For (p1=0; p1<numpts(wC);p1+=1)
        makebCoordWfromStr(p1)
wave xw,yw
        setdrawlayer/W=fval#g0 userfront
        setdrawenv/W=fval#g0 xcoord=top,ycoord=left,txtrgb=(65000,65000,65000), fsize=10, fstyle=.5,save

```

```

        wavestats/Q xw; variable meanX=V_avg
        wavestats/Q yw; variable meanY=V_avg
        drawtext/W=fval#g0 meanX,meanY, "b_"+num2str(p1+1)///10,10,"ROI_"+num2str(p1+1)//
    EndFor
End

Function drawallrois()
    monitor("drawallrois()")
SVAR picN
wave/T wC=root:$(picN):$("roiCOORD")
variable p1

make/O/N=0 xA,yA
    For (p1=0; p1<numpts(wC);p1+=1)

        makeCoordWfromStr(p1)
wave xw,yw
        concatenate/NP {xw},xA
        concatenate/NP {yw},yA

        insertpoints (numpts(xA)),1, xA,yA
        xA[numpts(xA)-1]=NaN
        yA[numpts(xA)-1]=NaN
    EndFor
    setdrawlayer/W=fval#g0/K progfront
    setdrawenv/W=fval#g0 xcoord=top,ycoord=left,linefgc=(65000,65000,0), linethick=1, fillpat=0,save
    drawpoly/W=fval#g0/ABS 0,0,1,1,xA,yA

    controlinfo/W=fval check01
End

Function drawallbrois()
    monitor("drawallbrois()")
SVAR picN
wave/T wC=root:$(picN):$("roibCOORD")
variable p1

make/O/N=0 xB,yB
    For (p1=0; p1<numpts(wC);p1+=1)
        makebCoordWfromStr(p1)
wave xw,yw
        concatenate/NP {xw},xB
        concatenate/NP {yw},yB

        insertpoints (numpts(xB)),1, xB,yB
        xB[numpts(xB)-1]=NaN
        yB[numpts(xB)-1]=NaN
    EndFor
    setdrawlayer/W=fval#g0/K userfront
    setdrawenv/W=fval#g0 xcoord=top,ycoord=left,linefgc=(65000,65000,65000), linethick=.5, fillpat=0,save
    drawpoly/W=fval#g0/ABS 0,0,1,1,xB,yB
End

Function makeCoordWfromStr(num)
variable num
    monitor("makeCoordWfromStr("+num2str(num)+")")
    make/O/N=0 root:xW,root:yW
wave xw=root:xw,yw=root:yw
SVAR picN
wave/T wC=root:$(picN):$("roiCOORD")
string curC=wc[num]

variable p1
    For(p1=0; p1<itemsinlist(curC,";"); p1+=1)
string curCoord=stringfromlist(p1,curC,";")
        insertpoints p1, 1, xw,yw

```

```

                xw[p1]=str2num(stringfromlist(0,curcoord,";"))
                yw[p1]= str2num(stringfromlist(1,curcoord,";"))
            EndFor
End

Function makebCoordWfromStr(num)
variable num

    monitor("makebCoordWfromStr("+num2str(num)+")")
    make/O/N=0 root:xW,root:yW
wave xw=root:xw,yw=root:yw
SVAR picN
wave/T wC=root:$(picN):$("roibCOOR")
string curC=wc[num]

variable p1
    For(p1=0; p1<itemsinlist(curC,";"); p1+=1)
string curCoord=stringfromlist(p1,curC,";")
        insertpoints p1, 1, xw,yw
        xw[p1]=str2num(stringfromlist(0,curcoord,";"))
        yw[p1]= str2num(stringfromlist(1,curcoord,";"))
    EndFor
End

///ROI action
Function Delete_allROIs(ctrlName) : ButtonControl
    String ctrlName
    monitor("Delete_allROIs("+ctrlName+")")
    make/O/N=0 traces,traces_X, curTrace
wave/T listW
    listW=""
SVAR picN

    Make/O/N=0/T root:$(picN):roiMean
    Make/O/N=0/T root:$(picN):roiCOOR
    Make/O/N=0/T root:$(picN):roiNum
    Make/O/N=0/T root:$(picN):roibMean
    Make/O/N=0/T root:$(picN):roibCOOR
    Make/O/N=0 root:$(picN):roiArea
    Make/O/N=0/T root:$(picN):roiLoc
    SetDrawLayer/K/W=fval#g0 progfront
    SetDrawLayer/K/W=fval#g0 userfront
    removefromgraph/Z/W=fval#g0 yw,xw
End

Function deleteROI(ctrlName) : ButtonControl
    String ctrlName
    //monitor("deleteROI("+ctrlName+")")
    controlinfo/W=fval list0
variable num=V_value ///curent item in list; 2b deleted!

SVAR picN
wave/T wn=root:$(picN):$("roiNUM")
wave/T wC=root:$(picN):$("roiCOOR")
wave wM=root:$(picN):$("roiMEAN")
wave/T wbC=root:$(picN):$("roibCOOR")
wave wbM=root:$(picN):$("roibMEAN")
wave wA=root:$(picN):$("roiArea")
wave/T wL=root:$(picN):$("roiLoc")
wave curCa=root:$(picN)
NVAR chan=:numchan
variable numP=(dimsize(curca,2)-1)/chan
wave traces, traces_x
    deletepoints num*numP,numP+2, traces, traces_x
wave curtrace
    curtrace=nan

```

```

        deletepoints num,1, wn,wc,wm,wbC,wbm
        If(waveexists(wa))
            deletepoints num,1, wa
        EndIf
        If(waveexists(wl))
            deletepoints num,1, wl
        EndIf
        updateListW()
        drawallROIS()

        setdrawlayer/W=fval#g0/K userfront
End

//Listbox
Function displayCurTrace(row)
variable row
    monitor("displayCurTrace("+num2str(row)+"")

        getcurtrace(row)
string wl=wavelist("*",";","WIN:fval#g1")
    If(!stringmatch(wl,"*curtrace*"))
wave curtrace
        If(waveexists(curtrace))
            appEndtograph/W=fval#g1 curtrace
            modifygraph/W=fval#g1      rgb(curtrace)=(0,0,0), lsize(curtrace)=4
            resumeupdate
            ReorderTraces curtrace,{}
        EndIf
    EndIf
End

Function getframe(var)
variable var
    monitor("displayCurTrace("+num2str(var)+"")
variable expo=round(AxisValFromPixel("fval#g1", "bottom", var ))
SVAR picN
wave curW=root:$(picN)
    If(expo<0)
        expo=0
    EndIf
    If(expo>dimsize(curw,2))
        expo=dimsize(curw,2)-1
    EndIf

    return expo
End

Function DisplayEverything()
    monitor("displayEverything()")

    removefromgraph/W=fval#g0/Z yw,xw
    drawallrois()
    controlinfo/W=fval check01
    If(v_value)
        drawallbrois()
    EndIf
    MakeTraces(0)
End

Function MakeTraces(var)
variable var //either 1(calc all traces) or 0 (only appEnd curtrace)
    monitor("MakeTraces("+num2str(var)+"")

    If(var)///make all traces from beginning
        recalc_AllTraces()
    else ///only appEnd curtrace

```



```

        controlinfo/W=fval list0
        getcurTrace(v_value)
        appEndCurTrace("curTrace")
    EndIf

    killwaves/Z W_isBeamAvg,W_isBeamMin, W_isBeamMax
End

Function DrawROI(num)
variable num

    monitor("DrawROI("+num2str(num)+")")
    makeCoordWfromStr(num)
wave xw=root:xw,yw=root:yw
    setdrawenv/W=fval#g0 xcoord=top,ycoord=left,linefgc=(65000,0,0), linethick=2, fillpat=0,save
    setdrawlayer/W=fval#g0 userfront
    drawpoly/W=fval#g0/ABS 0,0,1,1,xw,yw
End

Function normT()
    monitor("normT()")
    controlinfo/W=fval check10
    If(v_value)
wave curtrace=:CurTrace
variable norm1

NVAR nc=NumChan
variable p1=0
        do
            norm1=curtrace[p1]
            If(!stringmatch(num2str(norm1),"NaN"))
                break
            EndIf
            p1+=1
        while(p1<nc)

        curtrace/=norm1
    EndIf
End

Function timer()
    monitor("timer()")
wave curTrace=:curtrace
wave traces_X
NVAR s1=:smpl

variable cf=numchan1()

variable smpl1=s1*cf
    If(smpl1)
        setscale/P x, 0,(1/smpl1), "s", curTrace
    EndIf
End

Function numchan1()
NVAR nc=:NumChan
    controlinfo/W=fval popup2
    If(!(v_value-1))//single
        return nc
    EndIf
    return 1
End

Function swapChan()

```

```

        monitor("swapChan()")

NVAR nc=:numchan

        If (nc-1)//more channels,i.e. ratio or 2 graphs possible
        ///at the moment, only 1 graph

                controlinfo/W=fval popup2
                If(v_value-1) ///ratio
                        controlinfo/W=fval popup1_1
variable r1=v_value-1
                        controlinfo/W=fval popup1_2
variable r2=v_value-1
                RatiometricCUR(r1,r2,nc)
                else ///single
                        decreaseCUR(nc)
                EndIf
        EndIf

        LabelY(v_value)
End

```

```

Function LabelY(var)
variable var
        monitor("LabelY("+num2str(var)+")")

string l1=""

```

```

        controlinfo/W=fval popup2
        If(v_value-1) ///ratio
                controlinfo/W=fval popup1_1
                l1+=S_value+ " / "
                controlinfo/W=fval popup1_2
                l1+=S_value
        else ///single
                controlinfo/W=fval check3
                If(v_value)

                        l1+=c1+" "

                EndIf
                controlinfo/W=fval check4
                If(v_value)

                        l1+=c2+" "

                EndIf
                controlinfo/W=fval check5
                If(v_value)

                        l1+=c3+" "

                EndIf
                controlinfo/W=fval check6
                If(v_value)

                        l1+=c4+" "

                EndIf
        EndIf

        Label/W=fval#G1 left, l1

End

```

```

Function decreaseCUR(nc)

```

```

variable nc
    monitor("decreaseCUR("+num2str(nc)+")")

///one wavelength
///only display
wave curtrace=:curtrace ///all chans in a row

    controlinfo/W=fval check3
string c1=num2str(v_value)+";"
    controlinfo/W=fval check4
    c1+=num2str(v_value)+";"
    controlinfo/W=fval check5
    c1+=num2str(v_value)+";"
    controlinfo/W=fval check6
    c1+=num2str(v_value)+";"

    duplicate/O curtrace,f1
wave f1
variable p1, q1

    For(p1=0; p1<numpts(f1); p1+=nc)///
        For(q1=0;q1<nc;q1+=1)
            If(!str2num(stringfromlist(q1,c1,";"))))
                F1[p1+q1]=nan
            EndIf
        EndFor
    EndFor
    duplicate/O f1,curtrace
    killwaves/z f1

End

Function RatiometricCUR(chan1,chan2,nc)
variable chan1
variable chan2
variable nc
    monitor("RatiometricCUR("+num2str(chan1)+num2str(chan2)+num2str(nc)+")")
///one wavelength
///only display
wave traces=:curtrace ///all chans in a row

variable p1
    make/O/N=0 F1,F2
    For(p1=0; p1<numpts(traces); p1+=nc)///
        insertpoints (numpts(F1)), 1, F1,F2
        F1[numpts(F1)-1]=traces[p1+chan1]
        F2[numpts(F1)-1]=traces[p1+chan2]
    EndFor
    duplicate/O F1, curtrace
    curtrace/=F2

    killwaves/Z f1,f2

End

Function setChanNum(ctrlName,varNum,varStr,varName) : SetVariableControl
    String ctrlName
    Variable varNum
    String varStr
    String varName
    monitor("setChanNum("+ctrlName+", "+num2str(varNum)+", "+varStr+", "+varName+")")
string svs= "setvar9;setvar10;setvar12;setvar13"
variable p1
    For(p1=0; p1<itemsinlist(svs,";");p1+=1)
        SetVariable $(stringfromlist(p1,svs,";")),win=fval, noedit=0, valueBackColor=0,valueColor=(0,0,0)
    EndFor
EndFunction

```

```

                If(varnum<p1+1)
                    SetVariable
noedit=1,valueBackColor=(47872,47872,47872),valueColor=(47872,47872,47872)
                EndIf
            EndFor
            Slider slider0 win=fval,limits={0,860,varnum}
End

Function fvalPopupMenuAction (ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum    // which item is currently selected (1-based)
    String popStr      // contents of current popup item as string
//feedback
    monitor("fvalPopupMenuAction("+ctrlName+", "+num2str(popNum)+", "+popStr+")")
    fvaldisplayframe(popNum-1)
End

Function CheckProc(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked
    monitor("CheckProc("+ctrlName+", "+num2str(checked)+")")
    maketraces(1)
End

Function PopMenuProc(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum
    String popStr
    monitor("PopMenuProc("+ctrlName+", "+num2str(popnum)+", "+popStr+")")
    maketraces(1)
End

Function SetVarProc_Appli(ctrlName,varNum,varStr,varName) : SetVariableControl
    String ctrlName
    Variable varNum
    String varStr
    String varName
    monitor("SetVarProc_Appli("+ctrlName+", "+num2str(varNum)+", "+varStr+", "+varName+")")

    make/O/N=18 appli=nan, appli_x=nan
SVAR pn=root:picN
variable p=0, isit=0
do
SVAR curS=root:${pn}:${"s"+num2str(p/3+1)}
SVAR curE=root:${pn}:${"e"+num2str(p/3+1)}
    If( (str2num(curS)) &&str2num((curE)))
        isit=1
    EndIf
    //print curS,"s"+num2str(p/3+1)
    appli[p]=p
    appli[p+1]=p

    appli_x[p]=str2num(curS)
    appli_x[p+1]=str2num(curE)
    appli_x[p+2]=NAN

    p+=3
    while(p<18)

    If( isit)
//        print "YES"
        Apply2Graph()
    EndIf
    return 0
End

Function Apply2Graph()

```

```

//print strlen(listmatch(wavelist("**",",","WIN:Graph0#G0"), "appli"))
//application bars
    monitor ("Apply2Graph()")
wave appli, appli_X
    If(strlen(listmatch(wavelist("**",",","WIN:fval#g1"), "appli")))
        If(waveExists(appli))
            AppEndToGraph/W=fval#g1/L=IA appli vs appli_x
            ModifyGraph/W=fval#g1 noLabel(IA)=2,axThick(IA)=0,axisEnab(left)={0,0.8};DelayUpdate
            ModifyGraph/W=fval#g1 axisEnab(IA)={0.9,1}, rgb(appli)=(0,0,0),lsize(appli)=2
            ModifyGraph/W=fval#g1 axisEnab(left)={0,0.8}
            setdrawlayer/K/W=fval#g1 userback
            setdrawenv/W=fval#g1 fsize=8, xcoord=bottom, ycoord=IA, save
        EndIf
    else
        setdrawlayer/K/W=fval#g1 userback
        setdrawenv/W=fval#g1 fsize=8, xcoord=bottom, ycoord=IA, save
    EndIf
//text above app bars
Variable p=0
do
SVAR picN=root:picN
SVAR appl=root:$(picN):$("w"+num2str(p+1))
SVAR sA= root:$(picN):$("s"+num2str(p+1))

        drawtext/W=fval#g1 str2num(sA), p*3, appl
        p+=1
        while(p<6)
//mark peakvals
//        peakValAppl()

End

```

```

Function TabC1(ctrlName,tabNum) : TabControl
String ctrlName
Variable tabNum
monitor("TabC1{"+ctrlName+"", "+num2str(tabNum)+"}")
tabcontrol tab1,win=fval, value=tabnum ///feedback

GroupBox group10,disable=(tabnum!=0),win=fval
SetVariable setvar9,disable=(tabnum!=0),win=fval
SetVariable setvar11,disable=(tabnum!=0),win=fval
SetVariable setvar10,disable=(tabnum!=0),win=fval
SetVariable setvar12,disable=(tabnum!=0),win=fval
SetVariable setvar13,disable=(tabnum!=0),win=fval
GroupBox group13 disable=(tabnum!=0),win=fval
SetVariable setvar14 disable=(tabnum!=0),win=fval
CheckBox check02,disable=(tabnum!=0),win=fval
CheckBox check03,disable=(tabnum!=0),win=fval
Button button06,disable=(tabnum!=0),win=fval

GroupBox group11,disable=(tabnum!=1),win=fval
CheckBox check1,disable=(tabnum!=1),win=fval
PopupMenu popup2,disable=1,win=fval
CheckBox check3,disable=1,win=fval
CheckBox check4,disable=1,win=fval
CheckBox check5,disable=1,win=fval
CheckBox check6,disable=1,win=fval
PopupMenu popup1_1,disable=1,win=fval
PopupMenu popup1_2,disable=1,win=fval
TitleBox title0,disable=1,win=fval
CheckBox check10,disable=(tabnum!=1),win=fval

GroupBox group12,disable=(tabnum!=1),win=fval
CheckBox check2,disable=(tabnum!=1),win=fval
PopupMenu popup3,disable=1,win=fval
CheckBox check7,disable=1,win=fval

```

```

CheckBox check8,disable=1,win=fval
CheckBox check9,disable=1,win=fval
CheckBox check07,disable=1,win=fval
PopupMenu popup1_3,disable=1,win=fval
PopupMenu popup1_4,disable=1,win=fval
TitleBox title1,disable=1,win=fval
CheckBox check11,disable=(tabnum!=1),win=fval

If(tabnum==1)
    controlinfo/W=fval check1
    DisableGraph("check1",v_value)
    controlinfo/W=fval check2
    DisableGraph("check2",v_value)
EndIf
End

Function CheckProc_1(ctrlName,checked) : CheckBoxControl
String ctrlName
Variable checked

If(checked)
    checkbox check10, win=fval,value=0
EndIf
End

Function setdisp(ctrlName,popNum,popStr) : PopupMenuControl
String ctrlName
Variable popNum
String popStr
monitor(" setdisp("+ctrlName+" ,"+num2str(popNum)+", "+popStr+" ) ")
If(stringmatch(ctrlName, "*2"))///graph1
    controlinfo/W=fval check1
    DisableGraph("check1",v_value)
    MakeTraces(1)

elseif(stringmatch(ctrlName, "*3"))///graph2
    controlinfo/W=fval check2
    DisableGraph("check2",v_value)
EndIf
End

Function/S getNameChan()
monitor("getNameChan()")
string expo=""
string normal=""
NVAR numchan
variable p1
    For (p1=0; p1<numchan;p1+=1)
        normal="channel "+num2str(p1+1)
SVAR curS=root:$("chan"+num2str(p1+1))
        If(strlen(curS))
            normal=curS
        EndIf
        expo+=normal+";"
    EndFor
return expo
End

Function DisableGraph(ctrlName,checked) : CheckBoxControl
String ctrlName
Variable checked
monitor(" DisableGraph("+ctrlName+" ,"+num2str(checked)+")")
variable disable

If(stringmatch(ctrlName, "check1"))///Graph1
    GroupBox group11,disable=(!checked*2),win=fval

```

```

//CheckBox check1,disable=(checked+1),win=fval
PopupMenu popup2,disable=(!checked*2),win=fval
CheckBox check3,disable=1,win=fval
CheckBox check4,disable=1,win=fval
CheckBox check5,disable=1,win=fval
CheckBox check6,disable=1,win=fval
PopupMenu popup1_1,disable=1,win=fval
PopupMenu popup1_2,disable=1,win=fval
TitleBox title0,disable=1,win=fval

If(checked)
    CheckBox check3,disable=(checked*whichPLOT("single", "popup2")),win=fval
    CheckBox check4,disable=(checked*whichPLOT("single", "popup2")),win=fval
    CheckBox check5,disable=(checked*whichPLOT("single", "popup2")),win=fval
    CheckBox check6,disable=(checked*whichPLOT("single", "popup2")),win=fval
    PopupMenu popup1_1,disable=(checked*whichPLOT("ratio", "popup2")),win=fval
    PopupMenu popup1_2,disable=(checked*whichPLOT("ratio", "popup2")),win=fval
    TitleBox title0,disable=(checked*whichPLOT("ratio", "popup2")),win=fval
EndIf

elseif(stringmatch(ctrlName, "check2"))//Graph2
    GroupBox group12,disable=(!checked*2),win=fval
    //CheckBox check2,disable=(checked+1),win=fval
    PopupMenu popup3,disable=(!checked*2),win=fval

    CheckBox check7,disable=1,win=fval
    CheckBox check8,disable=1,win=fval
    CheckBox check9,disable=1,win=fval
    CheckBox check07,disable=1,win=fval
    PopupMenu popup1_3,disable=1,win=fval
    PopupMenu popup1_4,disable=1,win=fval
    TitleBox title1,disable=1,win=fval

    If(checked)
        CheckBox check7,disable=(!checked*whichPLOT("single", "popup3")),win=fval
        CheckBox check8,disable=(whichPLOT("single", "popup3")),win=fval
        CheckBox check9,disable=(whichPLOT("single", "popup3")),win=fval
        CheckBox check07,disable=(whichPLOT("single", "popup3")),win=fval
        PopupMenu popup1_3,disable=(whichPLOT("ratio", "popup3")),win=fval
        PopupMenu popup1_4,disable=(whichPLOT("ratio", "popup3")),win=fval
        TitleBox title1,disable=(whichPLOT("ratio", "popup3")),win=fval
    EndIf
EndIf

End

Function whichPLOT(str, cmp)
string str///
string cmp/// controlinfo; which popup

    monitor("whichPLOT("+str+", "+cmp+")")
    controlInfo/W=fval $(cmp)
    return (!stringmatch(str,s_value))
End

Function getroidata(infostr)
string infostr

///creates a roi around the cell; originates from mx,my (mouse location)

    monitor("getroidata("+infostr+")")

variable mX= AxisValFromPixel("fval#g0", "top",str2num(stringbykey("MOUSEX", infostr,":",";")))
variable mY= AxisValFromPixel("fval#g0", "left",str2num(stringbykey("MOUSEY", infostr,":",";")))

```

```

        controlinfo/W=fval tab0
        strswitch (s_value)
        case "Rectangle":
controlinfo/W=fval check44
            If(V_value)
                circCellproc(mx,my)
            else
                RectCell(mx,my)
            EndIf
            break
        case "Fit Cell":
            SnakeCell(mx,my)
            break
        case "Auto":
            break
        case "Manual":
            break
        Endswitch
End

Function rectCell(mx,my)
variable mx
variable my
    monitor ("rectCell("+num2str(mx)+","+num2str(my)+")")
NVAR bsh=:sbsh
NVAR bsv=:bsbv
    make/O/N=5 xw,yw
wave xw,yw
    xw[0]=mx-.5*bsh
    yw[0]=my-.5*bsv

    xw[1]=mx+.5*bsh
    yw[1]=my-.5*bsv

    xw[2]=mx+.5*bsh
    yw[2]=my+.5*bsv

    xw[3]=mx-.5*bsh
    yw[3]=my+.5*bsv

    xw[4]=mx-.5*bsh
    yw[4]=my-.5*bsv

End

Function SnakeCell(mx,my)
variable mx
variable my
    monitor ("SnakeCell("+num2str(mx)+","+num2str(my)+")")
Variable/G isDraw=1
NVAR h1=:sbsh,v1=:bsbv
SVAR picN
wave curIMG=${picN}

    duplicate/O/R=[mx-h1/2,mx+h1/2][my-v1/2,my+v1/2] curIMG, temp44

    imagertransform/O/METH=2 zProjection temp44
    //newimage/K=1 temp44
wave curi1=:M_zProjection
NVAR thr=:thr
    imagerthreshold/Q/T=(mean(cur1)+thr)/I curi1
wave curi=:M_imageThresh
    //OwnThresh("M_zProjection")
NVAR cs=:cellsize

    insertpoints/M=0 0,2, curi; curi[0][]=0;setscale/P x, (dimoffset(cur1,0)-2),1,curi; curi[0,1][]=255

```



```

insertpoints/M=0 (dimsize(curi,0)),2, curi; curi[dimsize(curi,0)-2,dimsize(curi,0)-1][]=255
insertpoints/M=1 0,2, curi; curi[][0]=0;setscale/P y, (dimoffset(curi,1)-2),1,curi; curi[][0,1]=255
insertpoints/M=1 (dimsize(curi,1)),2, curi; curi[][dimsize(curi,1)-2,dimsize(curi,1)-1]=255

imageanalyzeParticles/W/A=(cs)/Q/B stats curi

transferBoundary(mx,my)

killwaves/Z curi,curi1, temp44
End

Function AutoRecognition(ctrlName) : ButtonControl
    String ctrlName
    monitor("AutoRecognition("+ctrlName+")")
    delete_AllROIs("")
    controlinfo/W=fval check0
    If(!V_value)
        ShowMaxZ("check0",1)
    EndIf
    OwnThresh("CurlMG") ///needs boxing, thresh info
wave curi
NVAR maxCS=:Maxcellsize
NVAR MinCS=:cellsize
    imageanalyzeparticles/A=(minCS)/MAXA=(maxCS)/W/B/Q stats curi

variable p1
//print np
wave xw=:xw,yw=:yw
wave sx=w_boundaryX, sy=W_BoundaryY,BI=W_BoundaryIndex
    For(p1=0; p1<(numpnts(BI)-1); p1+=1)
        duplicate/O/R=[BI[p1],BI[p1+1]] sx,xw
        duplicate/O/R=[BI[p1],BI[p1+1]] sy,yw
        take_ROI()
    EndFor
    //ROIleg("button4")
    MakeTraces(0)
End

Function OwnThresh(waveN)
string waveN
    monitor("OwnThresh("+waveN+")")

    duplicate/O $(waveN), curi

wave curi
NVAR thr=:thr
    curi=0
NVAR bsh=:Sbsh, bsv=:Sbsv///boxsize
variable offsX=dimoffset$(waveN,0),offsY=dimoffset$(waveN,1)

variable p1, q1
For (p1=0; p1<dimsize$(waveN),1)-bsh; p1+=bsh) //hori
    For (q1=0; q1<dimsize$(waveN),0)-bsv; q1+=bsv) //verti

        duplicate/O/R=[q1,q1+bsv-1][p1,p1+bsh-1] $waveN, temp1
        wavestats/Q temp1
        IF(V_max-V_avg>thr)

            temp1=(temp1>v_avg+thr)//temp1*(temp1>v_avg+thr)
            curi[q1,q1+bsv-1][p1,p1+bsh-1]=temp1(x)(y)
            //print dimsize(temp1,0),q1,q1+bsv-1,dimsize(temp1,1),p1,p1+bsh-1

        EndIf
    EndFor

```

```

EndFor
redimension/B/U curi
curi=(curi==0)

// newimage dest
End

Function transferBoundary(mx,my)
variable mx,my
    monitor("transferBoundary("+num2str(mx)+","+num2str(my)+")")
wave by=W_BoundaryY,bx=W_BoundaryX,bl=W_BoundaryIndex,ba=W_ImageObjArea
wave W_ImageObjArea
variable num=numpts(W_ImageObjArea)
//print num
    If(num>1)//more than 1 Particle found
//        print "MORE"

        wavestats/Q ba
variable    IsEv=V_maxloc

//variable rem=1e10, cursum//BIG
        //        For(p1=0;p1<numpts(bl)-1;p1+=1)
        //            wavestats/Q/R=[Bl[p1],bl[p1+1]] bx
        //            cursum=abs(V_avg-mx)
        //            wavestats/Q/R=[Bl[p1],bl[p1+1]] by
        //            cursum+=abs(V_avg-my)
        //            If(cursum<rem)
        //                rem=cursum
        //                IsEv=p1
        //            EndIf
        //        EndFor
        duplicate/O/R=[Bl[IsEv],bl[IsEv+1]] bx,xW
        duplicate/O/R=[Bl[IsEv],bl[IsEv+1]] by,yW
    else
//        print "ONE"
        duplicate/O bx,xW
        duplicate/O by,yW
    EndIf
End

Function take_ROI()
    monitor("take_ROI()")

SVAR picN
    If(!datafolderexists(picN))
        imageGlob()///datafolder For image analysis is created
    EndIf
Variable/G IsDraw=0

wave/T wn=root:$(picN):$("roiNUM")
wave/T wC=root:$(picN):$("roiCOORD")///coordinates of therespective ROI, stored as list (",";)
wave/T wM=root:$(picN):$("roiMEAN")///not normalized; AU value avg inside the ROI, wave stored as string (list,";")
wave/T wBC=root:$(picN):$("roibCOORD")///coordinates of therespective ROI, stored as list (",";)
wave/T wBM=root:$(picN):$("roibMEAN")///not normalized; AU value avg inside the ROI, wave stored as string (list,";")
wave wA=root:$(picN):$("roiArea")
wave/T wL=root:$(picN):$("roiLoc")

wave listW=root:listW

    insertpoints (numpts(wn)), 1, wn, wC,wM,wBC,wBM,wA,wL //listW is extEnded later
    wn[numpts(wn)-1]=picN
    wC[numpts(wn)-1]=getCOORDstr()
    wM[numpts(wn)-1]=getmean()
    wA[numpts(wn)-1]=getarea("M_ROIMask")

    wL[numpts(wn)-1]=getLocstr()

```

```

        updateListW()
        listbox list0, win=fval, selrow=(numpnpts(wn)-1)
End

Function/S getCOORDstr()
wave xW,yW
monitor("getCOORDstr()")
string expo=""
variable p1

        For(p1=0; p1<numpnpts(xw); p1+=1)
                expo+=num2str(xw[p1])+","+num2str(yw[p1])+","
        EndFor
        return expo
End

Function/S getmean()
        monitor("getmean()")
wave xw,yw
SVAR picN
        dowindow/K tempDR
        newimage/N=tempDR/HIDE=1 $(picN)
        setdrawlayer/W=tempDR/K progfront
        setdrawenv/W=tempDR xcoord=top, ycoord=left
        drawpoly/W=tempDR/ABS 0,0,1,1, xw,yw

        ImageGenerateROIMask/W=tempDR/E=1/I=0 $(picN)
wave M_ROIMask

        dowindow/K tempDR
//        duplicate/O $(picN), temp444
//        redimension/S temp444
//        temp444*=(temp444>93)
//        temp444/=(temp444>93)
        imagestats/R=m_roiMask/BEAM/M=1 $(picN)// temp444

        return wave2str("W_isBeamAvg",1)
End

Function getR(infostr)
string infostr

variable MODI=str2num(stringbykey("MODIFIERS", infostr,":",";"))
string event=stringbykey("EVENT", infostr,":",";")
string sw=stringbykey("HCSPEC", infostr,":",";")

        If(stringmatch(sw, "*0"))///curlIMG win

                If(modi==5)

//print 5

                        If(stringmatch(event,"mousedown"))
                                monitor("HOOK:getR("+infostr+")")
                                getroidata(infostr) ///creates the ROI around the cell; xW,yW
                                return 1
                        EndIf
                EndIf

                If(modi==4)

//print 4
//print infostr

                        If(stringmatch(event,"mousedown"))///background ROI
                                monitor("HOOK:getR("+infostr+")")
                                ///coordinates
                                variable mX= AxisValFromPixel("fval#g0", "top",str2num(stringbykey("MOUSEX", infostr,":",";")))
                                variable mY= AxisValFromPixel("fval#g0", "left",str2num(stringbykey("MOUSEY", infostr,":",";")))

```

```

//10x10px rect
variable bsh=5
variable bsv=5

make/O/N=5 xw,yw

wave xw,yw

xw[0]=mx-.5*bsh
yw[0]=my-.5*bsv

xw[1]=mx+.5*bsh
yw[1]=my-.5*bsv

xw[2]=mx+.5*bsh
yw[2]=my+.5*bsv

xw[3]=mx-.5*bsh
yw[3]=my+.5*bsv

xw[4]=mx-.5*bsh
yw[4]=my-.5*bsv

take_bROI()
drawallbrois()
return 0
EndIf

If(stringmatch(event,"mouseup"))
    monitor("HOOK:getR("+infostr+")")
    graphNormal/W=fval#g0
    controlinfo/W=fval check44
    If(v_value)
        take_ROIproc()///creates the ROI trace and stores raw data and processes
    else
        take_ROI()///creates the ROI trace and stores raw data
    EndIf
    DisplayEverything()
    return 1
EndIf

EndIf

If(modi==8)
    //print 8
    If(stringmatch(event,"mouseup"))
        monitor("HOOK:getR("+infostr+")")
        graphWavedraw/O/L/T/W=fval#g0 yW,xW///draw mode activated
        return 0
    EndIf
EndIf

return 0

EndIf

If(stringmatch(sw, "*1"))///traces win
    If(modi==5)
        //print 5
        If(stringmatch(event,"mousedown"))
            monitor("HOOK:getR("+infostr+")")
            variable curframe=getframe(str2num(stringbykey("mousex", infostr,":",";")))
            fvalDisplayframe(curframe)

```

```

                                EndIf

                                EndIf
                                return 0
                                EndIf

End

Function/S wave2str(waven,kill)
string waven
variable kill
    monitor("wave2str("+waven+", "+num2str(kill)+")")

wave curw=root:$(waven)
string expo=""
variable p1
    For(p1=0;p1<numpts(curw);p1+=1)
        expo+=num2str(curw[p1])+";"
    EndFor
    If(kill)
        killwaves/Z curw
    EndIf
    return expo
End

Function  updateListW()
    monitor("updateListW()")
SVAR picN

wave/T wM=root:$(picN):$("roiMEAN")
wave/T wbM=root:$(picN):$("roibMEAN")

make/O/N=(numpts(wM),3)/T root:listW
    If(!numpts(wM))
        make/O/N=(1,3)/T root:listW
    EndIf
wave/T listW=root:listW
listW=""

    If(!waveexists(wM))
        return 0
    EndIf

variable p1
    For(p1=0; p1<numpts(wM);p1+=1)

        listW[p1][0]="ROI_"+num2str(p1+1)
        listW[p1][1]=stringfromlist(0,(wm[p1]),";")
        listW[p1][2]=stringfromlist(0,(wbm[p1]),";")

    EndFor
End

Function NormProc(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked
    monitor ("NormProc("+ctrlName+", "+num2str(checked)+")")
    MakeTraces(1)
End

Function FILTER()
wave curtrace
monitor("FILTER()")
SVAR picn
    duplicate/O curtrace,$("F_"+removeEnding(picn,".tif"))
wave curW=root:$( "F_"+removeEnding(picn,".tif"))

```

```

wave normW=root:'leer.tif':curtrace
curW/=normW
wave light=root:'leer.tif':light
dowindow/K $("G_" + removeEnding(picn, ".tif"))
display /K=1/N=$("G_" + removeEnding(picn, ".tif")) curW vs light as removeEnding(picn, ".tif")
ModifyGraph/W=$("G_" + removeEnding(picn, ".tif")) lsize=2
setaxis/W=$("G_" + removeEnding(picn, ".tif")) bottom, 350,680

End

Function checkgROI()
wave CurCa
monitor("checkgROI()")
wave Mini
If(waveexists(mini))
If(!stringmatch(note(mini), "fixed"))
duplicate/O curca, mini
EndIf
EndIf
wave Maxi
If(waveexists(maxi))
If(!stringmatch(note(maxi), "fixed"))
duplicate/O curca, maxi
EndIf
EndIf

End

imagertransform zProjection curca
wave M_zProjection
duplicate/O M_zProjection, root:$(FN):zIMG
killwaves/Z M_zProjection

Function fval_analyse(lasch)
string lasch
monitor("fval_analyse()")
wave traces, traces_X
wave curtrace
//SVAR s1,s2,s3,s4,s5,s6
//SVAR e1,e2,e3,e4,e5,e6
//SVAR w1,w2,w3,w4,w5,w6

SVAR pn=:picN
variable numR=numpts(root:$(pn):roiMean) ///number of rois
variable numP=numpts(curTrace)
variable p1, q1

//Creating the results
For (p1=0;p1<6;p1+=1)
make/O/N=(numR,7) root:$(pn):$("roiRes_" + num2str(p1))=nan
EndFor

If(!waveexists(root:$(pn):zIMG))
imagertransform zProjection root:$(pn)
wave M_zProjection
duplicate/O M_zProjection, root:$(pn):zIMG
killwaves/Z M_zProjection
wave zIMG=root:$(pn):zIMG
SVAR path=root:pathexp
note/K zIMG path+pn
EndIf

//delay?

wave/T w_option=root:W_options

```

```

variable delay =0
    If(str2num(w_option[8][1])) //calc with delay
        delay=calcdelay()
    EndIf

    w_option[9][1]=num2str(delay)
wave/T wm=root:$(pn):ROIMean

    For (q1=0;q1<(numR);q1+=1) ///number of rois
        //print q1
        getcurTrace(q1)
        //duplicate/O curtrace, temp44
wave temp44 =root:curtrace //is the trace of roi #q1
variable norm1=1//mean(temp44,0,5)

        For (p1=0;p1<6;p1+=1)
SVAR curS=root:$(pn):$("s"+num2str(p1+1))
SVAR curE=root:$(pn):$("e"+num2str(p1+1))
            If((strlen(curS))&&(strlen(curE)))

string curR=fval_analyseSeg(curS,curE)
//print p1, curR
wave ResW= root:$(pn):$("roiRes_" +num2str(p1))
                ResW[q1][0]=str2num(stringfromlist(0,curR,";"))/norm1 //mean seg
                ResW[q1][1]=str2num(stringfromlist(1,curR,";"))/norm1 //max seg
                ResW[q1][2]=str2num(stringfromlist(2,curR,";"))/norm1 //min seg
                ResW[q1][3]=str2num(stringfromlist(3,curR,";"))/norm1 //mean intEnded seg
                ResW[q1][4]=str2num(stringfromlist(4,curR,";"))/norm1 //bas avg
                ResW[q1][5]=str2num(stringfromlist(5,curR,";"))/norm1 //bas sd
                ResW[q1][6]=str2num(stringfromlist(6,curR,";"))/norm1 //mean max points of seg
            //print p1,q1, pn, nameofwave(resW),curR
        EndIf

    EndFor
EndFor
//killwaves/Z temp44
///All Analyses are stored in the roiDATA waves in the PicN folder
displayres(3)///men_int
displayPERC()

saveDATAFOLDER(pN)

End

Function/D Median(w, x1, x2) // Returns median value of wave w
    Wave w
    Variable x1, x2 // range of interest

    Variable result

    Duplicate/R=(x1,x2) w, tempMedianWave // Make a clone of wave
    Sort tempMedianWave, tempMedianWave // Sort clone
    SetScale/P x 0,1,tempMedianWave
    result = tempMedianWave((numpts(tempMedianWave)-1)/2)
    KillWaves tempMedianWave

    return result
End

Function/S fval_analyseSeg(s1,e1)//,w1)//,type)
string s1
string e1

    monitor("fval_analyseSeg"+"s1+"+"e1+"")
wave/T w_option=root:W_options

```

```

        If(str2num(w_option[8][1])      )
variable delay=str2num(w_option[9][1])

        s1=num2str(str2num(s1)+delay)
        e1=num2str(str2num(e1)+delay)
    EndIf

//string type///0=mean;1=min;2=max;3=mean_nested
NVAR smpl=root:smpl

wave temp44=root:curtrace
variable b1=str2num(w_option[5][1])
variable b2=str2num(w_option[6][1])

        wavestats/Q/R=[str2num(s1)*smpl-b1,str2num(s1)*smpl-b2] temp44
variable bef1=v_avg    ///beFore
variable sdbef1=V_sdev
//        print str2num(s1)*smpl-b1,str2num(s1)*smpl-b2, s1, e1, b1,b2,bef1

        wavestats/Q/R=[str2num(s1)*smpl,str2num(e1)*smpl] temp44
string expo=""
        expo+=num2str(v_avg)+";"          //avg of seg 0
        expo+=num2str(v_max)+";"          //max seg 1
        expo+=num2str(v_min)+";"          //min seg 2
variable sd1=V_sdev

variable int=(str2num(w_option[7][1])/100)*(str2num(e1)*smpl-str2num(s1)*smpl)
        wavestats/Q/R=[str2num(s1)*smpl+int,str2num(e1)*smpl-int] temp44
        expo+=num2str(v_avg)+";"          //avg intEnded seg 3

        expo+=num2str(bef1)+";"           //avg baseline 4
        expo+=num2str(sdbef1)+";"         ///SD value of the baseline 5

        duplicate/O/R=[str2num(s1)*smpl,str2num(e1)*smpl] temp44, DT44
        sort/R DT44,DT44
        expo+=num2str(mean(DT44,0,abs(b2-b1)))+";"

        return expo
End

Function implementDATA(lasch)
string lasch

string dfl= sortlist(replacestring(",","stringbykey("FOLDERS",datafolderdir(1),":",";",";"))
variable p1=1
        For(p1=0;p1<itemsinlist(dfl,";");p1+=1)//
                saveDATAFOLDER(stringfromlist(p1,dfl,";"))
        EndFor
        setdatafolder root:

End

Function saveDATAFOLDER(str)
string str ///datafoldername

        setdatafolder root:${str}
wave Cl= root:${str}:zIMG
string path=removeEnding(note(Cl),".tif")  +"_ana"
        print "Implement...."+path
//        newpath/O path1
                savedata/Q/O/D path
                print "done"
                setdatafolder root:

```


End

Function calcdelay()

SVAR pn=root:picN

SVAR ss=root:\$(pn):s6

SVAR es=root:\$(pn):e6

variable s1=str2num(ss)

variable e1=str2num(es)

wave/T w_options

variable bef1=round(str2num(w_options[10][1])/100*(e1-s1))

variable after1=round(str2num(w_options[11][1])/100*(e1-s1))

If ((!strlen(ss)) || (!strlen(es)))

return 0

EndIf

wave/T wm=root:\$(pn):ROI Mean

variable numR=numpts(root:\$(pn):roiMean) ///number of rois

make/O/N=(round(bef1+after1+((e1)-(s1)))) meanW

variable q1

For (q1=0;q1<(numR);q1+=1) ///number of rois

//print q1

getcurTrace(q1)

wave curT= root:curtrace

duplicate/O/R=[s1-bef1,e1+after1] curT temp44 ///is the trace of roi #q1

meanW+=temp44

EndFor

meanw/=numR

differentiate meanw

wavestats/Q meanw

variable delay=v_maxloc-bef1

killwaves/Z meanw, temp44

NVAR d1=root:\$(pn):delay

d1=delay

return delay

End

Function displayPERC()

monitor("displayPERC()")

SVAR pn=root:picN

wave/T wo=W_options

variable p1,q1

variable active

For (p1=0;p1<6;p1+=1)

///applications

active=0

wave ResW= root:\$(pn):\$("roiRes_" + num2str(p1))

SVAR curS=root:\$(pn):\$("s" + num2str(p1+1))

SVAR curE=root:\$(pn):\$("e" + num2str(p1+1))

variable s1=str2num(curS)

variable e1=str2num(curE)

variable delay=str2num(wo[9][1])

If(str2num(wo[8][1]))

s1+=delay

e1+=delay

EndIf

variable s2=str2num(wo[18][1])

variable e2=str2num(wo[19][1])

wo[14][1]=num2str(s1-e2)

wo[15][1]=num2str(s1-s2)

```

For(q1=0;q1<dimsize(ResW,0);q1+=1)

//wave/T wm=root:$(pn):ROI Mean
//                               getcurtrace(q1)
//wave temp45=root:curtrace

variable SD1//=calcSD()

//      print p1, q1,s1,e1,delay, s1-e2,s1-s2, SD1
//      If(ResW[q1][6]-ResW[q1][4]>4*ResW[q1][5])      //mean-bas >4*sd
//          active+=1
//      print "y",q1
//          If(p1==1)
//              print q1
//          EndIf
//      EndIf
//      EndFor

NVAR test=      root:$(pn):$("PerC"+num2str(p1))
test=(100*active/(dimsize(ResW,0)))

EndFor

End

Function calcSD()
wave temp45=root:curtrace
wave/T wo=root:w_options

If(str2num(wo[12][1]))//fix thresh
    wavestats/Q/R=[str2num(wo[18][1]),str2num(wo[19][1])] temp45
    return v_avg/temp45[0]*(str2num(wo[20][1])/100)
else
    wavestats/Q/R=[str2num(wo[14][1]),str2num(wo[15][1])] temp45
    return str2num(wo[16][1])*V_sdev/temp45[0]
EndIf

End

Function displayres(var)///men_int
variable var///0=mean;1=min;2=max;3=mean_nested; 4=SD
monitor("displayres("+num2str(var)+")")
SVAR pn=:picN
popupmenu popup4, win=fval, mode=(var+1)

If (var==4)
    var=5
EndIf

variable p1

For (p1=0;p1<6;p1+=1)      ///appliNum
SVAR curS=root:$(pn):$("s"+num2str(p1+1))
SVAR curE=root:$(pn):$("e"+num2str(p1+1))
NVAR curR=root:$(pn):$("r"+num2str(p1+1))
    curR=0
    If((strlen(curS))&&(strlen(curE)))
wave ResW= root:$(pn):$("roiRes_" + num2str(p1))
        make/O/N=(dimsize(resw,0)) temp44
        temp44=resW[p][var]

        curR=(mean(temp44))
    EndIf
//      fval_ColBG(var, p1)
EndFor

```

End

Function fval_ColBG(var1,var2)

variable var1

variable var2

monitor("fval_ColBG("+num2str(var1)+","+num2str(var2)+")")

variable r1=255,g1=255,b1=255

 If(var1==0)///mean

 r1=49152

 g1=65280

 b1=32768

 elseif(var1==1)///max

 r1=65280

 g1=16384

 b1=16384

 elseif(var1==2)///min

 r1=16384

 g1=48896

 b1=65280

 elseif(var1==3)///mean_nest

 r1=65280

 g1=65280

 b1=48896

 EndIf

 setvariable \$("setvarR"+num2str(var2+1)), win=fval, valueBackColor=(r1,g1,b1),styledText=1

End

Function fval_popResults(ctrlName,popNum,popStr) : PopupMenuControl

String ctrlName

Variable popNum

String popStr

monitor("fval_popResults("+ctrlname+","+num2str(popNum)+","+popstr+")")

displayres(popnum-1)///men_int

End

/////FILE loader

#pragma rtGlobals=3 // Use modern global access method and strict wave access.

Function ButtonProc(ctrlName) : ButtonControl

String ctrlName

monitor("ButtonProc("+ctrlName+")")

setdatafolder root:

//Creating global variables

globals1()

FrameSelectionPanel("", "", "")

End ///ButtonProc(ctrlName)

Function globals1()

///listbox selection

monitor("globals1()")

make/O/N=1 selw1=0

make/O/T/N=1 listw1=""

variable/G f1, f2

string/G pathexp///path to the images on disk

End

Function FrameSelectionPanel(str1,str2,str3)

```
string str1//suggestions /G:movsug
string str2//file list /G: filelist
string str3//pathtofile
    MONITOR("FrameSelectionPanel("+str1+", "+str2+", "+str3+"")

    dowindow/K FSP

    PauseUpdate; Silent 1          // building window...
    Display/W=(378,44,822,334.25)/K=1/N=FSP as "Select movie frames..."
    Make/O/N=(2,2) curFSP0=0
    AppEndImage curFSP0
    ModifyImage curFSP0 ctab= {*,*,Grays,0}

    ModifyGraph margin(left)=3,margin(bottom)=3,margin(top)=3,margin(right)=3,wbRGB=(56576,56576,56576)
    ModifyGraph gbRGB=(56576,56576,56576)
    ModifyGraph mirror=2
    ModifyGraph noLabel=2
    ModifyGraph lblMargin(bottom)=11
    ModifyGraph axThick=0
    ModifyGraph lblLatPos(bottom)=1
    ModifyGraph axisEnab(left)={0.09,0.87}
    ModifyGraph axisEnab(bottom)={0.4,0.98}

SVAR pathexp
    Button button0,pos={16,13},size={50,20},proc=pathbuttonproc,title="Path"
    SetVariable setvar0,pos={79,15},size={497,16},title=" "
    SetVariable setvar0,limits={-inf,inf,0},value= pathexp

wave/T listw1
wave selw1
    TabControl tab0,pos={17,45},size={201,242},proc=TabProc
    TabControl tab0,tabLabel(0)="stacked tif",tabLabel(1)="bmp folder",value= 0
    ListBox list0,pos={17,68},size={200,219},proc=ListBoxProcFSP,listWave=root:listW1
    ListBox list0,selWave=root:selw1,mode= 1,selRow= 0

NVAR f1,f2
variable /G numchan,whichchan=0, redfac=1
GroupBox group0,pos={16,295},size={202,87},labelBack=(39168,39168,39168)
    SetVariable setvar1,pos={23,302},size={64,16},bodyWidth=40,title="from"
    SetVariable setvar1,limits={-inf,inf,0},value= f1
    SetVariable setvar2,pos={94,301},size={53,16},bodyWidth=40,title="to"
    SetVariable setvar2,limits={-inf,inf,0},value= f2
    SetVariable setvar4,pos={150,302},size={62,16},bodyWidth=40,title="red.", value=redfac

    CheckBox check0,pos={28,332},size={21,14},title=" ",value= 0, proc=CheckProc_2enableFSP
    SetVariable setvar3,pos={144,332},size={68,16},bodyWidth=40,title="chan", disable=2, value=numchan
    SetVariable setvar5,pos={64,332},size={64,16},bodyWidth=40,title="num",disable=2, value=whichchan

    Button cancel,pos={44,356},size={50,20},proc=Loadframes,title="Cancel"
    Button load,pos={137,355},size={50,20},proc=loadframes,title="Load"

    Slider slider0,pos={238,360},size={338,19},proc=FSPSliderProc
    Slider slider0,labelBack=(39168,39168,39168)
    Slider slider0,limits={0,329,1},value= 178,side= 2,vert= 0,ticks= 0

End

Function CheckProc_2enableFSP(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked
    SetVariable setvar3, disable=2*(checked==0)
    SetVariable setvar5, win=FSP,disable=2*(checked==0)

End
```

```

Function pathbuttonproc(lasch):ButtonControl
string lasch
    monitor("pathbuttonproc("+lasch+")")
SVAR ptf=:pathexp
string expo=openfolder("Define a path to raw data images")

```

```

    If(!strlen(expo))//cancelled
        return 0
    EndIf

```

```

    ptf=expo
    controlinfo/W=FSP tab0
    TabProc("",V_value)
    return 1

```

```

End

```

```

Function TabProc(ctrlName,tabNum) : TabControl
    String ctrlName
    Variable tabNum
    monitor(" TabProc("+ctrlName+", "+num2str(tabNum)+")")

```

```

    Make/O/N=(2,2) curFSP0=0
    makeListW(tabnum)
    return 0

```

```

End

```

```

Function ListBoxProcFSP(ctrlName,row,col,event) : ListBoxControl

```

```

    String ctrlName
    Variable row
    Variable col
    Variable event //1=mouse down, 2=up, 3=dbl click, 4=cell select with mouse or keys
    //5=cell select with shift key, 6=begin edit, 7=End
    monitor("ListBoxProcFSP("+ctrlName+", "+num2str(row)+", "+num2str(col)+", "+num2str(event)+") ")
    If(event!=4)
        return 0
    EndIf

```

```

    selectFrame(0,row)

```

```

End

```

```

Function/S openfolder(str)
string str //message text
    monitor(" openfolder("+str+")")
    NewPath/Q/M=str/O temp

```

```

    If (V_flag)//cancelled

```

```

        return ""
    EndIf

    pathinfo temp
    return s_path

End

Function  makeListW(tabnum)
variable tabnum
///0= tlf; 1=bmp folder

    monitor("makeListW("+num2str(tabnum)+")")
    If(!pathCHCK())
        return 0
    EndIf
SVAR ptf=:pathEXP
    newpath/O/Z/Q jax1, ptf

string lstr=""
    If(!tabnum)///tifs
        lstr=sortlist(indexedfile(jax1,-1,".tlf"))
    elseif(tabnum==1)
        lstr=sortlist(indexeddir(jax1,-1,0))
    EndIf

    make/O/N=(itemsinlist(lstr)) selw1=0
    make/O/T/N=(itemsinlist(lstr)) listW1=stringfromlist(p,lstr,";")

End

Function  selectFrame(frame,row)

variable frame///frame in movie
variable row        ///in listW1
    monitor("selectFrame("+num2str(frame)+","+num2str(row)+")")
wave/T listW1=:listW1
SVAR ptf=:PathEXP
string curlIMG=ptf
variable numIMG

    controlinfo/W=FSP tab0
    If(!v_value)///stacked tlf
        curlIMG+=listW1[row]
        ImageLoad/N=test/RTIO/Q curlIMG
        killdatafolder/Z tag0
        numIMG=V_numImages
        ImageLoad/N=curFSP/O/Q/S=(frame)/C=1 curlIMG

        If(waveexists$("curFSP"+num2str(frame))))
wave killme=:$("curFSP"+num2str(frame))
            duplicate/O killme, curFSP0
            killwaves/Z killme
        EndIf
wave curFSP0

        elseif(v_value==1)///singleimages in Folder
            curlIMG+=listW1[row]+":"

```

```

        newpath/O/Q temp,curIMG
string files=indexedfile(temp,-1,".bmp")
        curIMG+=stringfromlist(frame,files,";")
        numIMG=itemsinlist(files,";")-1
        load16bitBMP(curIMG)
wave curFSP0
        If(dimsz(curFSP0, 2)>1)
                imagetransForm/P=2/O removeZplane curFSP0
                imagetransForm/P=0/O removeZplane curFSP0
        EndIf
EndIf

Slider slider0,limits={0,numIMG,1},value=(frame), win=FSP
NOTE/K curFSP0 "Frame:"+num2str(frame)
NVAR f1
NVAR f2
f1=0; f2=numIMG
setdrawlayer/K/W=FSP userfront
setdrawenv/W=FSP xcoord=prel,ycoord=prel, textrgb=(56576,56576,56576),fstyle= 1
drawtext/W=FSP .85,.2, num2str(frame)+" / "+ num2str(numIMG)

return 0
End

```

Function load16bitBMP(ptf)

```

string ptf/!="W:OLD DATA:Setup_IV:Maria:20120127:ss-20120127-001:ss_0329_001.bmp"
monitor( "load16bitBMP"+"ptf+"")
        gbloadwave/B=1/N=temp/T={32,4}/O/Q/U=10/W=1/S=18 ptf
wave temp0
variable w1=temp0[0]///width in pixels
variable h1=temp0[1]///height in pixels
        gbloadwave/B=1/N=temp/T={32,4}/O/Q/U=10/W=1/S=10 ptf
variable start=temp0[0]///height in pixels
        gbloadwave/B=1/N=temp/T={16,4}/O/Q/U=10/W=1/S=28 ptf
variable depth=temp0[0]///bit depth

        gbloadwave/B=1/N=pic/T={16,4}/O/Q/U=(h1)/W=(w1)/V=1/S=(start) ptf
make/O/N=(w1,h1) curFSP0=nan
variable p1
        For(      p1=0;p1<w1;p1+=1)
wave curW=:$("pic"+num2str(p1))
                curFSP0[p1][]=curW[q]
                killwaves/Z curW
        EndFor
        killwaves/Z temp0

//wave curFSP0
END

```

Function stackbmp(path)

```

string path
        newpath/O/Q temp,path
        monitor("stackbmp"+"path+"")
string files=sortlist(indexedfile(temp,-1,".BMP"))
string curIMG=""

        newdatafolder/O/S tempload
NVAR f1=root:f1
NVAR f2=root:f2
variable p1

```

```

        For(p1=f1;p1<f2; p1+=1)//10;p1+=1)//
            curlIMG=path+":"+stringfromlist(p1,files,";")
            load16bitBMP(curlIMG)
wave curFSP=:curFSP0

        duplicate/O curFSP, $("temp"+num2str(p1))
    EndFor
wave temp1

    imagertransform stackImages temp1
wave M_stack
    duplicate/O :M_Stack, root:curca
    setdatafolder root:
    killdatafolder/Z root:tempload
End

```

```

Function pathCHCK()
    monitor("pathCHCK()")
SVAR ptf=:pathEXP
    GetFileFolderInfo/Q/Z ptf
    If(v_flag)//FOLDER NOT FOUND!
        return pathbuttonproc("")
    EndIf

    return 1
End

```

```

Function/S getframes()
    monitor("getframes()")
string expo=""
wave curca
variable numFrames=dimsize(curca,2)-1

variable p1
    For(p1=0;p1<numFrames;p1+=1)
        expo+="Frame_"+num2str(p1)+";"
    EndFor

    return expo
End

```

```

Function FSPSliderProc(ctrlName,sliderValue,event) : SliderControl
    String ctrlName
    Variable sliderValue
    Variable event // bit field: bit 0: value set, 1: mouse down, 2: mouse up, 3: mouse moved
    monitor(" FSPSliderProc("+ctrlName+", "+num2str(sliderValue)+", "+num2str(event)+")")

    If(event==9)

        Controlinfo/W=FSP list0
        selectFrame(sliderValue,V_Value)

    EndIf
    return 0
End

```



```

Function setexpvalues(path, file)
string path
string file
//print file
    monitor("setexpvalues("+path+", "+file+"")")
wave curca
string FN=chngFN(file)
SVAR picN
NVAR wc=:whichchan

    picN=FN+"_" +num2str(wc)
    duplicate/O curca, $(FN)

    SelPICT("popup0",whichlistItem(FN,getPictlist(),",",),FN)
    checkbox check0, win=fval, value=0

    killwaves/Z curca
    imageglob()
End

```

```

Function /S chngFN(SFN)
string SFN

    monitor(" chngFN("+SFN+"")")
string expo=SFN
    If(strlen(SFN)>30)
        expo=SFN[strlen(SFN)-30,strlen(SFN)]
    EndIf
    return expo
End

```

```

Function calcmergedtrace()
monitor("calcmergedtrace()")

SVAR pn=root:picN
wave curtrace

variable numR=numpts(root:$(pn):roiMean) ///number of rois
variable numP=numpts(curTrace)

    make/O/N=(numP) mergedTrace=0

variable q1
    For (q1=0;q1<(numR);q1+=1) ///number of rois

        getcurTrace(q1)
        mergedTrace+=curTrace

    EndFor

    mergedtrace/=numR

    dowindow/K merged
    display/K=1/N=merged mergedtrace as "Average responses of " + pn

End

```

```

Function SelPICT(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum
    String popStr
    monitor("SelPICT("+ctrlName+", "+num2str(popnum)+", "+popstr+"")")
    popupmenu popup0, win=fval,mode=whichlistitem(popstr,getPICTlist(),",")+1///feedback

```

```

SVAR picN=:picN
    picN=popstr
    duplicate/O $(picN), curlIMG
    Note/K curlIMG, "0"

```

```

    updateListW()

```

```

    drawallrois()

```

```

    makeTraces(1)

```

```

NVAR zmin=root:zmin,zmax=root:zmax
    If(datafolderexists(picN))

```

```

    NVAR Smin=root:$(picN):zmin,Smax=root:$(picN):zmax
        zmin=Smin
        Zmax=Smax
        setvariable setvar0, win=fval, value=smin
        setvariable setvar1, win=fval, value=smax
        setAPPLiRes()
        SetVarProc_Appli("",0,"0", "")
        //apply2graph()
        //imageGlob()
    else
        zmin=wavemin( $(picN))
        zmax=wavemax( $(picN))
    EndIf
    ModIfyImage/W=fval#g0 curlIMG ctab= {zmin,zmax,Grays,0}

```

```

End

```

```

Function imageGlob()
    monitor("imageGlob()")

```

```

SVAR picN

```

```

    If(datafolderexists(picN))
        return 0
    EndIf

```

```

    newdatafolder/O $(picN)
    make/O/N=0/T root:$(picN):$("roiCOOR")
    make/O/N=0/T root:$(picN):$("roiMEAN")
    make/O/N=0/T root:$(picN):$("roibCOOR")
    make/O/N=0/T root:$(picN):$("roibMEAN")
    make/O/N=0/T root:$(picN):$("roiNUM")
    make/O/N=0 root:$(picN):$("roiArea")
    make/O/N=0/T root:$(picN):$("roiLoc")
    setdatafolder root:$(picN)

```

```

variable/G zmin,zmax

```

```

variable/G delay=NAN

```

```

variable p1

```

```

wave/T wo=root:W_options

```

```

    For(p1=0;p1<6;p1+=1)

```

```

    string/G :$("s"+num2str(1+p1))=wo[23+3*p1][1]

```

```

    string/G :$("e"+num2str(1+p1))=wo[24+3*p1][1]

```

```

string/G :$("w"+num2str(1+p1))=wo[22+3*p1][1]
variable/G :$("r"+num2str(1+p1))=nan
variable/G :$("Perc"+num2str(0+p1))      =nan
      EndFor

      setdatafolder root:
NVAR curmin=root:zmin
NVAR curmax=root:zmax
NVAR newmin=root:$(picN):zmin
NVAR newmax=root:$(picN):zmax
      newmin=curmin
      newmax=curmax

      For(p1=0; p1<6;p1+=1)
          setvariable $("setvarA"+num2str(p1+1)), win=fval, value=root:$(picN):$("s"+num2str(p1+1))
          setvariable $("setvarA"+num2str(p1+7)), win=fval, value=root:$(picN):$("e"+num2str(p1+1))
          setvariable $("setvarA"+num2str(p1+13)), win=fval, value=root:$(picN):$("w"+num2str(p1+1))
          setvariable $("setvarR"+num2str(p1+1)), win=fval, value=root:$(picN):$("r"+num2str(p1+1))
          setvariable $("POPPerc"+num2str(p1+1)), win=fval, value=root:$(picN):$("perc"+num2str(p1))

      EndFor

          setvariable setvar0, win=fval, value=newmin
          setvariable setvar1, win=fval, value=newmax

End

Function loadframes(lasch)
string lasch
      monitor("loadframes("+lasch+")")

wave/T listW1=:listW1
SVAR ptf=:PathEXP

variable numIMG
      controlinfo/W=FSP tab0
variable loadF=v_value
      controlinfo/W=FSP list0
variable row=V_value

      If (stringmatch(lasch, "cancel"))
          dowindow/K FSP
          killwaves/Z curFSP0,listW1,selW1
          killvariables/Z f1,f2

          return -1
      EndIf

      If(!loadF)//stacked tlf
NVAR f1=root:f1
NVAR f2=root:f2
NVAR redfac

          controlinfo/W=FSP check0

          If(V_value)
              If(!loadsinglechan(ptf+listW1[row]))//The "out of memory" problem...
                  return 0
              EndIf
          else //not checked

              ImageLoad/N=curca/O/Q/LR3D/S={f1}/C={f2} ptf+listW1[row]

wave curca

          If(redfac)//reduce size

```

```

ImageInterpolate/PXSZ={redfac,redfac} Pixelate curca
wave M_PixelatedImage
duplicate/O M_PixelatedImage,curca
EndIf

EndIf

else ///singleimages in Folder

stackbmp(ptf+listW1[row])
wave curca
If(redfac)//reduce size
ImageInterpolate/PXSZ={redfac,redfac} Pixelate curca
wave M_PixelatedImage
duplicate/O M_PixelatedImage,curca
EndIf

EndIf
dowindow/K FSP
setexpvalues(ptf,listW1[row])//copying the raw file, setting the variables

killwaves/Z curFSP0,listW1,selW1
killvariables/Z f1,f2
End

Function make_option()

string s1=""

s1+="box_x"+",";
s1+="box_y"+",";
s1+="AU thresh"+",";
s1+="cell min"+",";
s1+="cell max"+",";
s1+="beFore t1"+",";
s1+="beFore t2"+",";
s1+="int %"+",";
s1+="delay"+",";
s1+="delay val"+",";
s1+="delay calc bef %"+",";
s1+="delay calc after %"+",";

s1+="resp {SD,thresh}"+",";
s1+="SD calc {whole,int}"+",";
s1+="SDcalc s1"+",";
s1+="SDcalc e1"+",";
s1+="SD thresh"+",";
s1+="thresh calc {whole,int}"+",";
s1+="thresh calc s1"+",";
s1+="thresh calc e1"+",";
s1+="thresh"+",";
s1+=""+",";

variable p1
For(p1=0; p1<6;p1+=1)
s1+="application "+num2str(p1+1)+",";
s1+="start appli "+num2str(p1+1)+",";
s1+="End appli"+num2str(p1+1)+",";
EndFor

string s2=""

s2+="10"+",";
s2+="10"+",";
s2+="5"+",";

```

```
s2+="0"+",";
s2+="1"+",";
s2+="0"+",";
s2+="10"+",";
s2+="3"+",";
s2+="0"+",";
s2+="0"+",";
s2+="10"+",";
s2+="5"+",";
s2+=""+",";
```

[illegible]

End

```

//header
string headers=""
        headers+="experiment"+" ";
        headers+="file path"+" ";
        headers+="#cells"+" ";
        headers+="FO"+" ";
        headers+="delay"+" ";

```

```

variable p1=0
    For(p1=0;p1<6;p1+=1)
        headers+="Appl"+num2str(p1+1)+";"
        headers+="Appl"+num2str(p1+1)+"-start"+";"
        headers+="Appl"+num2str(p1+1)+"-End"+";"
        headers+="Appl"+num2str(p1+1)+"-FbeFore"+";"
        headers+="Appl"+num2str(p1+1)+"-Fmax"+";"
        headers+="Appl"+num2str(p1+1)+"-Fmin"+";"
        headers+="Appl"+num2str(p1+1)+"-Fmean"+";"
        headers+="Appl"+num2str(p1+1)+"-Fmean_int"+";"
        headers+="Appl"+num2str(p1+1)+"-resp_thresh"+";"
        headers+="Appl"+num2str(p1+1)+"-%responding"+";"
        headers+=""+";"
    EndFor

variable col1=itemsinlist(headers,";")
    make/O/T/N=(2, col1) W_Export
    W_Export[0][]=stringfromlist(q,headers,";")

//data
string dfl=sortlist(replacestring(" ",stringbykey("FOLDERS",datafolderdir(1),".:",";")))

variable q1

    For(q1=0;q1<itemsinlist(dfl,";");q1+=1)
        insertpoints (dimsize(W_Export,0)), 1,      W_Export
string curdata=""
string cda=stringfromlist(q1,dfl,";")

        If(waveexists(root:$(cda):ROI MEAN)) ///checking If this is an fval datafolder
            curdata+=cda+";"
wave cIMG=root:$(cda):zIMG
            curdata+=note(cIMG)+";"
wave/T rm=root:$(cda):ROI MEAN
            curdata+=num2str(numpts(rm))+";"
            curdata+=num2str(1)+";"
NVAR d1=root:$(cda):delay
            curdata+=num2str(d1)+";"
            For(p1=0;p1<6;p1+=1)
SVAR w1=root:$(cda):$("w"+num2str(p1+1))
                curdata+=w1+";"
SVAR s1=root:$(cda):$("s"+num2str(p1+1))
                curdata+=s1+";"
SVAR e1=root:$(cda):$("e"+num2str(p1+1))
                curdata+=e1+";"
wave rr=root:$(cda):$("ROI Res_"+num2str(p1))

                    make/O/N=(dimsize(rr,0)) temp44
                    temp44=rr[p][0]
                    curdata+=num2str(mean(temp44))+";"
                    temp44=rr[p][1]
                    curdata+=num2str(mean(temp44))+";"
                    temp44=rr[p][2]
                    curdata+=num2str(mean(temp44))+";"
                    temp44=rr[p][3]
                    curdata+=num2str(mean(temp44))+";"
                    temp44=rr[p][4]
                    curdata+=num2str(mean(temp44))+";"
                    curdata+= interpretethresh()+";"
NVAR perc0=root:$(cda):$("Perc"+num2str(p1))
                    curdata+=num2str(perc0)+";"
                    curdata+=""+";"
            EndFor
    EndIf

```

```

        W_Export[q1+1][]=stringfromlist(q,curdata,";")
    EndFor

    killwaves/z temp44
    dowindow/K expoW
    edit/K=1/N=expoW w_export

```

End

```

Function/T interpretethresh()
string expo=""
wave/T wo=root:W_options

```

```

        If(str2num(wo[12]))///abs
            expo=wo[20][1]+"% baseline elevation"
        else
            expo=wo[16][1]+"% SD of frames " +wo[14][1]+ " to " +wo[15][1]
        EndIf

```

```

        return expo

```

End

```

Function CheckProc_keep(ctrlName,checked) : CheckBoxControl
String ctrlName
Variable checked

```

```

        If(!checked)
            return 0
        EndIf

```

```

wave /T wo=root:W_options
SVAR picN=root:picN

```

variable p1

```

        For(p1=0; p1<6;p1+=1)
SVAR curS=root:${picN}:${"s"+num2str(p1+1)}
SVAR curE=root:${picN}:${"e"+num2str(p1+1)}
SVAR curW=root:${picN}:${"w"+num2str(p1+1)}
            wo[22+3*p1]=(curW)
            wo[23+3*p1]=(curS)
            wo[24+3*p1]=(curE)
        EndFor

```

End

```

Function take_bROI()
    monitor("take_bROI()")

```

```

SVAR picN
    If(!datafolderexists(picN))
        return 0////datafolder For image analysis is created
    EndIf
Variable/G IsDraw=0
    controlinfo/W=fval list0
variable row=v_value

```

```

wave/T wbC=root:$(picN):$("roibCOORD")///coordinates of the respective ROI, stored as list ("","")
wave/T wbM=root:$(picN):$("roibMEAN")///not normalized; AU value avg inside the ROI, wave stored as string (list,";")

```

```

        wbC[row]=getCOORDstr()
        wbM[row]=getmean()

```

```

        updateListW()
End

```

```

Function getcurTrace(num)

```

```

    variable num
        monitor("getcurTrace("+num2str(num)+")")

```

```

    SVAR picn=root:picN

```

```

    wave/T wm=root:$(picN):ROIbMean

```

```

    string data=wm[num]

```

```

        make/O/N=(itemsinlist(data,";")) curtrace=str2num(stringfromlist(p,data,";"))

```

```

        controlinfo/W=fval check02    ///background?

```

```

        If(v_value)

```

```

    wave/T bm=root:$(picN):ROIbMean

```

```

    string back=bm[num]

```

```

        If(strlen(back))

```

```

            make/O/N=(itemsinlist(back,";")) curBack=str2num(stringfromlist(p,back,";"))

```

```

            controlinfo/W=fval check03

```

```

            If(v_value)

```

```

    variable v1=curtrace[0]

```

```

    variable b1=curback[0]

```

```

            curtrace/=v1

```

```

            curback/=b1

```

```

        EndIf

```

```

        curtrace-=curBack

```

```

    else

```

```

        curtrace=nan

```

```

    EndIf

```

```

    EndIf

```

```

    swapChan()

```

```

    normT()

```

```

    timer()

```

```

End

```

```

Function testback()

```

```

    svar picn

```

```

    wave/T bm=root:$(picN):ROIbMean

```

```

    string back=bm[0]

```

```

        make/O/N=(itemsinlist(back,";")) curBack=str2num(stringfromlist(p,back,";"))

```

```

End

```

```

Function appEndCurTrace(waveN)

```

```

    string waveN

```

```

        monitor("appEndCurTrace("+waveN+")")

```

```

    wave traces_x, traces

```

```

    wave curtrace=:$(waveN)

```

```

NVAR nc=:numChan

```

```

    variable p1

```

```

        For(p1=0; p1<nc;p1+=1)

```

```

            insertpoints (numpts(traces)), 1, traces,traces_x

```



```

        traces [numpts(traces)-1]=NaN
        traces_x [numpts(traces)-1]=NaN
    EndFor
    insertpoints (numpts(traces)),(numpts(curtrace)), traces,traces_x
    traces [(numpts(traces))-(numpts(curtrace)), numpts(traces)-1]=curtrace[p-(numpts(traces)-numpts(curtrace))]
    make/O/N=(numpts(curtrace)) temp44=p
    temp44/=numpts(curtrace)
    temp44*=rightx(curtrace)
    traces_x [(numpts(traces))-(numpts(curtrace)), numpts(traces)-1]=temp44[p-(numpts(traces)-numpts(curtrace))]
    killwaves/Z temp44

End

```

```

Function recalc_AllTraces()
    monitor("recalc_AllTraces()")
SVAR picN=root:picN
    make/o/N=0 traces,traces_X,curTrace

wave/T st=root:$(picN):roiMEAN
    If(!waveexists(st))
        return 0
    EndIf

variable p1
    For(p1=0;p1<numpts(st);p1+=1)
        getcurTrace(p1)
        appEndCurTrace("Curtrace")
    EndFor

End

```

```

Function updateTrace(lasch):buttoncontrol
string lasch
    recalc_AllTraces()

End

```

```

Function CheckProc_back(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked
    updateTrace("")

End

```

```

Function useUpdate(lasch)
string lasch
SVAR picN

setdatafolder root:$(picN)
wave rm= root:$(picN):roiMean
    make/O/N=(numpts(rm))/T roibMean,roibCOOR
setdatafolder root:
wave/T listw
    redimension/N=(dimsize(listw,0),dimsize(listw,1),3) listw

```

```

wave selw
      redimension/N=(dimsize(listw,0),dimsize(listw,1),3) selw
End

```

Function selR()

```

variable s1 //start frame
variable e1 //End fram

prompt s1, "start frame"
prompt e1, "End frame"
doprompt "Enter range...", s1,e1

SVAR picN
wave curW=root:$(picN)
      duplicate/O/R={}[] [s1,e1] curW,test
      smooth 10,test
      imagetransForm /METH=3 zprojection test ///minimum
wave M_zProjection
      imagethreshold/M=1/I M_zProjection
wave M_ImageThresh
      imageanalyzeParticles/M=3/A=100/Maxa=1000/B/W/F/Q stats M_ImageThresh

      transferBoundaries()
End

```

Function transferBoundaries()

```

variable mx,my

wave by=W_BoundaryY
wave bx=W_BoundaryX
wave bi=W_BoundaryIndex
string rem=""
variable num=numpts(bi) ///number of ROIs
wave curimg
      dowindow/K tempDR
      newimage/N=tempDR/HIDE=1 curIMG
variable p1,q1
For(p1=0; p1<numpts(bi)-1 ; p1+=1)
      duplicate/O/R=(bi[p1],bi[p1+1]) bx, xW
      duplicate/O/R=(bi[p1],bi[p1+1]) by, yW
      q1=0
      do
            If(numtype(xw[q1]))
                  deletепoints q1,1000,xw,yw
            EndIf
            q1+=1
            while(q1<numpts(xw))
            // print p1
            take_ROI44()///creates the ROI trace and stores raw data

EndFor
dowindow/K tempDR
deletenanrois()
updatelistW()
recalc_AllTraces()
drawallROIS()

```

End

Function deletenanrois()

string rem

variable num

SVAR picN

wave/T wn=root:\$(picN):\$("roiNUM")

wave/T wC=root:\$(picN):\$("roiCOOR")

wave/T wM=root:\$(picN):\$("roiMEAN")

wave/T wbC=root:\$(picN):\$("roibCOOR")

wave wbM=root:\$(picN):\$("roibMEAN")

wave wA=root:\$(picN):\$("roiArea")

wave/T wL=root:\$(picN):\$("roiLoc")

wave curCa=root:\$(picN)

variable numP=(dimsize(curca,2)-1)

wave traces, traces_x

variable p1

For(p1=0;p1<numpts(wM);p1+=1)

If(!strlen(wm[p1]))

num=p1

deletepoints num*numP,numP+2, traces, traces_x

deletepoints num,1, wn,wc,wm,wbC,wbm,wa,wl

EndIf

EndFor

setdrawlayer/W=fval#g0/K userfront

End

Function fillxy()

wave xw,yw

variable xmin=wavemin(xw)

variable xmax=wavemax(xw)

variable ymin=wavemin(yw)

variable ymax=wavemax(yw)

//print xmin,xmax,ymin,ymax

wave m_roimask

duplicate/O/R=[xmin,xmax][ymin,ymax] m_roimask,tempM

wavestats/Q tempm

make/O/N=(V_npnts-V_sum) xw,yw

variable p1,q1,r1

For(p1=0;p1<dimsize(tempm,0);p1+=1)

For(q1=0;q1<dimsize(tempm,1);q1+=1)

If(!tempm[p1][q1])

xw[r1]=p1+xmin

yw[r1]=q1+ymin

r1+=1

EndIf

EndFor

EndFor

End

```

Function ButtonProc_1DoAutoROI(ctrlName) : ButtonControl
    String ctrlName
    If(str2num(ctrlName[6])==3)
        transferboundaries()
    EndIf
    dowindow/K w_autoROI
    killwaves/Z M_ImageThresh,M_zProjection,M_Particle
End

```

```

Function fW_autoROI(lasch):buttonControl
string lasch
Delete_allROIs("")
dowindow/K w_autoroi
    PauseUpdate; Silent 1    // building window...
wave /Z  M_zProjection,M_Particle
    If(!waveexists(M_zProjection))
        make/U/O/N=(2,2) M_zProjection
    EndIf
    If(!waveexists(M_Particle))
        make/U/O/N=(2,2) M_Particle
    EndIf

    Display /W=(327,161,935.25,587.75)/K=1/N=W_autoROI as ""

    AppEndImage/T M_zProjection
    ModifyImage M_zProjection ctab= {10,100,Grays,0}

    AppEndImage/T M_Particle
    ModifyImage M_Particle ctab= {17,63,Grays,0}
    ModifyImage M_Particle minRGB=NaN,maxRGB=NaN //invisible at beginning

    SetAxis/A/R left
    ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1
    ModifyGraph mirror=0
    ModifyGraph nticks=0
    ModifyGraph noLabel=2
    ModifyGraph standoff=0
    ModifyGraph axThick=0
    ModifyGraph axisEnab(top)={0,0.7}

    GroupBox group0,pos={580,10},size={220,60},title="Display"
    GroupBox group0,labelBack={39168,39168,39168}
        PopupMenu popup0,pos={702,36},size={60,21},bodyWidth=60,proc=PopupMenuProc_1colorM
        PopupMenu popup0,mode=19,popColor= (65280,0,0),value= #"\"*COLORPOP*\"""
        CheckBox check0,pos={603,40},size={44,14},proc=CheckProc_2showM,title="Mask"
        CheckBox check0,value= 0

    GroupBox group1,pos={580,85},size={220,200},title="zProjection"
    GroupBox group1,labelBack={39168,39168,39168}
        GroupBox group6,pos={589,110},size={203,55},title="Movie Frames"
        GroupBox group6,labelBack={39168,39168,39168}
variable/G ar_minf
variable/G ar_maxf
    SetVariable setvar4,pos={610,137},size={67,16},bodyWidth=40,title="from "
    SetVariable setvar4,limits={0,2000,0},value=ar_minf
    SetVariable setvar5,pos={712,137},size={53,16},bodyWidth=40,title="to"
    SetVariable setvar5,limits={0,2000,0},value=ar_maxf
    GroupBox group7,pos={589,180},size={203,55},title="Smooth"
    GroupBox group7,labelBack={39168,39168,39168}

```

```

        CheckBox check1,pos={610,207},size={52,14},title="smooth",value= 1,side= 1
variable/G ar_sf=10
        SetVariable setvar1,pos={693,207},size={71,16},bodyWidth=40,title="factor"
        SetVariable setvar1,limits={0,2000,0},value=ar_sf
        PopupMenu popup3,pos={599,250},size={80,21},bodyWidth=50,title="Meth."
        PopupMenu popup3,mode=1,popvalue="Max.",value= #"\"Max.;Mean;Min.\""
        Button button2,pos={717,252},size={50,20},proc=ButtonProc_1calcminiZ,title="Calculate"

GroupBox group3,pos={580,300},size={220,200},title="Particles"
GroupBox group3,labelBack=(39168,39168,39168)
        GroupBox group4,pos={589,325},size={203,55},title="Threshold"
        GroupBox group4,labelBack=(39168,39168,39168)
        PopupMenu popup2,pos={598,349},size={92,21},bodyWidth=50,title="Method "
        PopupMenu popup2,mode=1,popvalue="1",value= #"\"0;1;2;3;4;5\""
        PopupMenu popup2 proc=PopupMenuProc_1thrMeth
variable/G ar_thr =20
        SetVariable setvar0,pos={722,352},size={40,16},bodyWidth=40,title=" "
        SetVariable setvar0,limits={0,2000,0},value= ar_thr, disable=2
        GroupBox group5,pos={589,395},size={203,55},title="Particle Size"
        GroupBox group5,labelBack=(39168,39168,39168)
variable/G ar_A=100,ar_maxa=1000
        SetVariable setvar8,pos={614,422},size={61,16},bodyWidth=40,title="Min"
        SetVariable setvar8,limits={0,2000,0},value=ar_a
        SetVariable setvar9,pos={698,422},size={64,16},bodyWidth=40,title="Max"
        SetVariable setvar9,limits={0,inf,0},value=ar_maxa
        CheckBox check2,pos={617,470},size={52,14},title="Region",value= 0, proc=CheckProc_2selREG
        Button button4,pos={717,467},size={50,20},proc=ButtonProc_1calcMask,title="Calculate"

        Button button3,pos={710,520},size={80,40},title="Calculate ROIs",proc=ButtonProc_1DoAutoROI
        Button button5,pos={590,520},size={80,40},proc=ButtonProc_1DoAutoROI,title="Cancel"

```

End

```

Function CheckProc_2showM(ctrlName,checked) : CheckBoxControl
String ctrlName
Variable checked
checkbox check0, win=W_autoROI,value=(checked)
If(checked)
    controlinfo/W=W_autoROI popup0
    ModifyImage M_Particle minRGB=(V_red,V_green,V_blue),maxRGB=NaN
//invisible at beginning
else
    ModifyImage M_Particle minRGB=NaN,maxRGB=NaN //invisible at beginning
EndIf
End

```

```

Function PopMenuProc_1colorM(ctrlName,popNum,popStr) : PopupMenuControl
String ctrlName
Variable popNum
String popStr
CheckProc_2showM("check0",1)

```

End

```

Function ButtonProc_1calcminiZ(ctrlName) : ButtonControl
String ctrlName

```

```

NVAR s1=ar_minf
NVAR e1=ar_maxF
If(e1-s1<=0)
    return 0
EndIf

```

```

SVAR picN
wave curW=root:$(picN)
duplicate/O/R=[][][s1,e1] curW,test

```

```

    controlinfo/W=w_autoROI check1
    If(v_value)//smooth
NVAR smo=root:ar_sf
    smooth (smo),test
    EndIf

    controlinfo/W=W_autoROI popup3
    imagetransForm /METH=(V_value) zprojection test ///minimum
    killwaves/Z test

End

```

```

Function ButtonProc_1calcMask(ctrlName) : ButtonControl
    String ctrlName

```

```

    wave M_zProjection
    NVAR ar_thr
        controlinfo/w=w_autoROI popup2
        imagethreshold/Q/M=(v_value-1)/I/T=(ar_thr) M_zProjection
    //End

    If(v_value-1)
        ar_thr=V_threshold
    EndIf
    wave M_ImageThresh
        controlinfo/W=w_autoROI check2
        If(v_value) //select region
    wave xw,yw
        dowindow/K tempDR
        newimage/HIDE=1/N=tempDR/HIDE=1 M_ImageThresh
        setdrawlayer/W=tempDR/K progfront
        setdrawenv/W=tempDR xcoord=top, ycoord=left
        drawpoly/W=tempDR/ABS 0,0,1,1, xw,yw
        ImageGenerateROIIMask/W=tempDR/E=0/I=1 M_ImageThresh
    wave M_ROIMask
        M_ImageThresh*=M_ROIMask
        dowindow/K tempDR
    EndIf

    NVAR ar_a
    NVAR ar_maxa
        imageanalyzeParticles/M=3/A=(ar_a)/Maxa=(ar_maxa)/B/W/F/Q stats M_ImageThresh
        GraphNormal/W=W_autoROI
        CheckProc_2showM("check0",1)
End

```

```

Function PopMenuProc_1thrMeth(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum
    String popStr
    // print popnum
    setvariable setvar0,win=w_autoROI, disable=(2*(popnum!=1))
End

```

```

Function CheckProc_2selREG(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked
    If(checked)
        make/O/N=0 xw,yw
        graphWavedraw/O/L/T/W=w_autoROI yW,xW
    else

```

```

        removefromgraph/Z/W=W_autoROI xw,yw

    EndIf

End

Function take_ROI44()
    monitor("take_ROI()")

SVAR picN
    If(!datafolderexists(picN))
        imageGlob()///datafolder For image analysis is created
    EndIf
Variable/G IsDraw=0

wave/T wn=root:$(picN):$("roiNUM")
wave/T wC=root:$(picN):$("roiCOORD")///coordinates of therespective ROI, stored as list (";")
wave/T wM=root:$(picN):$("roiMEAN")///not normalized; AU value avg inside the ROI, wave stored as string (list,";")
wave/T wC=root:$(picN):$("roibCOORD")///coordinates of therespective ROI, stored as list (";")
wave/T wM=root:$(picN):$("roibMEAN")///not normalized; AU value avg inside the ROI, wave stored as string (list,";")
wave wA=root:$(picN):$("roiArea")
    If(!waveexists(wa))
        make/O/N=(numpts(wn)) root:$(picN):$("roiArea")
    EndIf
wave/T wL=root:$(picN):$("roiLoc")
    If(!waveexists(wL))
        make/O/T/N=(numpts(wn)) root:$(picN):$("roiLoc")
    EndIf
wave listW=root:listW

        insertpoints (numpts(wn)), 1, wn, wC,wM,wC,wM,wa,wl //listW is extEnded later
        wn[numpts(wn)-1]=picN
        wC[numpts(wn)-1]=getCOORDstr()

wave curlmg//=root:$(picN)
wave xw,yw

// setdrawlayer/W=tempDR/K progfront
// setdrawenv/W=tempDR xcoord=top, ycoord=left
// drawpoly/W=tempDR/ABS 0,0,1,1, xw,yw
// ImageGenerateROIMask/W=tempDR/E=1/I=0 curimg
//fillxy()
wave xw
    If(numpts(xw))
        imagestats/M=1/BRXY={xw,yw}/BEAM root:$(picN)
        wm[numpts(wn)-1]=wave2str("W_ISBeamAvg",1)

        wA[numpts(wn)-1]=getarea("tempm")
        wL[numpts(wn)-1]=getLocstr()
    else
        return 0
    EndIf
    return 1
//updateListW()
//istbox list0, win=fval, selrow=(numpts(wn)-1)
End

Function importROIs(lasch):ButtonControl
string lasch

    fw_ir()
    pauseForuser w_ir

```

End

```
Function /S getf()
string expo
expo=replacestring(" ",stringbykey("FOLDERS",datafolderdir(1),":",";"))
return expo

End
```

```
Function ButtonProc_1importROI(ctrlName) : ButtonControl
String ctrlName
```

```
    If(stringmatch(ctrlName[6],"0"))
```

```
controlinfo/W=w_ir popup0
string odf=S_value
```

```
wave/T own=root:$(odf):$("roiNUM")
wave/T owC=root:$(odf):$("roiCOOR")///coordinates of therespective ROI, stored as list (";")
wave/T owM=root:$(odf):$("roiMEAN")///not normalized; AU value avg inside the ROI, wave stored as string (list,";")
wave/T owbC=root:$(odf):$("roibCOOR")///coordinates of therespective ROI, stored as list (";")
wave/T owbM=root:$(odf):$("roibMEAN")///not normalized; AU value avg inside the ROI, wave stored as string (list,";")
wave owA=root:$(odf):$("roiArea")
wave/T owL=root:$(odf):$("roiLoc")
```

SVAR picn

```
duplicate/O      root:$(odf):$("roiNUM"),root:$(picn):$("roiNUM")
duplicate/O      root:$(odf):$("roiCOOR"),root:$(picn):$("roiCOOR")
duplicate/O      root:$(odf):$("roibCOOR"),root:$(picn):$("roibCOOR")
duplicate/O      root:$(odf):$("roiArea"),root:$(picn):$("roiArea")
duplicate/O      root:$(odf):$("roiLoc"),root:$(picn):$("roiLoc")
duplicate/O      root:$(odf):$("roiMEAN"),root:$(picn):$("roiMEAN")
duplicate/O      root:$(odf):$("roibMEAN"),root:$(picn):$("roibMEAN")
```

```
wave/T wM=root:$(picn):$("roiMEAN")///not normalized; AU value avg inside the ROI, wave stored as string (list,";")
```

```
variable p1
    For(p1=0;p1<numpts(own);p1+=1)
make/O/N=0 root:xW,root:yW
wave xw=root:xw,yw=root:yw
string curC=owc[p1]
variable q1
    For(q1=0; q1<itemsinlist(curC,";"); q1+=1)
string curCoord=stringfromlist(q1,curC,";")
        insertpoints q1, 1, xw,yw
        xw[q1]=str2num(stringfromlist(0,curcoord,";"))
        yw[q1]= str2num(stringfromlist(1,curcoord,";"))
    EndFor
imagestats/M=1/BRXY={xw,yw}/BEAM root:$(picN)
wm[p1]=wave2str("W_ISBeamAvg",1)
```

```
EndFor
```

```
deletenanrois()
updatelistW()
```



```

recalc_AllTraces()
drawallROIS()

    EndIf
dowindow/K w_ir

End

Function take_ROI44a()
    monitor("take_ROI()")

SVAR picN
    If(!datafolderexists(picN))
        imageGlob()////datafolder For image analysis is created
    EndIf
Variable/G IsDraw=0

wave/T wn=root:$(picN):$("roiNUM")
wave/T wC=root:$(picN):$("roiCOORD")////coordinates of therespective ROI, stored as list (",";)
wave/T wM=root:$(picN):$("roiMEAN")////not normalized; AU value avg inside the ROI, wave stored as string (list,",";)
wave/T wBC=root:$(picN):$("roibCOORD")////coordinates of therespective ROI, stored as list (",";)
wave/T wBM=root:$(picN):$("roibMEAN")////not normalized; AU value avg inside the ROI, wave stored as string (list,",";)
wave wA=root:$(picN):$("roiArea")
    If(!waveexists(wa))
        make/O/N=(numpts(wn)) root:$(picN):$("roiArea")
    EndIf
wave/T wL=root:$(picN):$("roiLoc")
    If(!waveexists(wL))
        make/O/T/N=(numpts(wn)) root:$(picN):$("roiLoc")
    EndIf
wave listW=root:listW

        insertpoints (numpts(wn)), 1, wn, wC,wM,wBC,wBM,wa,wL //listW is extEnded later
        wn[numpts(wn)-1]=picN
        wC[numpts(wn)-1]=getCOORDstr()

wave curlmg//=root:$(picN)
wave xw,yw

    setdrawlayer/W=tempDR/K progfront
    setdrawenv/W=tempDR xcoord=top, ycoord=left
    drawpoly/W=tempDR/ABS 0,0,1,1, xw,yw
    ImageGenerateROIMask/W=tempDR/E=1/I=0 curimg
    fillxy()
wave xw
    If(numpts(xw))
        imagestats/M=1/BRXY={xw,yw}/BEAM root:$(picN)
        wm[numpts(wm)-1]=wave2str("W_ISBeamAvg",1)

        wA[numpts(wm)-1]=getarea("tempm")
        wL[numpts(wm)-1]=getLocstr()
    else
        return 0
    EndIf
    return 1
//updateListW()
//istbox list0, win=fval, selrow=(numpts(wm)-1)
End

Function fW_ir()
    PauseUpdate; Silent 1          // building window...

```

```

dowindow/K W_ir
NewPanel /N=W_ir/W=(499,321,774,493) as "Import ROIs"
PopupMenu popup0,pos={39,47},size={200,21},bodyWidth=200
PopupMenu popup0,mode=1,popvalue="190506 sl1 cc atp.tif",value= #"getf()"
Button button0,pos={159,91},size={60,40},proc=ButtonProc_1importROI,title="Do it"
Button button1,pos={52,91},size={60,40},proc=ButtonProc_1importROI,title="Cancel"
End

```

```

Function/S getLocstr()
wave xw,yw
return(num2str(mean(xw))+";"+num2str(mean(yw)))

```

End

```

Function getarea(str)
string str // wavename

wave roi=root:$(str)//M_ROIMask
wavestats/Q roi
return (v_npnts-v_sum)

```

End

```

Function showarea(lasch):ButtonControl

string lasch
SVAR picn
wave wA=root:$(picn):roiArea
make/O/N=200 roiArea_Hist
Histogram/B={0,5,200} wa,roiArea_Hist

display/N=W_hist/K=1 roiArea_Hist as "ROI Area: "+picN
modifygraph/W=W_hist mode=5
End

```

```

Function defPNT(lasch):ButtonControl
string lasch
UserCursorAdjust("fval",10)
End

```

```

Function UserCursorAdjust(graphName,autoAbortSecs)
String graphName
Variable autoAbortSecs

wave curimg
make/O/N=1 x1,y1
appEndtograph/W=fval#g0/B/R y1 vs x1
setaxis/A top
setaxis/A/R left
setaxis bottom, 0, dimsize(curimg,1)
setaxis right, dimsize(curimg,0),0
ModifyGraph mode=3,marker=19
dowindow/F fval
SetWindow fval hook=getxy, hookevents=1

ftmp_PauseForCursor()

```

End

Function getxy(infostr)
string infostr

```
variable MODI=str2num(stringbykey("MODIFIERS", infostr,":",";"))
string event=stringbykey("EVENT", infostr,":",";")
string sw=stringbykey("HCSPEC", infostr,":",";")

wave x1,y1
print sw,stringmatch(sw,"fval#G0")
If(stringmatch(sw,"fval#G0"))
    x1[0]= AxisValFromPixel("fval#g0", "top",str2num(stringbykey("MOUSEX", infostr,":",";")))
    y1[0]= AxisValFromPixel("fval#g0", "left",str2num(stringbykey("MOUSEY", infostr,":",";")))
EndIf
return 1
End
```

```
Function UserCursorAdjust_ContButtonProc(ctrlName) : ButtonControl
    String ctrlName

    If(!str2num(ctrlName[6]))
svar picn=root:picn
svar pn1=root:pn1
wave x1=root:x1
wave y1=root:y1
        setdatafolder root:$(picN)

string/G points
        points+= pn1+"."+num2str(x1[0])+","+num2str(y1[0])+"\r"
        setdatafolder root:

    EndIf
    DoWindow/K tmp_PauseForCursor // Kill self
    dowindow/F fval
    RemoveFromGraph/Z/W=fval#G0 y1
    SetWindow fval hook=getR, hookevents=1

End
```

```
Function ftmp_PauseForCursor()
dowindow/K tmp_PauseForCursor
    PauseUpdate; Silent 1 // building window...
    NewPanel/N=tmp_PauseForCursor /K=1 /W=(351,401,601,545) as "Pause For Cursor"
    SetDrawLayer UserBack
    DrawText 3,16,"Position (click on graph)"
    DrawText 5,71,"Name of point (enter here)"
    Button button0,pos={140,115},size={92,20},proc=UserCursorAdjust_ContButtonProc,title="Do it"
    Button button1,pos={20,115},size={92,20},proc=UserCursorAdjust_ContButtonProc,title="Cancel"
string/G pn1=""
    SetVariable setvar0,pos={94,73},size={150,16},bodyWidth=150,title=" "
    SetVariable setvar0,valueColor=(65280,0,0),limits={-inf,inf,0},value=pn1
    SetVariable setvar1,pos={182,26},size={62,16},bodyWidth=50,title=" y"
    SetVariable setvar1,valueColor=(65280,0,0),limits={-inf,inf,0},value= y1[0]
    SetVariable setvar2,pos={105,26},size={59,16},bodyWidth=50,title="x"
    SetVariable setvar2,valueColor=(65280,0,0),limits={-inf,inf,0},value= x1[0]

End
```

```

Function SetVarProc_setnumchan(ctrlName,varNum,varStr,varName) : SetVariableControl
    String ctrlName
    Variable varNum
    String varStr
    String varName

    setvariable setvar5, win=fsp, limits={0,varnum-1,1}

End

Function ShowMaxZ(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked
    monitor("ShowMaxZ("+ctrlName+" "+num2str(checked)+")")
    checkbox check0, win=fval, value=checked ///feedback
wave curlmg
string rem=note(curing)
SVAR PicN

wave curW=${PicN}

    ///pauseForuser

    If(checked)

Variable s1=0,e1=dimsize(curW,2), meth=1

        If(stringmatch(ctrlName,"AUTO*"))
            s1=str2num(ctrlName[4,8])
            e1=s1+60
        else
            Prompt s1, "Enter number of first frame: "           // Set prompt For x param
            Prompt e1, "Enter number of last frame: "           // Set prompt For y param
            Prompt meth, "1=max;2=average; 3=min projection"
            DoPrompt "Specify range For zProjection", s1,e1,meth
            If (V_Flag)
                return -1                                     // User
            canceled
        EndIf
        print s1,e1
        print ((s1!=0)),((e1!=dimsize(curW,2)))
    EndIf
    If((s1!=0) || (e1!=dimsize(curW,2)))//not changed
        print "OK"
        duplicate/R=[][s1,e1]/O curw, temp44
        imagertransform/METH=(meth) zProjection temp44
        killwaves/Z temp44
    else
        print "no"
        imagertransform/METH=(meth) zProjection curw
    EndIf
wave M_zProjection

    duplicate/O M_zProjection, curlmg
    killwaves/Z M_zProjection

else

```

```

        imagetransForm/P=(str2num(rem)) getplane          curW
wave M_Imageplane
        duplicate/O          M_Imageplane          , curlIMG
        killwaves/Z M_Imageplane

    EndIf
    note/K curlIMG, rem

    controlinfo/W=fval check13
    If(V_value)//checked
wave bas
        redimension/S curlIMG
        curimg-=bas
    EndIf
NVAR If1=root:imgFilter
    If(If1>1)
        imagefilter/N=(If1) gauss curlIMG
    EndIf
End

Function mIMG(num)

variable num//start app

CheckProc_2_setbackgr("AUTO"+num2str(num),1)
ShowMaxZ("AUTO"+num2str(num),1)

wave curimg
imagefilter/N=3/O gauss curimg;imagefilter/N=3/O gauss curimg;imagefilter/N=3/O gauss curimg
duplicate/O curimg, $("curimg"+num2str(num))
dowindow/K $("W_app"+num2str(num))
newimage/K=1/S=0/N=$("W_app"+num2str(num)) $("curimg"+num2str(num))
dowindow/T $("W_app"+num2str(num)),"Responses from "+(num2str(num))+s to "+(num2str(num+60))+s (background-corrected)"
ModIfyImage $("curimg"+num2str(num)) ctab= {100,1000,Green,0}

End

Function CheckProc_2_setbackgr(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked

    checkbox check13, win=fval, value=checked
    If(!checked)
        killwaves/Z bas
        return 0
    EndIf

variable st1,et1
    If(stringmatch(ctrlName,"AUTO*"))
        et1=str2num(ctrlName[4,8])
        st1=et1-60
    else
        prompt st1, "from [s]"
        prompt et1, "to [s]"
        doprompt "Define background window", st1,et1
    EndIf
    If(V_flag)
        checkbox check13, win=Fval, value=0
        return -1
    else
svar picn

```

```

wave curM=root:$(picN)
    duplicate/R=[][st1,et1]/O curM, temp44
    imagertransform/METH=2 zProjection temp44
    killwaves/Z temp44
wave m_zProjection
    duplicate/O m_zProjection, bas
wave curlIMG
    redimension/S curlIMG
    curimg-=bas
EndIf

End

Function ListBoxProc(ctrlName,row,col,event) : ListBoxControl
    String ctrlName
    Variable row
    Variable col
    Variable event //1=mouse down, 2=up, 3=dbl click, 4=cell select with mouse or keys
    //5=cell select with shift key, 6=begin edit, 7=End
    monitor("ListBoxProc("+ctrlName+", "+num2str(row)+", "+num2str(col)+", "+num2str(event)+")")
wave/T listW
    If(row>dimsz(listw,0)-1)
        return 0
    EndIf

    If(!strlen(listW[row][col]))
        return 0
    EndIf

    If(event==4)
        If(strlen(listW[row][0]))
            DrawROI(row)
            displayCurTrace(row)
        EndIf
    EndIf
wave curtrace
    smooth 1, curtrace
    return 0
End

```

```

#pragma rtGlobals=3 // Use modern global access method and strict wave access.

```

```

Function circCellproc(mx,my)
variable mx,my

monitor ("circCellproc("+num2str(mx)+", "+num2str(my)+")")
NVAR bsh=:sbsh
NVAR bsv=:bsbv
make/O/N=21 xw=bsh*sin(x/pi)+mx,yw=bsv*cos(x/pi)+my
    xw[20]=xw[0]
    yw[20]=yw[0]

make/O/N=21 xwp=3*bsh*sin(x/pi)+mx,ywp=3*bsv*cos(x/pi)+my
    xwp[20]=xwp[0]
    ywp[20]=ywp[0]

insertpoints 21,21, xwp,ywp
    xwp[21,41]=xwp[p-21]
    ywp[21,41]=ywp[p-21]

    xwp[41]=xwp[21]

```

```

        ywp[41]=Ywp[21]

End

Function take_ROIproc()
    monitor("take_ROI()")

SVAR picN
    If(!datafolderexists(picN))
        imageGlob()////datafolder For image analysis is created
    EndIf
Variable/G IsDraw=0

wave/T wn=root:$(picN):$("roiNUM")
wave/T wC=root:$(picN):$("roiCOORD")////coordinates of therespective ROI, stored as list (";")
wave/T wM=root:$(picN):$("roiMEAN")////not normalized; AU value avg inside the ROI, wave stored as string (list,";")
wave/T wbC=root:$(picN):$("roibCOORD")////coordinates of therespective ROI, stored as list (";")
wave/T wbM=root:$(picN):$("roibMEAN")////not normalized; AU value avg inside the ROI, wave stored as string (list,";")
wave wA=root:$(picN):$("roiArea")
wave/T wL=root:$(picN):$("roiLoc")

wave listW=root:listW
wave xw,yw
    insertpoints (numpts(wn)), 1, wn, wC,wM,wbC,wbM,wa,wl //listW is extEnded later
    wn[numpts(wn)-1]=picN
    wC[numpts(wn)-1]=getCOORDstr1(xw,yw)
    wM[numpts(wn)-1]=getmean()//1(xw,yw)
    wA[numpts(wn)-1]=getarea("M_ROIMask")

    wl[numpts(wn)-1]=getLocstr1(xw,yw)

    updateListW()
    listbox list0, win=fval, selrow=(numpts(wn)-1)

    wave xwp,ywp
    insertpoints (numpts(wn)), 1, wn, wC,wM,wbC,wbM,wa,wl //listW is extEnded later
    wn[numpts(wn)-1]=picN
    wC[numpts(wn)-1]=getCOORDstr1(xwp,ywp)
    wM[numpts(wn)-1]=getmean1(xwp,ywp)
    wA[numpts(wn)-1]=getarea("M_ROIMask")

    wl[numpts(wn)-1]=getLocstr1(xw,yw)

    updateListW()
    listbox list0, win=fval, selrow=(numpts(wn)-1)

End

Function/S getmean1(xw,yw)
wave xw,yw
monitor("getmean()")
SVAR picN
wave curMOV=root:$(picN)
    duplicate/O /R=[wavemin(xw),wavemax(xw)][wavemin(yw),wavemax(yw)] curmov, mov44
    dowindow/K tempDR
    newimage/N=tempDR/HIDE=0 $("mov44")

```

```

        setdrawlayer/W=tempDR/K progfront
        setdrawenv/W=tempDR xcoord=top, ycoord=left
        drawpoly/W=tempDR/ABS 0,0,1,1, xw,yw

        ImageGenerateROIMask/W=tempDR/E=1/I=0 $("mov44")
wave M_ROIMask//e1_i0

        dowindow/K tempDR
        redimension/S mov44

        procmask()
wave m_zprojection //e0_i1
        m_roimask+=m_zprojection
        m_roimask=(m_roimask>0)
//        imagefilter/N=3 gauss3d mov44
//wavestats/Q mov44
//variable t1=6//v_avg
//pixels below t1 ==Nan
//mov44*=(mov44>t1)
//        mov44/=(mov44>t1)

        imagestats/R=m_roiMask/BEAM/M=1 $("mov44")

        killwaves/Z mov44
        return wave2str("W_isBeamAvg",1)

```

End

```

Function procmask()
wave mov44
svar picn=root:picn
SVAR s6=root:$(picn):s6
SVAR e6=root:$(picn):e6
        If((!strlen(s6)) || (!strlen(e6)))

                return 0
        EndIf
variable s1=str2num(s6)
variable e1=str2num(e6)
//print abs(s1-e1), picn

        make/O/N=(dimsize(mov44,0),dimsize(mov44,1),abs(s1-e1)) temp44
        temp44[][][]=mov44[p][q][r+min(s1,e1)]

        imagetransForm zprojection temp44
wave m_zprojection
        imagefilter/N=7 gauss m_zprojection
        imagethreshold/O/M=1/Q m_zprojection
        m_zprojection=(m_zprojection==0)

```

End

```

Function/S getLocstr1(xw,yw)
wave xw,yw
        return(num2str(mean(xw))+"," + num2str(mean(yw)))

```

End

```

Function/S getCOORDstr1(xw,yw)

```



```

wave xW,yW
monitor("getCOORDstr()")
string expo=""
variable p1

    For(p1=0; p1<numpts(xw); p1+=1)
        expo+=num2str(xw[p1])+","+num2str(yw[p1])+";"
    EndFor
    return expo
End

```

```

Function Circl(mx,my,d1)
variable mx
variable my
variable d1
Delete_allROIs("")
    monitor ("Circl("+num2str(mx)+","+num2str(my)+")")
SVAR picN
    fvalPopupMenuAction ("",1,"1")
wave curlIMG
//      imagethreshold/m=1/l curimg
//variable thr=V_threshold
wave curM=$(picN)
//      wavestats/Q curm
variable/G LLthr

    If(!LLthr)
        return 0
    EndIf

    make/O/N=0 xw,yw
Variable/G isDraw=1

nvar llres, xr,yr
//variable h1=45, v1=45///15 µm
variable h2=xr/llres, v2=h2//35 µm
variable h3=yr/llres, v3=h3//70 µm

    setdrawlayer/K/W=fval#g0 progfront
    setdrawenv/W=fval#g0 xcoord=top, ycoord=left
    SetDrawEnv/W=fval#g0 linefgc=(65280,0,0),linethick= 2.00,save

    SetDrawEnv/W=fval#g0 linefgc=(65280,0,0),fillpat=1
    SetDrawEnv fillfgc= (65280,0,0)
//      drawoval/W=fval#g0 mx-h1/2,my+v1/2,mx+h1/2,my-v1/2

```

```

        SetDrawEnv/W=fval#g0 linefgc=(65280,0,0),fillpat=0
        drawoval/W=fval#g0 mx-h2/2,my+v2/2,mx+h2/2,my-v2/2
        imagegenerateROImask/W=fval#g0/E=1/l=0 curimg
wave M_roimask
    duplicate/O m_roimask, ROIx

        SetDrawEnv/W=fval#g0 linefgc=(65280,0,0),fillpat=0
        drawoval/W=fval#g0 mx-h3/2,my+v3/2,mx+h3/2,my-v3/2
        imagegenerateROImask/W=fval#g0 curimg
wave M_roimask
    duplicate/O m_roimask, ROIy
    Killwaves/Z M_roimask

    duplicate/O curimg, temp44
    imagethreshold/O/Q/T=(LLthr) temp44
    temp44/=255
    temp44*=roix
    temp44*=roiy
    wavestats/Q temp44
variable ry0=v_sum

        make/O/N=(dimsize(curM,2)) curtrace=0
        roix+=1; roix*=(roix[p][q]==1)
variable p1=0
        For(p1=0; p1<(dimsize(curM,2)-1); p1+=1)
            fvalPopupMenuAction ("",p1+1,num2str(p1+1))
            duplicate/O curimg, temp44
            imagethreshold/O/Q/T=(LLthr) temp44
            temp44/=255
            temp44*=roix
            wavestats/Q temp44
            curtrace[p1]=v_sum
        EndFor
variable rx0=curtrace[0]

        curtrace-=rx0

        curtrace/=ry0

        appEndCurTrace("curTrace")

make/O/N=1 xw,yw
    xw[0]=mx
    yw[0]=my

    If(d1)
        //takeROI1()
    EndIf
End

```

Package 2: Ca2+ imaging secondary analysis (26.8.21)

```
#pragma rtGlobals=3          // Use modern global access method and strict wave access.

CheckBox check1,pos={891,9},size={94,14},proc=CheckProc_corr,title="Correct baseline"
CheckBox check1,value= 1

///exclude

•variable/G ThrEx=200
  CheckBox check1,pos={831,11},size={94,14},proc=CheckProc_corr,title="Correct baseline"
  CheckBox check1,value= 0
  SetVariable setvar24 title=" "
  CheckBox check2,pos={831,11},size={94,14},proc=CheckProc_corr,title="Correct baseline"
  CheckBox check2,value= 0
  SetVariable setvar24,pos={1021,8},size={50,16},limits={-inf,inf,0}, value=ThrEx
  Button button7,pos={941,8},size={70,20},title="Mark fluo4-", proc=distF0

//////////

function dup2merge(s2,e2,waven,corr, corrATP,fy)
variable s2
variable e2
string waven
variable corr
variable corrATP
variable fy//filtered
  if(!fy)
    return 0
  endif
  //print s2,e2, waven
  duplicate/O/R=(s2,e2) root:anasel:${waven}, root:merge:${waven}
wave curW=root:merge:${waven}
  controlinfo/W=cana popup5
  if(v_value==1)
    curW/=corr
  elseif(v_value==2)
    curW-=corr
    curW/=corrATP //only for subtraction
  elseif(numtype(corr)==2)
    wave tempcorr
    curW-=tempcorr
    curW/=corrATP //only for subtraction
  endif
NVAR smo44=root:smo44
  if(smo44)
    smooth (smo44), curw
  endif
  //setscale/P x,0,1 ,curW
end

function reversecorr()
wave/T lw=root:lw
variable p1=8

  for(p1=0;p1<numpts(lw); p1+=1)
wave curW=root:anasel:${lw[p1]}
string note1=note(curw)
    if(str2num(stringfromlist(0,stringbykey("corr",note(curw),":","\r"),";"))==1)
variable m1=str2num(stringfromlist(1,stringbykey("corr",note(curw),":","\r"),";"))
variable n1=str2num(stringfromlist(2,stringbykey("corr",note(curw),":","\r"),";"))
```

```

        duplicate/O curw,tempcorr
        tempcorr=m1*p+n1
        curw+=tempcorr
string newN1="0;" + num2str(m1) + ";" + num2str(n1)
        note1=replacestringbykey("corr",note1,newN1,":","\r")
        note/K curw, note1
    endif

endfor

end

function maketempcorr()

variable v1,v2//basal avg app,ATP

wave/T lw=root:lw
variable p1=8

    for(p1=0;p1<numpts(lw); p1+=1)
wave curW=root:anasel:$(lw[p1])
string note1=note(curw)
        //print "M",stringbykey("corr",note(curw),":","\r")
        if(!str2num(stringfromlist(0,stringbykey("corr",note(curw),":","\r"),";")))
(!strlen(stringbykey("corr",note(curw),":","\r"))))
||

NVAR bas=root:$( "bas" + num2str(1))
NVAR bae=root:$( "bae" + num2str(1))
NVAR bas1=root:$( "bas" + num2str(2))
NVAR bae1=root:$( "bae" + num2str(2))

        v1=mean(root:anasel:$(lw[p1]), bas,bae)//y app
        v2=mean(root:anasel:$(lw[p1]), bas1,bae1)//y ATP

        duplicate/O curw,tempcorr
variable m1=(v2-v1)/(bas1-bas)
variable n1=v1-m1*bas
        tempcorr=m1*p+n1

        curw-=tempcorr
string newN1="1;" + num2str(m1) + ";" + num2str(n1)

        note1=replacestringbykey("corr",note1,newN1,":","\r")
        note/K curw, note1

        //print "YES"

    endif

    //display curw,tempcorr

endfor

end

Function CheckProc_corr(ctrlName,checked) : CheckBoxControl
String ctrlName
Variable checked

    if(checked)

        maketempcorr()

```

```

        else
            reversecorr()
        endif

End

function rdf()

string dfl=replacestring (" ",stringbykey("FOLDERS",datafolderdir(1),":",";",";",";"))

variable p1
    for(p1=0;p1<itemsinlist(dfl,";");p1+=1)
        renamedatafolder $(stringfromlist(p1,dfl,";")),$(replacestring(" ",stringfromlist(p1,dfl,";"),"_ "))
    endfor

    dfl=replacestring (" ",stringbykey("FOLDERS",datafolderdir(1),":",";",";",";"))
    for(p1=0;p1<itemsinlist(dfl,";");p1+=1)
        renamedatafolder $(stringfromlist(p1,dfl,";")),$(replacestring(".tif",stringfromlist(p1,dfl,";"),"."))
    endfor

    dfl=replacestring (" ",stringbykey("FOLDERS",datafolderdir(1),":",";",";",";"))
    for(p1=0;p1<itemsinlist(dfl,";");p1+=1)
string n1=stringfromlist(p1,dfl,";")
string n2=n1[strlen(n1)-1]
        renamedatafolder $(stringfromlist(p1,dfl,";")),$(AddListItem(n2,n1,"_",2))
    endfor

end

function update()

dowindow/K canasel
dowindow/K W_curing
dowindow/K win_export
dowindow/K Graph0
dowindow/K Graph1
dowindow/K Graph2
dowindow/K Table0
dowindow/K Table1
dowindow/K Table2
dowindow/K w_merge
dowindow/K ttp
dowindow/K W_dist
dowindow/k W_app3

dowindow/F CANA
killcontrol/W=CANA button7
killcontrol/W=CANA check1
killcontrol/W=CANA popup2
CheckBox fil1,pos={635,176},size={52,14},title="filtered ",value= 0, win=CANA
CheckBox fil1 proc=CheckProc_usefilter, win=CANA
CheckBox fil2,pos={1115,171},size={52,14},title="filtered ",value= 0, win=CANA
CheckBox fil2 proc=CheckProc_usefilter, win=CANA

end

function /S printFilter()

string expo=""
    controlinfo/w=w_filter check5a

    if(v_value) //rel dist
NVAR n1=root:analysis:F_dist1
NVAR n2=root:analysis:F_dist2

```

```

        controlinfo/w=w_filter popup0
string poi1=s_value
        controlinfo/w=w_filter popup1
string poi2=s_value
        expo=" rel. distance (" +poi1+"-"+poi2+"): "+num2str(n1)+"% to "+num2str(n2)+"%"
        return expo
    endif

        controlinfo/w=w_filter check6a
        if(v_value) //abs dist
NVAR nA=root:analysis:F_distA
NVAR nF=root:analysis:F_distF

        controlinfo/w=w_filter popup2
string poi3=s_value
        controlinfo/w=w_filter check7
        if(v_value)//μm
            expo+=" abs. distance from "+poi3+": "+num2str(nA)+" μm"
        else
            expo+=" abs. distance from "+poi3+": "+num2str(nA)+" px"
        endif
        return expo
    endif

    return ""
end

```

```

function selectLater(var, var1,var2)
variable var//time or distance
variable var1//0=distance,1=time
variable var2//0= all merged; 1= previously selected
variable q1
string str1
    setdatafolder root:merge
    if(var2)//previously selected
        str1=wavelist("",";", "WIN:W_later")
    else
        str1=wavelist("",";", "")
    endif
    setdatafolder root:
wave/T t2p_T=root:t2p_T

wave dist
wave t2p

variable p1
    dowindow/K w_later
    display/N=w_later as "traces responding later than "+ num2str(var)+" s after STM"
    for(p1=0;p1<numpts(t2p);p1+=1)

        if(stringmatch(str1,"*" +t2p_T[p1]+"*"))
wave curw=root:merge:$(t2p_T[p1])
            if(var1)//time
                if(t2p[p1]>var)
                    appendtograph/W=w_later curw
                    q1+=1
                endif
            else //distance
                if(dist[p1]>var)
                    appendtograph/W=w_later curw
                    q1+=1
                endif
            endif
        endif
    endif
end

```

```

        endfor
print q1, " traces"
end

```

```

function selectearly(var, var1, var2)
variable var///time or distance
variable var1//0=distance,1=time
variable var2//0= all merged; 1= previously selected
variable q1
wave t2p, dist
wave/T t2p_T=:t2p_T
string str1

```

```

        setdatafolder root:merge
        if(var2)//previously selected
            str1=wavelist("*,",",", "WIN:W_later")
        else
            str1=wavelist("*,",",", "")
        endif
        setdatafolder root:

```

```

variable p1
        dowindow/K w_later
        display/K=1/N=w_later as "traces responding earlier than "+ num2str(var)+" s after STM"
        for(p1=0;p1<numpts(t2p);p1+=1)
            if(stringmatch(str1,"*" +t2p_T[p1]+"*"))
wave curw=root:merge:$(t2p_T[p1])
                if(var1)//time
                    if(t2p[p1]<var)
                        appendtograph/W=w_later curw
                        q1+=1
                    endif
                else //distance
                    if(dist[p1]<var)
                        appendtograph/W=w_later curw
                        q1+=1
                    endif
                endif
            endif
        endfor
print q1, " traces"

```

```

end

```

```

function bin(lasch)
string lasch

```

```

wave d1=root:dist
wave t1=t2p

```

```

variable last=wavemax(d1)
variable p1
NVAR bin1
        for(p1=1;p1<bin1+1;p1+=1)
make/O/N=0 root:$( "b"+num2str(p1))
        endfor

```

```

variable q1
        for(p1=0;p1<numpts(d1);p1+=1)
            for(q1=0;q1<bin1;q1+=1)
                if((d1[p1]>(last/bin1)*q1)&&(d1[p1]<(last/bin1)*(1+q1)))

```

```

wave curb=root:$( "b"+num2str(1+q1))

                                insertpoints (numpnts(curb)),1,curb
                                curb[numpnts(curb)-1]=t1[p1]
                                endif
                                endfor
                                endfor

make/O/N=(bin1) t2p_hist
                                for(p1=1;p1<bin1+1;p1+=1)
wave curb=root:$( "b"+num2str(p1))
                                wavestats/Q curb
                                t2p_hist[p1-1]=V_avg

                                // killwaves/Z curb
                                endfor
                                setscale/P x, 0,(last/bin1),t2p_hist
                                removefromgraph/Z/W=W_dist t2p_hist

                                appendtograph/W=W_dist t2p_hist
                                ModifyGraph/W=W_dist mode(t2p_hist)=5
                                ModifyGraph/W=W_dist lsize(t2p_hist)=2,hbFill(t2p_hist)=2;DelayUpdate
                                ModifyGraph/W=W_dist useBarStrokeRGB(t2p_hist)=1
                                ModifyGraph/W=W_dist rgb(t2p_hist)=(0,26112,26112)
end

```

```

function/S findloc(str)
string str//item in lw wave
wave curw=root:anel:$(str)
string info=note(curw)
//print info
string df= stringfromlist(0,info,";")
wave/T curC=root:$(df):roiCOOR
variable n1=str2num(stringfromlist(1,info,";"))

return stringfromlist (0,curC[n1],";")

```

```

end

```

```

function getdist(str1)

//from stm
string str1//wavename
variable var1///resolution

//cell location
variable cx1=str2num(stringfromlist(0,findloc(str1),",","))///cell x
variable cy1=str2num(stringfromlist(1,findloc(str1),",","))///cell y

wave curw=root:merge:$(str1)
string test1=stringbykey("points",note(curw),":","\r")
variable where=strsearch(test1,"stm",0)

variable sx1=str2num(stringfromlist(1,test1[where,1000],"_"))///STM x

variable sy1=str2num(stringfromlist(2,test1[where,1000],"_"))///STM y
//print cx1,cy1,sx1,sy1

```



```

variable a1=cx1-sx1
variable b1=cy1-sy1
//print test1
//in pixels
return sqrt(a1*a1+b1*b1)

```

```

end

```

```

Function ButtonProc_4(ctrlName) : ButtonControl

```

```

    String ctrlName
    string cana="cana"
        if(stringmatch(ctrlName,"*3"))
            cana="w_app3"
        endif

```

```

    variable n1=str2num(ctrlName[5])

```

```

    NVAR s1=root:$( "as"+num2str(n1))
    NVAR e1=root:$( "ae"+num2str(n1))
    SVAR app1=root:$( "app"+num2str(n1))

```

```

    NVAR r1=root:$( "range"+num2str(n1)+"1")
    NVAR r2=root:$( "range"+num2str(n1)+"2")

```

```

    variable s2=s1-r1
    variable e2=e1+r2

```

```

    wave/T lw=root:lw
    wave selw=root:selW

```

```

        dowindow/K w_merge
        killdatafolder/Z root:merge
        newdatafolder/O root:merge

```

```

        controlinfo/W=$(cana) $( "sel"+num2str(n1))
    variable issel=v_value
        controlinfo/W=$(cana) $( "cor"+num2str(n1))
    variable iscor=V_value
        controlinfo/W=cana check2
    variable iscorATP=V_value
        controlinfo/W=$(cana) $( "fil"+num2str(n1))
    variable isfil=v_value

```

```

    variable corr=1
    variable corrATP=1
    variable v1,v2 ,v3
    variable p1
        for(p1=0;p1<numpts(lw); p1+=1)

```

```

                if(iscor)//corrected
    NVAR bas=root:$( "bas"+num2str(n1))
    NVAR bae=root:$( "bae"+num2str(n1))
    NVAR bas1=root:$( "bas"+num2str(2))
    NVAR bae1=root:$( "bae"+num2str(2))
    NVAR bas3=root:$( "bas"+num2str(3))
    NVAR bae3=root:$( "bae"+num2str(3))
    NVAR corr2=:correction

```

```

                v1=mean(root:anasel:$(lw[p1]), bas,bae)
                v2=mean(root:anasel:$(lw[p1]), bas1,bae1)

```

```

v3=mean(root:anasel:$(lw[p1]), bas3,bae3)
if(corr2==1)
    corr=(v1+v2)/2
elseif(corr2==2)
    corr=(v1+v3)/2
elseif(corr2==0)
    corr=v1
endif
//      print bas,bae,corr
endif
if(iscorATP)//corrected for ATP
root:analysis:app2_top10
wave top10=
wave bas2=
root:analysis:bas2_avg
    corrATP=top10[p1]-bas2[p1]
//      print bas,bae,corr
endif
variable fy
if(isfil)
    fy=filteredYES(lw[p1])
else
    fy=1
endif
if(issel)//selected only
    if(selw[p1])
        dup2merge(s2,e2,lw[p1], corr, corrATP,fy)
    endif
else
        dup2merge(s2,e2,lw[p1],corr, corrATP,fy)
endif
endifor
mergeIT(e2-s2, r1,r2,num2str(n1))

NVAR ae=:$("ae"+num2str(n1))
NVAR as=:$("as"+num2str(n1))
variable num1=abs(ae-as)/10
wave wave1=:merge1
    duplicate/O wave1, temp00

    sort/R temp00,temp00
    wavestats/Q/R=[0, num1] temp00
    killwaves/Z temp00

NVAR amp=:$("amp"+num2str(n1))
    amp=v_avg

End

```

```

//getwl(),fCanaSel()

```

Menu "GraphMarquee"

```
"Calibrate response", Calibrate()
"Time to Peak", time2peak()
"Time to response", time2event()
```

End

```
function getthr(str1,var1,l1,r1)
string str1//wavename
variable var1
variable l1,r1
```

```
NVAR as1=root:as1
NVAR mas1=root:range11
//l1+=as1-mas1
//r1+=as1-mas1
///only for app1!
```

```
wave /T cell=root:analysis:$("bas"+num2str(var1)+"_cell")
wave basA=root:analysis:$("bas"+num2str(var1)+"_avg")
wave basS=root:analysis:$("bas"+num2str(var1)+"_sdev")
wave curW1=root:anasel:$(str1)
```

```
duplicate/O curw1, root:curw
wave curw=root:curw
//smooth 1, curw
```

```
NVAR f1=root:factor
FindValue /TEXT=(str1)/Z cell
variable p1=v_value
```

```
variable thr=basA[p1]+f1*basS[p1]
```

```
FindLevel /EDGE=1/Q/R=(l1-2,r1) curw, thr
variable expo=V_LevelX/(-(as1-mas1)
killwaves/Z root:curw
```

```
return expo
```

end

```
function time2event()
if(!checkWIN())
return 0
endif
getmarquee/K/W=W_merge bottom
if(!v_flag)
return 0
endif
variable l1=v_left
variable r1=v_right
```

```
variable/G res1=1
variable rr1=res1
prompt rr1, "2-photon resolution (μm/px)"
doprompt "Enter image resolution...",rr1
res1=rr1
```

```
make/O/N=0 root:t2p,root:dist
make/O/N=0/T root:t2p_T
wave m1=root:t2p ,m2=root:dist
```

```

wave/T mT=t2p_T
    setdatafolder root:merge
string wl=wavelist("*",";","")
variable p1
NVAR t1=root:range11
//print t1
    for(p1=0;p1<itemsinlist(wl,";");p1+=1)
wave curW=root:merge:${stringfromlist(p1,wl,";")}

                                insertpoints (numpts(m1)),1,m1,mT,m2
//calculate time to response
variable ma1=getthr(nameofwave(curw),1,l1,r1)
                                m1[numpts(m1)-1]=ma1
variable ma2=getdist(nameofwave(curw))
NVAR res1=root:res1
                                ma2*=res1

                                m2[numpts(m1)-1]=ma2
                                mT[numpts(m1)-1]=nameofwave(curw)
endfor
setdatafolder root:
dowindow/K ttp
edit/K=1/N=ttp mt, m1,m2 as "time to response"
ModifyTable title(mt)="Trace",title(m1)="Time to response (s)",title(m2)="Distance to STM"

string unit=""
    if(res1==1)
        unit="px"
    else
        unit="μm"
    endif
dowindow/K w_dist
display /K=1/N=W_dist m2 vs m1 as "time to response"
ModifyGraph mode=3,marker=19
ModifyGraph rgb=(0,0,0)
•SetAxis left 0,1000;DelayUpdate
SetAxis bottom wavemin(m1),wavemin(m1)+60
Label left "distance from STM [" +unit+"]";DelayUpdate
Label bottom "time to response [s]"
Button button0,pos={450,35},size={50,20},title="Bin data", proc=bin
SetVariable setvar0,pos={385,10},size={113,16},bodyWidth=40,title="number of bins"
variable/G bin1=4
SetVariable setvar0,limits={1,inf,1},value= bin1
//CurveFit/NTHR=0 line m1 /X=m2 /D
end

function checkWIN()
getwindow $"", title
    if(!stringmatch(s_value,"Average Graph"))
        print "NO"
        return 0
    else
        return 1
    endif
end

function time2peak()
    if(!checkWIN())
        return 0
    endif
getmarquee/K/W=W_merge bottom
if(!v_flag)
    return 0

```

```

endif
variable l1=v_left
variable r1=v_right

make/O/N=0 root:t2p,root:dist
make/O/N=0/T root:t2p_T
wave m1=root:t2p ,m2=root:dist
wave/T mT=t2p_T
setdatafolder root:merge
string wl=wavelist("*,",",","")
variable p1
NVAR t1=root:range11
//print t1
for(p1=0;p1<itemsinlist(wl,",");p1+=1)
wave curW=root:merge:$(stringfromlist(p1,wl,","))

wavestats/Q/R=(l1,r1) curw
variable ma1=v_maxloc
insertpoints (numpts(m1)),1,m1,mT,m2
m1[numpts(m1)-1]=ma1//t1
variable ma2=getdist(nameofwave(curw))
NVAR res1
if(NVAR_exists(res1))
ma2*=res1
endif
m2[numpts(m1)-1]=ma2
mT[numpts(m1)-1]=nameofwave(curw)
endfor
setdatafolder root:
dowindow/K ttp
edit/K=1/N=ttp mt, m1,m2 as "time to response"
ModifyTable title(mt)="Trace",title(m1)="Time to response (s)",title(m2)="Distance to STM"

dowindow/K w_dist
display /K=1/N=W_dist m1 vs m2 as "time to peak"
ModifyGraph mode=3,marker=19
ModifyGraph rgb=(0,0,0)
SetAxis left 0,60;DelayUpdate
SetAxis bottom 0,1000
Button button0,pos={450,35},size={50,20},title="Bin data", proc=bin
variable/G bin1=4
SetVariable setvar0,pos={385,10},size={113,16},bodyWidth=40,title="number of bins"
SetVariable setvar0,limits={1,inf,1},value= bin1

// CurveFit/NTHR=0 line m1 /X=m2 /D

end

function sRD(picn)
string picn //data folder

wave/T wM=root:$(picn):$("roiMEAN")////not normalized; AU value avg inside the ROI, wave stored as string (list,";")

string odf=picn

wave/T own=root:$(odf):$("roiNUM")
wave/T owC=root:$(odf):$("roiCOORD")////coordinates of therespective ROI, stored as list (";")
//print numpts (own)

make/O/N=(800,800) t1=0

```

```

dowindow/K tempDR
newimage/N=tempDR/K=1 t1
setdrawlayer/W=tempDR/K progfront
setdrawenv/W=tempDR xcoord=top, ycoord=left,save

variable p1
variable q1
    for(p1=0;p1<numpnts(own);p1+=1)
        make/O/N=0 root:xW,root:yW
wave xw=root:xw,yw=root:yw
string curC=owc[p1]
    //print itemsinlist(curC),p1,stringfromlist(0,curC,";")

    for(q1=0; q1<itemsinlist(curC,";"); q1+=1)
string curCoord=stringfromlist(q1,curC,";")
        insertpoints q1, 1, xw,yw
        xw[q1]=round(str2num(stringfromlist(0,curCoord,";")))
        yw[q1]=round(      str2num(stringfromlist(1,curCoord,";"))      )
    endfor

drawpoly/W=tempDR/ABS 0,0,1,1, xw,yw
ImageGenerateROIMask/W=tempDR/E=0/I=1 t1
wave M_ROIMask
    t1+=M_roimask
    endfor
dowindow/T tempDR,odf
end

function concatenatefolders(lasch)
string lasch
    fconcat()
end

function fCanasel()

dowindow/K canasel

PauseUpdate; Silent 1 // building window...
Display/K=1 /W=(336,65.75,1125,529.25)/N=canasel as "Experiment Browser"
ModifyGraph margin(right)=240,wbRGB=(60928,60928,60928),gbRGB=(56576,56576,56576)

GroupBox group0,pos={766,0},size={288,622},labelBack={52224,52224,52224}

    GroupBox group6,pos={770,1},size={280,70},title="Experiment"
    GroupBox group6,labelBack={43520,43520,43520},fColor={4352,4352,4352}
        PopupMenu popup0,pos={818,32},size={150,21},bodyWidth=150,proc=PopupMenuProc_1
        PopupMenu popup0,mode=33,value= #"getwl()"
        Button button2,pos={978,32},size={30,20},proc=nexlas,title="<"
        Button button3,pos={1012,32},size={30,20},proc=nexlas,title=">"
        Button button5,pos={778,32},size={35,20},proc=concatenatefolders,title="merge"

string/G s_mouse
string/G s_age
string/G s_sex
string/G s_region
string/G s_geno

    GroupBox group7,pos={770,71},size={280,150},title="Properties"
    GroupBox group7,labelBack={43520,43520,43520},fColor={4352,4352,4352}

        SetVariable setvarA19,pos={776,99},size={138,16},bodyWidth=100,proc=SetVarProc_1,title=" mouse"
        SetVariable setvarA19,limits={-inf,inf,0},value= s_mouse
        SetVariable setvarA20,pos={922,99},size={122,16},bodyWidth=100,proc=SetVarProc_1,title="age"
        SetVariable setvarA20,limits={-inf,inf,0},value= s_age
        SetVariable setvarA21,pos={794,143},size={120,16},bodyWidth=100,proc=SetVarProc_1,title="sex"

```

```

SetVariable setvarA21,limits={-inf,inf,0},value= s_sex
SetVariable setvarA22,pos={781,189},size={133,16},bodyWidth=100,proc=SetVarProc_1,title="region"
SetVariable setvarA22,limits={-inf,inf,0},value= s_region
SetVariable setvarA23,pos={919,143},size={125,16},bodyWidth=100,proc=SetVarProc_1,title="gen."
SetVariable setvarA23,limits={-inf,inf,0},value= s_geno
Button button6,pos={947,188},size={80,20},title="From previous",proc=fromPrev

```

```

PopupMenu popup1,pos={894,141},size={20,21},bodyWidth=20,proc=PopupMenuProc_2
PopupMenu popup1,mode=1,popvalue="Male",value= #"\"Male;Female\"""
PopupMenu popup2,pos={894,187},size={20,21},bodyWidth=20,proc=PopupMenuProc_2
PopupMenu popup2,mode=2,popvalue="Corpus Callosum",value= #"getreg()"
PopupMenu popup3,pos={1024,141},size={20,21},bodyWidth=20,proc=PopupMenuProc_2
PopupMenu popup3,mode=2,popvalue="Csf1r-2A-mCherry-2A-GCaMP6m",value= #"getgeno()"

```

```

String/G s1
String/G s2
String/G s3
String/G s4
String/G s5
String/G s6
String/G e1
String/G e2
String/G e3
String/G e4
String/G e5
String/G e6
String/G w1
String/G w2
String/G w3
String/G w4
String/G w5
String/G w6

```

```

GroupBox group4,pos={770,221},size={280,210},title="Application"
GroupBox group4,labelBack=(43520,43520,43520),fColor=(4352,4352,4352)
SetVariable setvarA1,pos={795,251},size={35,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA1,limits={-inf,inf,0},value= s1
SetVariable setvarA7,pos={839,251},size={35,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA7,limits={-inf,inf,0},value= e1
SetVariable setvarA13,pos={884,251},size={150,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA13,limits={-inf,inf,0},value= w1
SetVariable setvarA2,pos={795,271},size={35,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA2,limits={-inf,inf,0},value= s2
SetVariable setvarA8,pos={839,271},size={35,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA8,limits={-inf,inf,0},value= e2
SetVariable setvarA14,pos={884,271},size={150,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA14,limits={-inf,inf,0},value= w2
SetVariable setvarA3,pos={795,291},size={35,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA3,limits={-inf,inf,0},value= s3
SetVariable setvarA9,pos={839,291},size={35,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA9,limits={-inf,inf,0},value= e3
SetVariable setvarA15,pos={884,291},size={150,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA15,limits={-inf,inf,0},value= w3
SetVariable setvarA4,pos={795,311},size={35,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA4,limits={-inf,inf,0},value= s4
SetVariable setvarA10,pos={839,311},size={35,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA10,limits={-inf,inf,0},value= e4
SetVariable setvarA16,pos={884,311},size={150,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA16,limits={-inf,inf,0},value= w4
SetVariable setvarA5,pos={795,331},size={35,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA5,limits={-inf,inf,0},value= s5
SetVariable setvarA11,pos={839,331},size={35,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA11,limits={-inf,inf,0},value= e5
SetVariable setvarA17,pos={884,331},size={150,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA17,limits={-inf,inf,0},value= w5
SetVariable setvarA6,pos={795,354},size={35,16},proc=SetVarProc_Appli1,title=" "

```

```

SetVariable setvarA6,limits={-inf,inf,0},value= s6
SetVariable setvarA12,pos={839,354},size={35,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA12,limits={-inf,inf,0},value= e6
SetVariable setvarA18,pos={884,354},size={150,16},proc=SetVarProc_Appli1,title=" "
SetVariable setvarA18,limits={-inf,inf,0},value= w6
PopupMenu app1,pos={1015,249},size={20,21},bodyWidth=20,proc=PopMenuProc_3enterAPP
PopupMenu app1,mode=2,popvalue="Female",value= #"getapp()"
PopupMenu app2,pos={1015,269},size={20,21},bodyWidth=20,proc=PopMenuProc_3enterAPP
PopupMenu app2,mode=2,popvalue="Female",value= #"getapp()"
PopupMenu app3,pos={1015,289},size={20,21},bodyWidth=20,proc=PopMenuProc_3enterAPP
PopupMenu app3,mode=2,popvalue="Female",value= #"getapp()"
PopupMenu app4,pos={1015,309},size={20,21},bodyWidth=20,proc=PopMenuProc_3enterAPP
PopupMenu app4,mode=2,popvalue="Female",value= #"getapp()"
PopupMenu app5,pos={1015,329},size={20,21},bodyWidth=20,proc=PopMenuProc_3enterAPP
PopupMenu app5,mode=2,popvalue="Female",value= #"getapp()"
PopupMenu app6,pos={1015,349},size={20,21},bodyWidth=20,proc=PopMenuProc_3enterAPP
PopupMenu app6,mode=2,popvalue="Female",value= #"getapp()"
Button button7,pos={951,397},size={80,20},proc=fromPrev,title="From previous"


GroupBox group5,pos={770,431},size={280,100},title="Display"
GroupBox group5,labelBack=(43520,43520,43520),fColor=(4352,4352,4352)


CheckBox bckgr4,pos={782,468},size={37,14},proc=CheckProcBCKGR_2,title="ROI"
CheckBox bckgr4,value= 1,mode=1
CheckBox bckgr2,pos={782,498},size={75,14},proc=CheckProcBCKGR_2,title="background"
CheckBox bckgr2,value= 0,mode=1


CheckBox bckgr1,pos={882,499},size={71,14},proc=CheckProcBCKGR_1,title="normalized "
CheckBox bckgr1,value= 1
CheckBox bckgr3,pos={882,469},size={86,14},proc=CheckProcBCKGR_1,title="bg subtracted "
CheckBox bckgr3,value= 1


Button button4,pos={992,469},size={40,40},proc=ButtonProc_showZIMG,title="img"


GroupBox group8,pos={770,537},size={280,77},labelBack=(43520,43520,43520)
GroupBox group8,fColor=(4352,4352,4352)
Button button0,pos={939,550},size={100,20},proc=ButtonProc_1,title="Append"
Button button1,pos={779,550},size={100,20},proc=ButtonProc_1,title="Cancel"
Button button02,pos={818,586},size={90,20},proc=definecelltypes,title="discriminate"
SetVariable setvar0,pos={780,588},size={30,16},title=" "
Variable/G mcthr
SetVariable setvar0,limits={-inf,inf,0},value= mcthr
Button button8,pos={914,586},size={60,20},proc=ButtonProc_1posneg,title="mCherry-"
Button button8,fColor=(0,13056,0)
Button button9,pos={977,585},size={60,20},proc=ButtonProc_1posneg,title="mCherry+"
Button button9,labelBack=(39168,13056,0),fColor=(39168,13056,0)


End

function fconcat()
    PauseUpdate; Silent 1          // building window...
    dowindow/K concat
    NewPanel/N=concat/K=1 /W=(712,108,999,374)
    PopupMenu popup0,pos={45,23},size={183,21},bodyWidth=150,proc=PopMenuProc_4concat,title="data 1"
    PopupMenu popup0,mode=33,popvalue="190802_sl4_CA1_TTX_ana",value= #"getwl()"
    PopupMenu popup1,pos={46,57},size={183,21},bodyWidth=150,title="data 2"
    PopupMenu popup1,mode=33,popvalue="190802_sl4_CA1_TTX_ana",value= #"getwl()"

variable/G space
    SetVariable setvar0,pos={41,101},size={186,16},bodyWidth=150,title="spacer"
    SetVariable setvar0,format="%g s", value=space

string/G nname1
    SetVariable setvar1,pos={25,170},size={203,16},bodyWidth=150,title="new name"
    SetVariable setvar1,limits={-inf,inf,0}, value=nname1

```



```

        Button button0,pos={170,200},size={60,40},title="Do it", proc=ButtonProc7_concat
        Button button1,pos={80,200},size={60,40},title="Cancel", proc=ButtonProc7_concat
EndMacro

```

```

Function PopMenuProc_4concat(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum
    String popStr

    svar nn1=:nname1
        nn1=popstr
end

```

```

function ButtonProc7_concat(lasch)
string lasch

        if(stringmatch(lasch[6],"0")) //do it
            controlinfo/W=concat popup0
string n1=s_value
            controlinfo/W=concat popup1
string n2=s_value
NVAR s1=root:space
SVAR nn=root:nname1

            concatenatedf(n1,n2,s1,nn)
        endif

        dowindow/K concat
end

```

```

function concatenatedf(s1,s2,v1,s3)
string s1 //folder 1
string s2//folder 2
variable v1//space
string s3//new folder name

        if(stringmatch(s1,s2))
            abort "Identical source data folders!"
            return 0
        endif
string sp=createSP(v1)

//roibMean
//roiMean

```

```

wave/T w1=root:$(s1):roiMEAN
variable l1=itemsinlist(w1[0],",")
wave/T w2=root:$(s2):roiMEAN
        if(numpts(w1)!=numpts(w2))
            print numpts(w1),numpts(w2)
            abort "Source data folders contain a different number of cells!"
            return 0
        endif

        w1+=sp
        w1+=w2

wave/T bw1=root:$(s1):roibMEAN
wave/T bw2=root:$(s2):roibMEAN
        bw1+=sp
        bw1+=bw2

```

```

//strings s,e,w
string ss=""
string es=""
string ws=""
variable p1
//folder 1
    for(p1=0;p1<6;p1+=1)
SVAR cw=root:$(s1):$("w"+num2str(p1+1))
SVAR cs=root:$(s1):$("s"+num2str(p1+1))
SVAR ce=root:$(s1):$("e"+num2str(p1+1))
        if(strlen(cw))
            ws+=cw+",";
            ss+=cs+",";
            es+=ce+",";
        endif
    endfor
//folder 2
    for(p1=0;p1<6;p1+=1)
SVAR cw=root:$(s2):$("w"+num2str(p1+1))
SVAR cs=root:$(s2):$("s"+num2str(p1+1))
SVAR ce=root:$(s2):$("e"+num2str(p1+1))
        if(strlen(cw))
variable v_cs=str2num(cs)
variable v_ce=str2num(ce)
            v_cs+=l1+v1//space plus length first data
            v_ce+=l1+v1//space plus length first data
            ws+=cw+",";
            ss+=num2str(V_cs)+",";
            es+=num2str(V_ce)+",";
        endif
    endfor
//new
    for(p1=0;p1<6;p1+=1)
SVAR nw=root:$(s1):$("w"+num2str(p1))
SVAR ns=root:$(s1):$("s"+num2str(p1))
SVAR ne=root:$(s1):$("e"+num2str(p1))
        nw=stringfromlist(p1,ws,",");
        ns=stringfromlist(p1,ss,",");
        ne=stringfromlist(p1,es,",");
    endfor

    renamedatafolder root:$(s1),$(s3)
    if(!stringmatch(s1,s3))
        killdatafolder/Z root:$(s1)
    endif
    killdatafolder/Z root:$(s2)
end

```

```

function/S createSP(var)
variable var
string expo=""
variable p1
    for(p1=0;p1<var;p1+=1)
        expo+=",";
    endfor
return expo
end

```

```

function /S getwl()

```

```

string expo=replacestring(" ",stringbykey("FOLDERS",datafolderdir(1),":",";"))
expo=removefromlist("Analysis", expo,";",0)
expo=removefromlist("Anasel", expo,";",0)
expo=removefromlist("curanasel", expo,";",0)
expo=removefromlist("merge", expo,";",0)
expo=removefromlist("Packages", expo,";",0)

```

```

return sortlist(expo)

end

```

```

menu "analysis"
" _ "
"Calcium",fCANA()
end

```

```

function mergetraces2()

wave traces2, traces_x2
variable e1= wavemax(traces_X2)

make/O/N=(e1) mergeW1=0,semW1,nW1
variable p1,q1,sum1,x1
for (p1=0;p1<e1;p1+=1)

findlevels/Q traces_x2,p1
wave W_FindLevels
make/O/N=(V_LevelsFound) temp44
for(q1=0;q1<V_LevelsFound;q1+=1)
temp44[q1]=traces2[W_FindLevels[q1]]
endfor
wvstats/Q temp44

mergeW1[p1]=v_avg
semW1[p1]=V_SEM
nW1[p1]=v_npnts

endfor

killwaves/Z temp44,W_FindLevels

end

```

```

function normT1()
//monitor("normT()")

wave curtrace=:CurTrace
variable norm1=curtrace[0]
curtrace/=norm1

end

```

```

Function appendCurTrace2(waveN)
string waveN
    //monitor("appendCurTrace("+waveN+"")")
wave traces_x=traces_x2, traces=traces2
wave curtrace=:$(waveN)

NVAR nc=:numChan
variable p1
    for(p1=0; p1<nc;p1+=1)
        insertpoints (numpnts(traces)), 1, traces,traces_x
        traces [numpnts(traces)-1]=NaN
        traces_x [numpnts(traces)-1]=NaN
    endfor
    insertpoints (numpnts(traces)),(numpnts(curtrace)), traces,traces_x
    traces [(numpnts(traces))-(numpnts(curtrace)), numpnts(traces)-1]=curtrace[p-(numpnts(traces)-numpnts(curtrace))]
    make/O/N=(numpnts(curtrace)) temp44=p
    temp44/=numpnts(curtrace)
    temp44*=rightx(curtrace)
    traces_X [(numpnts(traces))-(numpnts(curtrace)), numpnts(traces)-1]=temp44[p-(numpnts(traces)-numpnts(curtrace))]
    killwaves/Z temp44

```

End

```

Function appendCurTrace1(waveN)
string waveN
    //("appendCurTrace("+waveN+"")")
wave traces_x=traces_x1, traces=traces1
wave curtrace=:$(waveN)

```

```

NVAR nc=:numChan
variable p1
    for(p1=0; p1<nc;p1+=1)
        insertpoints (numpnts(traces)), 1, traces,traces_x
        traces [numpnts(traces)-1]=NaN
        traces_x [numpnts(traces)-1]=NaN
    endfor
    insertpoints (numpnts(traces)),(numpnts(curtrace)), traces,traces_x
    traces [(numpnts(traces))-(numpnts(curtrace)), numpnts(traces)-1]=curtrace[p-(numpnts(traces)-numpnts(curtrace))]
    make/O/N=(numpnts(curtrace)) temp44=p
    temp44/=numpnts(curtrace)
    temp44*=rightx(curtrace)
    traces_X [(numpnts(traces))-(numpnts(curtrace)), numpnts(traces)-1]=temp44[p-(numpnts(traces)-numpnts(curtrace))]
    killwaves/Z temp44

```

End

```

function fGraph0()

```

```

make/O/N=1 traces1,traces_X1
    PauseUpdate; Silent 1          // building window...
    Display /W=(402.75,140,900,509.75) traces1 vs traces_x1
    PopupMenu popup0,pos={6,7},size={150,21},bodyWidth=150,proc=PopupMenuProc_1
    PopupMenu popup0,mode=1,value=getwl()
    Button button0,pos={166,7},size={50,20},title="Append",proc=ButtonProc_1
    •Button button1 title="All",pos={230,7},size={50,20}
    •Button button1 proc=ButtonProc_appendall
make/O/N=1 traces2,traces_X2
    Display /W=(808.5,83.75,1329,447.5) traces2 vs traces_x2 as "EXPORT"

End

```

```

function action()

    getmarquee/Z/W=graph2 bottom

    if(!V_flag)
        return 0
    endif
variable l1=round(V_left),r1=round(V_right)
wave traces2,traces_x2
    findlevels/Q traces_x2,0
wave W_FindLevels
    make/O/N=(V_LevelsFound) temp44=nan

    duplicate/O traces2, curW_e

variable p1
    for(p1=0;p1<V_LevelsFound-1;p1+=1)
        findlevel/Q/R=[W_FindLevels[p1],W_FindLevels[p1+1]-1] traces_x2,l1
variable l2=v_levelX
        findlevel/Q/R=[W_FindLevels[p1],W_FindLevels[p1+1]-1] traces_x2,r1
variable r2=v_levelX
        wavestats/Q/R=[l2,r2] traces2

        temp44[p1]=v_max

        curW_e[W_FindLevels[p1],W_FindLevels[p1+1]-1]=v_min-1

    endfor

end

```

```

function fW_analysis()

string/G app1,app2
variable/G bas1,bae1,bas2,bae2
variable/G mb1,mb2
variable/G as1,ae1,as2,ae2

```

```

variable/G ma1,ma2
variable/G nc1,nc2,ne1,ne2
variable /G resp1,resp2
variable /G amp1,amp2
variable /G thresh
    dowindow/K w_analysis
    PauseUpdate; Silent 1          // building window...
    NewPanel/K=1 /W=(335,144,864,556)/N=W_analysis as "Analysis"

    Button ba5,pos={373,18},size={50,20},proc=Get_3SD,title="get 3*SD"
    SetVariable setvar16,pos={431,20},size={60,16},bodyWidth=60,title=" "
    SetVariable setvar16,limits={-inf,inf,0},value= thresh

    GroupBox group0,pos={5,44},size={518,147},title="application 1"
    GroupBox group0,labelBack={56576,56576,56576},frame=0

        SetVariable setvar0,pos={15,106},size={70,16},limits={-inf,inf,0},value=app1, title=" "

        GroupBox group1,pos={97,68},size={89,80},title="basal"
        GroupBox group1,labelBack={30464,30464,30464}
            SetVariable setvar2,pos={106,93},size={32,16},limits={-inf,inf,0}, value=bas1, title=" "
            SetVariable setvar3,pos={144,93},size={32,16},limits={-inf,inf,0}, value=bae1, title=" "
            SetVariable setvar1,pos={106,123},size={70,16},limits={-inf,inf,0}, title=" ",value=mb1
        Button ba1,pos={118,154},size={50,20},title="get basal", proc= getM

        GroupBox group2,pos={187,68},size={86,81},title="appl."
        GroupBox group2,labelBack={30464,30464,30464}
            SetVariable setvar4,pos={196,93},size={32,16},limits={-inf,inf,0}, title=" ",value=as1
            SetVariable setvar5,pos={234,93},size={32,16},limits={-inf,inf,0}, title=" ",value=ae1
            SetVariable setvar6,pos={196,123},size={70,16},limits={-inf,inf,0}, title=" ",value=ma1
        Button ba2,pos={205,154},size={50,20},title="get appl.", proc= getM

        GroupBox group3,pos={277,68},size={86,81},title="stats"
        GroupBox group3,labelBack={30464,30464,30464}
            SetVariable setvar07,pos={285,92},size={70,16},limits={-inf,inf,0},value=nc1, title=" ", format="%g cells"
            SetVariable setvar08,pos={285,122},size={70,16},limits={-inf,inf,0},value=ne1, title=" ", format="%g exp."

        GroupBox group4,pos={367,66},size={152,83},title="reuslts"
        GroupBox group4,labelBack={30464,30464,30464}
            SetVariable setvar05,pos={394,92},size={100,16},limits={-inf,inf,0}, title=" ",value=resp1, format="%g%"
            SetVariable setvar06,pos={394,122},size={100,16},limits={-inf,inf,0}, title=" ",value=amp1, format="%g AU"
        CheckBox check0,pos={371,161},size={64,14},title="relative to",value= 0
        PopupMenu popup0,pos={445,157},size={50,21},bodyWidth=50
        PopupMenu popup0,mode=1,popvalue="1",value= #"1;2\"

    GroupBox group5,pos={5,204},size={518,147},title="application 2"
    GroupBox group5,labelBack={56576,56576,56576},frame=0

        SetVariable setvar7,pos={15,266},size={70,16},limits={-inf,inf,0},value=app2, title=" "

        GroupBox group6,pos={97,228},size={89,80},title="basal"
        GroupBox group6,labelBack={30464,30464,30464}
            SetVariable setvar8,pos={106,253},size={32,16},limits={-inf,inf,0}, value=bas2, title=" "
            SetVariable setvar9,pos={144,253},size={32,16},limits={-inf,inf,0}, value=bae2, title=" "
            SetVariable setvar02,pos={106,283},size={70,16},limits={-inf,inf,0}, value=mb2, title=" "
        Button ba3,pos={118,314},size={50,20},title="get basal", proc= getM

        GroupBox group7,pos={187,228},size={86,81},title="appl."
        GroupBox group7,labelBack={30464,30464,30464}
            SetVariable setvar09,pos={196,253},size={32,16},limits={-inf,inf,0}, value=as2, title=" "
            SetVariable setvar10,pos={234,253},size={32,16},limits={-inf,inf,0}, value=ae2, title=" "
            SetVariable setvar11,pos={196,283},size={70,16},limits={-inf,inf,0}, value=ma2, title=" "
        Button ba4,pos={205,314},size={50,20},title="get appl.", proc= getM

```

```

GroupBox group8,pos={277,228},size={86,81},title="stats"
GroupBox group8,labelBack={30464,30464,30464}
    SetVariable setvar12,pos={285,252},size={70,16},limits={-inf,inf,0}, value=nc2, title=" ", format="%g cells"
    SetVariable setvar13,pos={285,282},size={70,16},limits={-inf,inf,0}, value=ne2, title=" ", format="%g exp."

GroupBox group9,pos={367,226},size={152,83},title="reuslts"
GroupBox group9,labelBack={30464,30464,30464}
    SetVariable setvar14,pos={394,252},size={100,16},limits={-inf,inf,0}, value=resp2, title=" ", format="%g%"
    SetVariable setvar15,pos={394,282},size={100,16},limits={-inf,inf,0}, value=amp2, title=" ", format="%g AU"
CheckBox check1,pos={371,321},size={64,14},title="relative to",value= 0
PopupMenu popup1,pos={445,317},size={50,21},bodyWidth=50
PopupMenu popup1,mode=1,popvalue="1",value= #"1;2\"

Button button4,pos={472,373},size={50,20},title="Edit ", proc= doResults

End

Function Get_3SD(ctrlName) : ButtonControl
string ctrlName

    getmarquee/Z/K/W=graph2 bottom

    if(!V_flag)
        return 0
    endif

variable l1=round(V_left),r1=round(V_right)
wave traces2,traces_x2
    findlevels/Q traces_x2,0
wave W_FindLevels
insertpoints (numpts(W_FindLevels)),1, W_FindLevels
    W_FindLevels[ numpts(W_FindLevels)-1]=numpts(traces_X2)-1

    make/O/N=(V_LevelsFound) temp44=nan

variable p1
    for(p1=0;p1<V_LevelsFound;p1+=1)
        findlevel/Q/R=[W_FindLevels[p1],W_FindLevels[p1+1]-1] traces_x2,l1
variable l2=v_levelX
        findlevel/Q/R=[W_FindLevels[p1],W_FindLevels[p1+1]-1] traces_x2,r1
variable r2=v_levelX
        wavestats/Q/R=[l2,r2] traces2
        temp44[p1]=v_sdev
    endfor
    duplicate/O temp44, W_3SD
    wavestats/Q temp44

nVAR thresh=root:thresh
nVAR tsd
    thresh=tsd*(V_avg)
end

function checkVAR(lasch,var)
string lasch
variable var

    if(stringmatch (lasch,"*1")) ///basal1
NVAR l2=bas1
NVAR r2=bae1
        //          setdrawlayer/K/W=graph2 progback

```

```

//      setdrawenv/W=graph2 linethick=0,fillfgc= (0,5224,5224), xcoord=bottom, ycoord=prel,save

elseif(stringmatch (lasch,"*2")) ///appl1
NVAR l2=as1
NVAR r2=ae1
//      setdrawlayer/W=graph2 progback
//      setdrawenv/W=graph2 linethick=0,fillfgc= (0,5224,5224), xcoord=bottom, ycoord=prel,save
elseif(stringmatch (lasch,"*3")) ///basal2
NVAR l2=bas2
NVAR r2=bae2
//      setdrawlayer/K/W=graph2 userback
//      setdrawenv/W=graph2 linethick=0,fillfgc= (5224,5224,5224), xcoord=bottom, ycoord=prel,save
elseif(stringmatch (lasch,"*4")) ///appl2
NVAR l2=as2
NVAR r2=ae2
//      setdrawlayer/W=graph2 userback
//      setdrawenv/W=graph2 linethick=0,fillfgc= (5224,5224,5224), xcoord=bottom, ycoord=prel,save
elseif(stringmatch (lasch,"*5")) ///basal3
NVAR l2=bas3
NVAR r2=bae3
//      setdrawlayer/K/W=graph2 userback
//      setdrawenv/W=graph2 linethick=0,fillfgc= (5224,5224,5224), xcoord=bottom, ycoord=prel,save
elseif(stringmatch (lasch,"*6")) ///appl3
NVAR l2=as3
NVAR r2=ae3
//      setdrawlayer/W=graph2 userback
//      setdrawenv/W=graph2 linethick=0,fillfgc= (5224,5224,5224), xcoord=bottom, ycoord=prel,save
endif

if(var)
    return r2
else
    return l2
endif
end

```

```

function/S wList(str1,var)
string str1
variable var
string expo=""
variable p1
wave/T lw=root:lw
for(p1=0;p1<var;p1+=1)
    expo+=lw[p1] +",";
endfor
return expo
end

```

```

function top100(str1,v1,v2,v3)
string str1//wave name
variable v1//left
variable v2//right
variable v3//NUMBER OF POINTS

wave wave1=root:anasel:$(str1)
//print waveexists(wave1), nameofwave(wave1),str1

duplicate/O/R=(v1,v2) wave1, temp00

sort/R temp00,temp00

```



```

        wavestats/Q/R=[0,v3] temp00
        killwaves/Z temp00

        return v_avg

end

function analysiT(var)
variable var

wave wA=root:$("Wmax_app"+num2str(var))
wave wB=root:$("Wavg_bas"+num2str(var))

NVAR resp=root:($"resp"+num2str(var))
NVAR amp=root:($"amp"+num2str(var))

        duplicate/O root:($"Wmax_app"+num2str(var)),temp47
        temp47-=wb

        wavestats/Q temp47
        amp=v_avg

wave wA1=root:($"Wavg_app"+num2str(var))
        duplicate/O wA1,temp47

        temp47-=wb

NVAR thresh
        if(thresh==0)
                thresh=0.01
        endif
variable/G tSD
        if(!tSD)
                tSD=3
        endif

wave W_3SD=:W_3SD
        temp47=(temp47[p]>tSD*W_3SD[p]) ///threshold calc
        wavestats/Q temp47
        resp=100*v_sum/v_npnts

end

function doResults(lasch)
string lasch

wave traces2
string n1=note(traces2)
        make/O /N=(itemsinlist(n1,"\r")) W_resp1,W_resp2,W_amp1,W_amp2,w_n,w_ampR1
        make/O/T/N=(itemsinlist(n1,"\r")) w_exp

wave w0=:Wavg_bas1
wave w1=:Wmax_app1
wave w2=:Wavg_bas2
wave w3=:Wmax_app2
wave w5=:W_3SD
wave w6=:Wavg_app1
wave w7=:Wavg_app2

//app1

```

```

duplicate/O w1,temp44
temp44-=w0          ///amp

duplicate/O w6, temp46
temp46-=w0
temp46=(temp46[p]>w5[p])    ///resp

//app2
duplicate/O w3,temp45
temp45-=w2          ///amp2

duplicate/O temp45, temp47
temp47=(temp47[p]>w5[p])    ///resp2

duplicate/O w7,temp48 //rel amp
temp48-=w2
temp48/=temp45
temp48*=100

variable count=0
variable p1=0
    for (p1=0; p1<itemsinlist(n1,"\\r");p1+=1)
string curS=stringfromlist(p1,n1,"\\r")
    w_exp[p1]=stringfromlist(0,curS,":")
variable cur_n=str2num(stringfromlist(1,curS,":"))
    w_n[p1]=cur_N

        duplicate/O/R=[count, count+cur_N] temp44, tempX    ///max.amplitudes app1
        wavestats/Q tempX
        W_amp1[p1]=v_avg

        duplicate/O/R=[count, count+cur_N] temp45, tempX    ///max.amplitudes app2
        wavestats/Q tempX
        W_amp2[p1]=v_avg

        duplicate/O/R=[count, count+cur_N] temp46, tempX    ///n resp app1
        wavestats/Q tempX
        W_resp1[p1]=100*v_sum/v_npnts

        duplicate/O/R=[count, count+cur_N] temp47, tempX    ///n resp app2
        wavestats/Q tempX
        W_resp2[p1]=100*v_sum/v_npnts

        duplicate/O/R=[count, count+cur_N] temp44, tempX    ///max.amplitudes app1
        duplicate/O/R=[count, count+cur_N] temp45, tempX1    ///max.amplitudes app2
        tempX/=tempX1
        tempX*=100
        wavestats/Q tempX
        W_ampR1[p1]=V_avg
        //print p1, count, cur_N
        count+=cur_N

    endfor
killwaves/Z tempX,tempX1,temp44,temp45,temp46,temp47,temp48

edit/K=1 w_exp,w_n,w_resp1,w_amp1,w_ampR1, w_resp2, w_amp2

end

//Window W_analysis() : Panel
PauseUpdate; Silent 1          /// building window...
NewPanel /K=1 /W=(294,239,823,651) as "Analysis"

```

```

ShowTools/A
GroupBox group0,pos={5,44},size={518,147},title="application 1"
GroupBox group0,labelBack=(56576,56576,56576),frame=0
SetVariable setvar0,pos={15,106},size={70,16},title=" "
SetVariable setvar0,limits={-inf,inf,0},value= app1
GroupBox group1,pos={97,68},size={89,80},title="basal"
GroupBox group1,labelBack=(30464,30464,30464)
SetVariable setvar2,pos={106,93},size={32,16},title=" "
SetVariable setvar2,limits={-inf,inf,0},value= bas1
SetVariable setvar3,pos={144,93},size={32,16},title=" "
SetVariable setvar3,limits={-inf,inf,0},value= bae1
SetVariable setvar1,pos={106,123},size={70,16},title=" "
SetVariable setvar1,limits={-inf,inf,0},value= mb1
Button ba1,pos={118,154},size={50,20},proc=getM,title="get basal"
GroupBox group2,pos={187,68},size={86,81},title="appl."
GroupBox group2,labelBack=(30464,30464,30464)
SetVariable setvar4,pos={196,93},size={32,16},title=" "
SetVariable setvar4,limits={-inf,inf,0},value= as1
SetVariable setvar5,pos={234,93},size={32,16},title=" "
SetVariable setvar5,limits={-inf,inf,0},value= ae1
SetVariable setvar6,pos={196,123},size={70,16},title=" "
SetVariable setvar6,limits={-inf,inf,0},value= ma1
Button ba2,pos={205,154},size={50,20},proc=getM,title="get appl."
GroupBox group3,pos={277,68},size={86,81},title="stats"
GroupBox group3,labelBack=(30464,30464,30464)
SetVariable setvar07,pos={285,92},size={70,16},title=" ",format="%g cells"
SetVariable setvar07,limits={-inf,inf,0},value= nc1
SetVariable setvar08,pos={285,122},size={70,16},title=" ",format="%g exp."
SetVariable setvar08,limits={-inf,inf,0},value= ne1
GroupBox group4,pos={367,66},size={152,83},title="reuslts"
GroupBox group4,labelBack=(30464,30464,30464)
SetVariable setvar05,pos={394,92},size={100,16},title=" ",format="%g%"
SetVariable setvar05,limits={-inf,inf,0},value= resp1
SetVariable setvar06,pos={394,122},size={100,16},title=" ",format="%g AU"
SetVariable setvar06,limits={-inf,inf,0},value= amp1
CheckBox check0,pos={371,161},size={64,14},title="relative to",value= 0
PopupMenu popup0,pos={445,157},size={50,21},bodyWidth=50
PopupMenu popup0,mode=1,popvalue="1",value= #"1;2\"
GroupBox group5,pos={5,204},size={518,147},title="application 2"
GroupBox group5,labelBack=(56576,56576,56576),frame=0
SetVariable setvar7,pos={15,266},size={70,16},title=" "
SetVariable setvar7,limits={-inf,inf,0},value= app2
GroupBox group6,pos={97,228},size={89,80},title="basal"
GroupBox group6,labelBack=(30464,30464,30464)
SetVariable setvar8,pos={106,253},size={32,16},title=" "
SetVariable setvar8,limits={-inf,inf,0},value= bas2
SetVariable setvar9,pos={144,253},size={32,16},title=" "
SetVariable setvar9,limits={-inf,inf,0},value= bae2
SetVariable setvar02,pos={106,283},size={70,16},title=" "
SetVariable setvar02,limits={-inf,inf,0},value= mb2
Button ba3,pos={118,314},size={50,20},proc=getM,title="get basal"
GroupBox group7,pos={187,228},size={86,81},title="appl."
GroupBox group7,labelBack=(30464,30464,30464)
SetVariable setvar09,pos={196,253},size={32,16},title=" "
SetVariable setvar09,limits={-inf,inf,0},value= as2
SetVariable setvar10,pos={234,253},size={32,16},title=" "
SetVariable setvar10,limits={-inf,inf,0},value= ae2
SetVariable setvar11,pos={196,283},size={70,16},title=" "
SetVariable setvar11,limits={-inf,inf,0},value= ma2
Button ba4,pos={205,314},size={50,20},proc=getM,title="get appl."
GroupBox group8,pos={277,228},size={86,81},title="stats"
GroupBox group8,labelBack=(30464,30464,30464)
SetVariable setvar12,pos={285,252},size={70,16},title=" ",format="%g cells"
SetVariable setvar12,limits={-inf,inf,0},value= nc2
SetVariable setvar13,pos={285,282},size={70,16},title=" ",format="%g exp."
SetVariable setvar13,limits={-inf,inf,0},value= ne2

```

```

GroupBox group9,pos={367,226},size={152,83},title="reuslts"
GroupBox group9,labelBack={30464,30464,30464}
SetVariable setvar14,pos={394,252},size={100,16},title=" ",format="%g%"
SetVariable setvar14,limits={-inf,inf,0},value= resp2
SetVariable setvar15,pos={394,282},size={100,16},title=" ",format="%g AU"
SetVariable setvar15,limits={-inf,inf,0},value= amp2
CheckBox check1,pos={371,321},size={64,14},title="relative to",value= 0
PopupMenu popup1,pos={445,317},size={50,21},bodyWidth=50
PopupMenu popup1,mode=1,popvalue="1",value= #"1;2\"
Button button4,pos={472,373},size={50,20},proc=doResults,title="Edit "
SetVariable setvar16,pos={431,20},size={60,16},bodyWidth=60,title=" "
SetVariable setvar16,limits={-inf,inf,0},value= thresh
Button ba5,pos={373,18},size={50,20},proc=getM,title="get 3*SD"
EndMacro

```

```

#pragma rtGlobals=3          // Use modern global access method and strict wave access.

```

```

function nexlas(lasch):ButtonControl
string lasch
    controlinfo/W=canasel popup0
string e1=s_value

string wl=getwl()+getwl()+getwl()
variable num=whichlistitem(e1,wl,";",itemsinlist(getwl()))
variable dest
string new=""
    if(stringmatch(lasch,"*3")) //forw
        new=stringfromlist(num+1,wl,";")
    endif
    if(stringmatch(lasch,"*2")) //rev
        new=stringfromlist(num-1,wl,";")
    endif

    dest=whichlistitem(new,getwl(),";")

    //print dest,num, new
    //print stringfromlist(num,wl,";")
    //print stringfromlist(num+1,wl,";")
    //print stringfromlist(num-1,wl,";")
    popupmenu popup0, win=CanaSel, mode=dest+1, popvalue=new
    PopMenuProc_1("",dest,new )
end

```

```

function getvaluesfromdf(str, lasch)
String str//datafolder
string lasch

if((stringmatch(lasch,"button6"))|| (stringmatch(lasch,"all")))//properties

```

```

string para="S_mouse;S_age;s_sex;s_region;s_geno"
variable p1
    for(p1=0; p1<itemsinlist(para,";");p1+=1)
SVAR/Z curP=root:$(str):$(stringfromlist(p1,para,";"))
        if(!SVAR_exists ( curP ))
            setdatafolder root:$(str):
string/G $(stringfromlist(p1,para,";"))=" "
            setdatafolder root:
        endif

```

```

        endfor
SVAR s_mouse=root:${str}:S_mouse
SVAR S_age=root:${str}:S_age
SVAR S_Sex=root:${str}:S_Sex
SVAR S_region=root:${str}:S_region
SVAR s_genotype=root:${str}:s_genotype
NVAR N_mcthr=root:${str}:mcthr

SVAR mouse=root:S_mouse
SVAR age=root:S_age
SVAR Sex=root:S_Sex
SVAR region=root:S_region
SVAR geno=root:s_genotype
NVAR mcthr=root::mcthr

        mouse=S_mouse
        age=S_age
        Sex=S_Sex
        region=S_region
        geno=s_genotype
        mcthr=N_mcthr
endif

if((stringmatch(lasch,"button7"))|| (stringmatch(lasch,"all")))
//application
SVAR s1=root:${str}:s1
SVAR s2=root:${str}:s2
SVAR s3=root:${str}:s3
SVAR s4=root:${str}:s4
SVAR s5=root:${str}:s5
SVAR s6=root:${str}:s6
SVAR e1=root:${str}:e1
SVAR e2=root:${str}:e2
SVAR e3=root:${str}:e3
SVAR e4=root:${str}:e4
SVAR e5=root:${str}:e5
SVAR e6=root:${str}:e6
SVAR w1=root:${str}:w1
SVAR w2=root:${str}:w2
SVAR w3=root:${str}:w3
SVAR w4=root:${str}:w4
SVAR w5=root:${str}:w5
SVAR w6=root:${str}:w6

        if(!SVAR_exists(s1))
                setdatafolder root:${str}:

String/G s1=""
String/G s2=""
String/G s3=""
String/G s4=""
String/G s5=""
String/G s6=""
String/G e1=""
String/G e2=""
String/G e3=""
String/G e4=""
String/G e5=""
String/G e6=""
String/G w1=""
String/G w2=""
String/G w3=""
String/G w4=""
String/G w5=""

```

```
String/G w6=""
```

```
setdatafolder root:
```

```
endif
```

```
SVAR s1=root:$(str):s1
SVAR s2=root:$(str):s2
SVAR s3=root:$(str):s3
SVAR s4=root:$(str):s4
SVAR s5=root:$(str):s5
SVAR s6=root:$(str):s6
SVAR e1=root:$(str):e1
SVAR e2=root:$(str):e2
SVAR e3=root:$(str):e3
SVAR e4=root:$(str):e4
SVAR e5=root:$(str):e5
SVAR e6=root:$(str):e6
SVAR w1=root:$(str):w1
SVAR w2=root:$(str):w2
SVAR w3=root:$(str):w3
SVAR w4=root:$(str):w4
SVAR w5=root:$(str):w5
SVAR w6=root:$(str):w6
```

```
SVAR s_1=root:s1
SVAR s_2=root:s2
SVAR s_3=root:s3
SVAR s_4=root:s4
SVAR s_5=root:s5
SVAR s_6=root:s6
SVAR e_1=root:e1
SVAR e_2=root:e2
SVAR e_3=root:e3
SVAR e_4=root:e4
SVAR e_5=root:e5
SVAR e_6=root:e6
SVAR w_1=root:w1
SVAR w_2=root:w2
SVAR w_3=root:w3
SVAR w_4=root:w4
SVAR w_5=root:w5
SVAR w_6=root:w6
```

```
s_1=s1
s_2=s2
s_3=s3
s_4=s4
s_5=s5
s_6=s6
e_1=e1
e_2=e2
e_3=e3
e_4=e4
e_5=e5
e_6=e6
w_1=w1
w_2=w2
w_3=w3
w_4=w4
w_5=w5
w_6=w6
```

```
endif
```

```
end
```

Function SetVarProc_1(ctrlName,varNum,varStr,varName) : SetVariableControl

```

        String ctrlName
        Variable varNum
        String varStr
        String varName

        controlinfo/W=canasel popup0
string str=s_value

```

```

SVAR s_mouse=root:$(str):S_mouse
SVAR S_age=root:$(str):S_age
SVAR S_Sex=root:$(str):S_Sex
SVAR S_region=root:$(str):S_region
SVAR S_geno=root:$(str):S_geno

```

```

SVAR mouse=root:S_mouse
SVAR age=root:S_age
SVAR Sex=root:S_Sex
SVAR region=root:S_region
SVAR geno=root:S_geno

```

```

        S_mouse=mouse
        S_age=age
        S_Sex=Sex
        S_region=region
        S_geno=geno
        appendinfo2exp()
End

```

```

Function GetTracesFromDF()

```

```

SVAR picN=root:picN1
        make/o/N=0 curTrace
wave/T st=root:$(picN):roiMEAN
        if(!waveexists(st))
//                print "?"
                                return 0
        endif
//                print "!"
        rmvAllTraces("Canasel")
        killdatafolder/Z root:curanasel

        controlinfo/W=canaSel popup0
string e1=S_value
        e1=replacestring(" ",e1,"_")
        e1=        replacestring(".tif",e1,"")
        newdatafolder/O root:curAnaSel

variable p1
        for(p1=0;p1<numpts(st);p1+=1)
                                getcurTrace1(p1)
wave curtrace
                                duplicate/O root:curtrace, root:curAnaSel:$(e1+"_"+num2str(p1))
        endfor
        displayTfromF("curAnasel", "Canasel")

end

```

```

function displayTfromF(df,win)
string df
string win

        setdatafolder root:$(df)
string wl=wavelist("*",";","")//getdatafolder
        setdatafolder root:

variable p1
        for(p1=0;p1<itemsinlist(wl,";");p1+=1)
                appendtograph/W=$(win) root:$(df):$(stringfromlist(p1,wl,";"))
        endfor

        //      add INFO!!
end

```

```

function rmvAllTraces(win)
string win

//      dowindow $(win)
//if(!V_flag)
//return 0
//endif

getwindow $(win), wavelist
wave/T wl=W_WaveList

variable p1

        for(p1=0;p1<dimsize(wl,0);p1+=1)
                removefromgraph/W=$(win)/Z $(wl[p1][0])
        endfor

end

```

```

function globals()

make/O/N=1/T lw
make/O/N=1 selw
newdatafolder/O root:anase1
newdatafolder/O root:analysis

variable/G xOff
variable/G MCthr=20
String/G app1=""
variable /Gbas1,bae1,mb1
variable /G as1,ae1,ma1
variable/G nc1,ne1
variable/G resp1,amp1
variable /G factor=3
String/G app2=""
variable /Gbas2,bae2,mb2
variable /G as2,ae2,ma2
variable/G nc2,ne2
variable/G resp2,amp2
variable/G smo44=1, correction=1
variable /G range11=60,range12=60,range21=60,range22=60
end

```



```

function getcurTrace1(num)
variable num
    //("getcurTrace("+num2str(num)+")")

SVAR picn=root:picN1
wave/T wm=root:$(picN):ROIbMean
wave/T bm=root:$(picN):ROIbMean
string data=wm[num]
string bdata=bm[num]

    make/O/N=(itemsinlist(data,";")) curtrace=str2num(stringfromlist(p,data,";"))
    make/O/N=(itemsinlist(bdata,";")) curbtrace=str2num(stringfromlist(p,bdata,";"))

    controlinfo/W=CanaSel bckgr4//roi
    if(v_value)
        controlinfo/W=CanaSel bckgr3//background sub
        if(v_value)
            curtrace-=curbtrace
        endif
    else
        curtrace=curbtrace
    endif

    controlinfo/W=CanaSel bckgr1//normalization
    if(v_value)
variable n1=mean(curtrace,0,10)
        curtrace/=n1
    endif
end

```

```

Function CheckProcBCKGR_1(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked
        controlinfo/W=CanaSel popup0
        PopMenuProc_1("",v_value,s_value)

End

```

```

Function SetVarProc(ctrlName,varNum,varStr,varName) : SetVariableControl
    String ctrlName
    Variable varNum
    String varStr
    String varName

```

```

wave/T lw=root:lw
wave selw=root:selW

```

```

variable p1
    for(p1=0;p1<numpts(lw); p1+=1)

```

```

            if(selw[p1])
wave curW=root:anasel:${lw[p1]}
                setscale/p x, (varNum),1, curW
            endif
        endfor
    end

```

End

```

function markT()
string seL=getseL()
variable p1
string t1=tracenamelist("CANA#ME",",",1)
string lasT=stringfromlist(itemsinlist(t1)-1,t1,";");//itemsinlist(t1)-1

    for (p1=0; p1<itemsinlist(seL,",");p1+=1)
        ModifyGraph/W=CANA#ME lsize($(stringfromlist(p1,seL,","))=2, rgb($(stringfromlist(p1,seL,",")))=(0,0,0)
        if(!stringmatch(lasT,""+stringfromlist(p1,seL,",")+""))
            reordertraces/W=CANA#ME $(lasT), {$(lasT), $(stringfromlist(p1,seL,","))}
            //print lasT,stringfromlist(p1,seL,",")
        endif
    endfor
end

```

```

function/S getseL()
wave/T lw
wave selw
string expo=""
variable p1
    for (p1=0; p1<numelems(selw);p1+=1)
        if(selw[p1])
            expo+=lw[p1]+","
        endif
    endfor

    return expo
end

```

```

function getM(lasch):buttonControl
string lasch

checkLW()
wave/T lw=root:lw
    if(!numelems(lw))
        return 0
    endif
string fil=""

    if(stringmatch (lasch,"*1"))
        controlinfo fil1
        if(v_value)//filtered
            fil="Y"
        endif
    endif

```

```

endif

if(stringmatch (lasch,"*2"))
    getm("ba1")//first the basal
    controlinfo fil1
    if(v_value)//filtered
        fil="Y"
    endif

endif

if(stringmatch (lasch,"*3"))
    controlinfo fil2
    if(v_value)//filtered
        fil="Y"
    endif

endif

if(stringmatch (lasch,"*4"))
    getm("ba3")//first the basal
    controlinfo fil2
    if(v_value)//filtered
        fil="Y"
    endif

endif

if(stringmatch (lasch,"*5"))
    controlinfo fil3
    if(v_value)//filtered
        fil="Y"
    endif

endif

if(stringmatch (lasch,"*6"))
    getm("ba5")//first the basal
    controlinfo fil3
    if(v_value)//filtered
        fil="Y"
    endif

endif

variable l1,r1
getmarquee/Z/K/W=CANA#ME bottom
if(V_flag)
    l1=v_left
    r1=v_right
else
    l1=checkVAR(lasch,0)
    r1=checkVAR(lasch,1)
endif

if(!(l1-r1))
    //print "NO"
    return 0
endif

calclT(l1,r1,fil)           // generating temp waves

drawIT(l1,r1,lasch)    //...and store it

analT()                  //updating all analysis

//killwaves/Z fw1

end

```

```

function analT()
    setdatafolder root:analysis
wave bas1_top10
wave bas1_max
wave bas1_min
wave bas1_avg
wave bas1_sdev
wave bas2_top10
wave bas2_max
wave bas2_min
wave bas2_avg
wave bas2_sdev
wave bas3_top10
wave bas3_max
wave bas3_min
wave bas3_avg
wave bas3_sdev

wave app1_top10
wave app1_max
wave app1_min
wave app1_avg
wave app1_sdev
wave app2_top10
wave app2_max
wave app2_min
wave app2_avg
wave app2_sdev
wave app3_top10
wave app3_max
wave app3_min
wave app3_avg
wave app3_sdev
NVAR factor=root:factor

    if(waveexists(app1_top10))
        duplicate/O app1_top10, app1_resp
        if(numpts(app1_top10)==numpts(bas1_top10))
            app1_resp=(bas1_sdev*factor+bas1_avg<app1_top10) //responding cell?
        endif
wave fw1=root:fw1

        app1_resp*=fw1

NVAR nc1=root:nc1 //number of cells
    wavestats/Q app1_max
    nc1=V_npnts
NVAR resp1=root:resp1//responding cells
    wavestats/Q app1_resp
    resp1=100*v_avg
NVAR ne1=root:ne1//number of experimnets
    ne1=countexp()
NVAR amp1=root:amp1//amplitude
    wavestats/Q app1_top10
    amp1=v_avg
    endif

    if(waveexists(app2_top10))
        duplicate/O app2_top10, app2_resp
        if(numpts(app2_top10)==numpts(bas2_top10))
            app2_resp=(bas2_sdev*factor+bas2_avg<app2_top10)
        endif
NVAR nc2=root:nc2

```

```

                nc2=umpnts(app2_max)
NVAR resp2=root:resp2
                wavstats/Q app2_resp
                resp2=100*v_avg
NVAR ne2=root:ne2
                ne2=countexp()
NVAR amp2=root:amp2
                amp2=mean(app2_top10)
        endif

        if(waveexists(app3_top10))
                duplicate/O app3_top10, app3_resp
                if(umpnts(app3_top10)=umpnts(bas3_top10))
                        app3_resp=(bas3_sdev*factor+bas3_avg<app3_top10)
                endif
NVAR nc3=root:nc3
                nc3=umpnts(app3_max)
NVAR resp3=root:resp3
                wavstats/Q app3_resp
                resp3=100*v_avg
NVAR ne3=root:ne3
                ne3=countexp()
NVAR amp3=root:amp3
                amp3=mean(app3_top10)
        endif
        setdatafolder root:
end

```

```

Function ButtonProc_3(ctrlName) : ButtonControl
        String ctrlName
variable a1=0
        if(stringmatch(ctrlName,"sel*"))//show responding
                a1=1
        endif

variable n1=2
        if(stringmatch(ctrlName,"*1"))
                n1=1
        elseif(stringmatch(ctrlName,"*2"))
                n1=2
        elseif(stringmatch(ctrlName,"*3"))
                n1=3
        endif

        wave selw=root:selW
        wave curw=root:analysis:$("app"+num2str(n1)+"_resp")

                selw=(curW==a1)

                ModifyGraph/W=CANA#ME lsize=1, rgb=(65000,0,0)
                markT()

```

End

m

```

function countexp()
wave/T n1=root:lw
wave ch1=root:fw1
        if(!waveexists(ch1))
                make/O/N=(umpnts(n1)) root:qwertz=1
wave ch1=root:qwertz
        endif

```

```

variable expo
string rem=""
variable p1
    for(p1=0;p1<numpts(n1);p1+=1)
        if(ch1[p1])
            wave curW=root:anasel:$(n1[p1])
            string e1=stringfromlist(0,note(curw),";");//str1[0,strsearch("190304_cc_thalamus_2__3","__",0)]
            e1=replacestring(" ",e1,"_")
            e1=        replacestring(".tif",e1,"__")

                if(findlistitem(e1,rem,";")<0)
                    rem+=e1+";"
                    expo+=1
                endif
            endif
        endif
    endfor

return expo

end

```

```

function/S getSelected()
wave selW=root:selw
wave/T Lw=root:lw

variable p1
string expo=""
    for(p1=0;p1<numpts(lw);p1+=1)
        if(selw[p1])
            expo+=lw[p1]+";"
        endif
    endfor

return expo

end

```

```

Function PopMenuProc_2(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum
    String popStr

    if(stringmatch(ctrlName,"*1"))///sex
        SVAR tar1=root:s_sex
            tar1=popStr
            SetVarProc_1("setvarA21",1,popstr,"s_sex")
        elseif(stringmatch(ctrlName,"*2"))///region
            SVAR tar1=root:s_region
                tar1=popStr
                SetVarProc_1("setvarA22",1,popstr,"s_region")
            elseif(stringmatch(ctrlName,"*3"))///genotype
                SVAR tar1=root:s_geno

```

```

        tar1=popStr
        SetVarProc_1("setvarA23",1,popstr,"s_geno")
    endif

End

function/S getreg()

string expo="Cortex;Corpus Callosum;Hippocampus;Striatum;Thalamus;Cerebellum"
    return expo

end

Function PopMenuProc_3enterAPP(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum
    String popStr

SVAR curW=root:${"W"+ctrlName[3]}
    curW=popstr
    SetVarProc_Appli1("setvarA"+num2str(12+str2num(ctrlName[3])),0,curW,"W"+ctrlName[3])

End

Function CheckProcBCKGR_2(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked
    ///bckgr4=ROI;bckgr2=back
        if(stringmatch(ctrlName,"*4"))//ROI
            checkbox bckgr2, win=canasel, value=0
            CheckBox bckgr3, win=canasel, disable=0
        endif

        if(stringmatch(ctrlName,"*2"))//back
            checkbox bckgr4, win=canasel, value=0
            CheckBox bckgr3, win=canasel, disable=2
        endif

        controlinfo/W=CanaSel popup0
        PopMenuProc_1("",v_value,s_value)

End

Function SetVarProc_Appli1(ctrlName,varNum,varStr,varName) : SetVariableControl
    String ctrlName
    Variable varNum
    String varStr
    String varName

    make/O/N=18 appli=nan, appli_x=nan
SVAR pn=root:picN1
variable p1=0, isit=0

do

```

```

SVAR curS=root:${"s"+num2str(p1/3+1))
SVAR curE=root:${"e"+num2str(p1/3+1))
    if( (str2num(curS)) &&str2num((curE)))
        isit=1
    endif
    appli[p1]=p1
    appli[p1+1]=p1

    appli_x[p1]=str2num(curS)
    appli_x[p1+1]=str2num(curE)
    appli_x[p1+2]=NAN

    p1+=3
    while(p1<18)

        if( isit)
            Apply2Graph1()
        endif

        controlinfo/W=canasel popup0
    string str=s_value

```

```

SVAR s1=root:${str}:s1
SVAR s2=root:${str}:s2
SVAR s3=root:${str}:s3
SVAR s4=root:${str}:s4
SVAR s5=root:${str}:s5
SVAR s6=root:${str}:s6
SVAR e1=root:${str}:e1
SVAR e2=root:${str}:e2
SVAR e3=root:${str}:e3
SVAR e4=root:${str}:e4
SVAR e5=root:${str}:e5
SVAR e6=root:${str}:e6
SVAR w1=root:${str}:w1
SVAR w2=root:${str}:w2
SVAR w3=root:${str}:w3
SVAR w4=root:${str}:w4
SVAR w5=root:${str}:w5
SVAR w6=root:${str}:w6

```

```

SVAR s_1=root:s1
SVAR s_2=root:s2
SVAR s_3=root:s3
SVAR s_4=root:s4
SVAR s_5=root:s5
SVAR s_6=root:s6
SVAR e_1=root:e1
SVAR e_2=root:e2
SVAR e_3=root:e3
SVAR e_4=root:e4
SVAR e_5=root:e5
SVAR e_6=root:e6
SVAR w_1=root:w1
SVAR w_2=root:w2
SVAR w_3=root:w3
SVAR w_4=root:w4
SVAR w_5=root:w5
SVAR w_6=root:w6

```

```

s1=s_1
s2=s_2
s3=s_3
s4=s_4
s5=s_5
s6=s_6
e1=e_1

```



```

e2=e_2
e3=e_3
e4=e_4
e5=e_5
e6=e_6
w1=w_1
w2=w_2
w3=w_3
w4=w_4
w5=w_5
w6=w_6

appendinfo2exp()
return 0

End

function Apply2Graph1()

//bars
wave appli, appli_X
if(strlen(listmatch(wavelist("*",",","WIN:canasel"), "appli")))
    if(waveExists(appli))
        AppendToGraph/W=canasel/L=IA appli vs appli_x
        ModifyGraph/W=canasel noLabel(IA)=2,axThick(IA)=0,axisEnab(left)={0,0.8};DelayUpdate
        ModifyGraph/W=canasel axisEnab(IA)={0.9,1}, rgb(appli)=(0,0,0),lsize(appli)=2
        ModifyGraph/W=canasel axisEnab(left)={0,0.8}
        setaxis/W=canasel IA 0,20
        setdrawlayer/K/W=canasel userback
        setdrawenv/W=canasel fsize=8, xcoord=bottom, ycoord=IA, save
    endif
else
    setdrawlayer/K/W=canasel userback
    setdrawenv/W=canasel fsize=8, xcoord=bottom, ycoord=IA, save
endif

//text above app bars
Variable p1=0
do
    SVAR appl=root:$("w"+num2str(p1+1))
    SVAR sA= root:$("s"+num2str(p1+1))

    drawtext/W=canasel str2num(sA), p1*3, appl
    p1+=1
    while(p1<6)

end

//
wave selw
wave/T lw

    findvalue/TEXT=(stringfromlist(p1,wl,",")) lw
    //print v_value,(stringfromlist(p1,wl,","))
    if(v_value>=0)
        deletepoints (v_value),1, selw,lw
        if(stringmatch(ctrlName,"FILTER"))
            wave fw1=root:analysis:filterW
            deletepoints (v_value),1, fw1
        endif
    endif

Function CheckProc_1(ctrlName,checked) : CheckBoxControl

```

```

String ctrlName
Variable checked
variable roi

    if(stringmatch(ctrlName,"*4"))//ROI
        checkbox bckgr2, win=cana, value=0
        CheckBox bckgr3, win=cana, disable=0
        roi=1
    endif

        if(stringmatch(ctrlName,"*2"))//back
            checkbox bckgr4, win=cana, value=0
            CheckBox bckgr3, win=cana, disable=2
            roi=0
        endif
    if(stringmatch(ctrlName,"*3"))//sub
        roi=1
    endif
    if(stringmatch(ctrlName,"*1"))//norm
        controlinfo/W=cana bckgr4//roi
        if(v_value)
            CheckProc_1("bckgr4",1)
        else
            CheckProc_1("bckgr2",1)
        endif
        return 0
    endif
wave/T lw=root:lw
variable numw=numpts(lw)

variable p1
    for(p1=0;p1<numw;p1+=1)

wave curw=root:anel:${lw[p1]}

string df=stringfromlist(0,note(curW),";")
variable numtrace=str2num(stringfromlist(1,note(curW),";"))

wave/T bm=root:${df}:ROIbMean
wave/T wm=root:${df}:ROIbMean
string data=wm[numtrace]
string bdata=bm[numtrace]
        redimension/N=(itemsinlist(bdata,";")) root:anel:${lw[p1]}
        if(!roi) ///background
            curw=str2num(stringfromlist(p,bdata,";"))
        else ///ROI
            controlinfo/W=cana bckgr3
            if(v_value)//subtract
                curw=str2num(stringfromlist(p,data,";"))-str2num(stringfromlist(p,bdata,";"))
            else
                curw=str2num(stringfromlist(p,data,";"))
            endif
        endif
        controlinfo/W=cana bckgr1
        if(v_value)//normalize
variable n1=mean(curw,0,10)
            curw/=n1
        endif

        doAction( StringByKey("action",note(curW),":",";"),nameofwave(curW))
    endfor

    updateCANAME()
    getm("ba2")
    getm("ba4")

```

End

```
Function ButtonProc_5reversel(ctrlName) : ButtonControl
    String ctrlName
    wave selw
    selw=(selw==0)
End
```

```
Function ButtonProc_5(ctrlName) : ButtonControl
    String ctrlName
    wave selw
    selw=1
End
```

```
function/S getselAPPL()
string expo=""
wave/T lw=root:lw
wave selw

string fl=""

variable p1
    for(p1=0;p1<numpnts(selw);p1+=1)
        if(selw[p1])
            wave curw=root:anasel:$(lw[p1])
            string df=stringfromlist(0,note(curW),";")
                if(!strlen(listmatch(fl,df,";")))
                    fl+=df+";"
                endif
            endif
        endif
    endfor
    //fl=replacestring(" ",fl,"_")
    print fl
variable q1
    for(p1=0;p1<itemsinlist(fl,";");p1+=1)
        for(q1=1;q1<7;q1+=1)
            SVAR curapp=root:$(stringfromlist(p1,fl,";")):$("w"+num2str(q1))
                if(strlen(curapp))
                    if(!strlen(listmatch(curapp,expo,";")))
                        expo+=curapp+";"
                    endif
                endif
            endif
        endfor
    endfor

    return sortlist(expo)
end
```

```
function/S getappl(fl)
string fl //folderlist
variable p1,q1
string expo=""
    for(p1=0;p1<itemsinlist(fl,";");p1+=1)
        for(q1=1;q1<7;q1+=1)
            SVAR curapp=root:$(stringfromlist(p1,fl,";")):$("w"+num2str(q1))
                if(strlen(curapp))
                    if(!stringmatch(expo,"*" + curapp + "*"))
                        expo+=curapp+";"
                    endif
                endif
            endif
        endfor
    endfor
```

```

                endif
            endfor
        endfor

        return expo
    end
end

```

```

Function PopMenuProc_3byAPP(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum
    String popStr

```

```

    wave/T lw=root:lw
    wave selw=root:selw
    variable rem
    variable p1
        for(p1=0;p1<numpts(selw);p1+=1)
            if(selw[p1])
                wave curW=root:anase1:$(lw[p1])
                variable start=findappstart(stringfromlist(0,note(curW),","), popstr)
                setscale/p x, (-1*start),1, curW
                if (start>rem)
                    rem=start
                endif
            endif
        endfor

        for(p1=0;p1<numpts(selw);p1+=1)
            if(selw[p1])
                wave curW=root:anase1:$(lw[p1])
                setscale/p x, (leftx(curW)+rem),1, curW
            endif
        endfor
    end

```

End

```

function findappstart(str1,str2)
    string str1//data folder
    string str2//application

    variable q1
        for(q1=1;q1<7;q1+=1)
            svar app=root:$(str1):$("w"+num2str(q1))
            svar appt=root:$(str1):$("s"+num2str(q1))
            if(stringmatch(app, str2))
                return str2num(appt)
            endif
        endfor
    end

```

```

Function PopMenuProc_3setGeno(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum
    String popStr

```

```

    SVAR gen=root:s_geno
    gen=popstr

```

End

Function CheckProc_2Filter(ctrlName,checked) : CheckBoxControl

String ctrlName
Variable checked

if(!checked)
 dowindow/K w_filter
 backuptraces(0)
 checkLW()
 killdatafolder/Z root:analysis:alltraces
 return 0
endif

backuptraces(1)
prepfiler()
Button button1,win=W_filter, pos={535,630},size={50,50},proc=ButtonProc_applyfilter,title="Do it"

end

function prepfiler()

if(!datafolderexists("root:analysis:filter"))
 newdatafolder root:analysis:filter
endif

SVAR/Z filters=root:analysis:filters

///always generate
 setdatafolder root:analysis:
string/G filters="age:0:0;0\r"
 filters+="sex:0:3\r"
 filters+="region:0:0:\r"
 filters+="geno:0:\r"
 filters+="app:0:\r"
 filters+="loc:0:;;0\r"
 filters+="locA:0:;;0\r"
 setdatafolder root:

NVAR/Z F_age1=root:analysis:F_age1

if(! NVAR_exists (F_age1))
 setdatafolder root:analysis:

variable/G F_age1
 setdatafolder root:
endif

NVAR/Z F_age2=root:analysis:F_age2

if(! NVAR_exists (F_age2))
 setdatafolder root:analysis:

variable/G F_age2
 setdatafolder root:
endif

NVAR/Z F_dist1=root:analysis:F_dist1

if(! NVAR_exists (F_dist1))
 setdatafolder root:analysis:

variable/G F_dist1
 setdatafolder root:
endif

NVAR/Z F_dist2=root:analysis:F_dist2

if(! NVAR_exists (F_dist2))
 setdatafolder root:analysis:

variable/G F_dist2
 setdatafolder root:

```

        endif
NVAR/Z F_distA=root:analysis:F_distA
        if(! NVAR_exists (F_distA) )
            setdatafolder root:analysis:
variable/G F_distA=250
            setdatafolder root:
        endif
NVAR/Z F_distF=root:analysis:F_distF
        if(! NVAR_exists (F_distF) )
            setdatafolder root:analysis:
variable/G F_distF=0.625
            setdatafolder root:
        endif
SVAR/Z F_points=root:analysis:F_points
        if(! SVAR_exists (F_points) )
            setdatafolder root:analysis:
string/G F_points
            setdatafolder root:
        endif
        checkLW()
        fW_filter()
        //applyfilters()

```

End

```

function makelbw()
    //list box waves
//region
string reg=getreg()
        make/O/T/N=(itemsinlist(reg)) root:analysis:reg_listw
wave/T lreg=root:analysis:reg_listw
        lreg=stringfromlist(p,reg,";")
        make/O/N=(itemsinlist(reg)) root:analysis:reg_selw
wave sreg=root:analysis:reg_selw
        sreg=0

//geno
string geno=getgeno()
        make/O/T/N=(itemsinlist(geno)) root:analysis:geno_listw
wave/T lgeno=root:analysis:geno_listw
        lgeno=stringfromlist(p,geno,";")
        make/O/N=(itemsinlist(geno)) root:analysis:geno_selw
wave sgeno=root:analysis:geno_selw
        sgeno=0

//application
string app=getapp()
        make/O/T/N=(itemsinlist(app)) root:analysis:app_listw
wave/T lapp=root:analysis:app_listw
        lapp=stringfromlist(p,app,";")
        make/O/N=(itemsinlist(app)) root:analysis:app_selw
wave sapp=root:analysis:app_selw
        sapp=0

end

```

```

function/S getsetLF()
wave/T lw=root:lw
wave selw=root:analysis:filterW

string expo=""
variable p1
    for (p1=0; p1<numpnts(selw);p1+=1)
        if(selw[p1])
            expo+=lw[p1]+";"
        endif
    endfor

    return expo
end

```

```

function/S getfiltersetl(str)
string str ///which parameter
wave/T curl=root:analysis:$(str+"_ListW")
wave curS=root:analysis:$(str+"_SelW")
string expo=""

variable p1
    for(p1=0;p1<numpnts(curl);p1+=1)
        if(curS[p1])
            expo+=curl[p1]+";"
        endif
    endfor

    return expo
end

```

```

function backuptraces(var)
variable var

    if(var)//backup traces
        newdatafolder/O root:analysis:allTraces
    variable p1
        for(p1=0;p1<countobjects("root:anaset",1); p1+=1)
            wave curW=root:anaset:$(getindexedobjName("root:anaset",1,p1))
            duplicate/O curW, root:analysis:allTraces:$(nameofwave(curW))
        endfor
    else
        //restore all traces
        for(p1=0;p1<countobjects("root:analysis:allTraces:",1); p1+=1)
            wave curW=root:analysis:allTraces:$(getindexedobjName("root:analysis:allTraces:",1,p1))
            duplicate/O curW, root:anaset:$(nameofwave(curW))
        endfor
    end

```

```

endif

checklw()

end

function updateCANAME()
    make/O/T/N=0 root:lw
    make/O/N=0 root:selw
    rmvalltraces("cana#me")

    variable numw=countobjects("root:anasel",1)
    variable p1
    for(p1=0;p1<numw;p1+=1)//3;p1+=1)//
    wave curW=root:anasel:$(getindexedobjName("root:anasel",1,p1))
        app2ME(curW,"",1)
    endfor
    wave/T lw=root:lw
    checkLW()
    if(numpts(lw))
        getm("ba2")
        getm("ba4")
    endif
end

```

```

Function ButtonProc_1(ctrlName) : ButtonControl
    String ctrlName

```

```

SVAR picN=root:picN1
    make/o/N=0 curTrace
    wave/T st=root:$(picN):roiMEAN
    if(!waveexists(st))
        return 0
    endif
    controlinfo/W=canaSel popup0
    string e1=S_value
    e1=replacestring(" ",e1,"_")
    e1=    replacestring(".tif",e1,"")
    if(!datafolderExists("AnaSel"))
        newdatafolder/O root:AnaSel
    endif
    //print e1
    variable p1
    for(p1=0;p1<numpts(st);p1+=1)
        getcurTrace1(p1)
    wave curtrace
        duplicate/O root:curtrace, root:AnaSel:$(e1+"__"+num2str(p1))
    wave cw=root:AnaSel:$(e1+"__"+num2str(p1))
        note cw,picN    +";"+num2str(p1)
        appendInfo(nameofwave(cw))
        app2ME(root:AnaSel:$(e1+"__"+num2str(p1)),e1,p1)
        if(datafolderexists("root:analysis:allTraces:"))
            duplicate/O root:AnaSel:$(e1+"__"+num2str(p1)), root:analysis:allTraces:$(e1+"__"+num2str(p1))
        else
            backuptraces(1)
        endif
    endfor

    setdatafolder root:anasel:

```



```

string wl=wavelist("*",",","")
SETDATAFOLDER root:
make/O/N=(itemsinlist(wl,",")) root:fw1=1
end

function makeFW(wl)
string wl
make/O/N=(itemsinlist(wl,",")) root:fw1=1
wave fw1 =root:fw1

variable p1
for(p1=0;p1<numpnts(fw1);p1+=1)
    if( !filteredYES(stringfromlist(p1,wl,",")))
        fw1[p1]=nan
    endif
endfor

end

```

```

function appendInfo(waveN)
string waveN
wave curW=root:anasel:$(waveN)

```

```

string n1=note(curw)
string df=stringfromlist(0,n1,",")

```

```

string i1="age:"
SVAR age=root:$(df):S_age
i1+=age+"\r"

```

```

i1+="sex:"
SVAR sex=root:$(df):S_sex
i1+=sex+"\r"

```

```

i1+="region:"
SVAR region=root:$(df):S_region
i1+=region+"\r"

```

```

i1+="genotype:"
SVAR geno=root:$(df):S_geno
i1+=geno+"\r"

```

```

i1+="app:"
SVAR w1=root:$(df):w1
i1+=w1+"\t"
SVAR w2=root:$(df):w2
i1+=w2+"\t"
SVAR w3=root:$(df):w3
i1+=w3+"\t"
SVAR w4=root:$(df):w4
i1+=w4+"\t"
SVAR w5=root:$(df):w5
i1+=w5+"\t"
SVAR w6=root:$(df):w6
i1+=w6+"\r"

```

```

i1+="points:"
SVAR poi=root:$(df):points
if(svar_exists(poi))
string add=replacestring(":",poi,"_")

```

```

        add=replacestring(";",add," ")
        add=replacestring("\r",add," ")
        i1+=add+"\r"
    else
        i1+="\r"
    endif

    i1+="loc:"
wave/T/Z loc=root:$(df):RoiLoc
    if(waveexists(loc))
        i1+=replacestring(";",loc[str2num(stringfromlist(1,n1,";"))],",")+ "\r"
    endif
    i1+="area:"
wave/Z ar=root:$(df):RoiArea
    if(waveexists(ar))
        i1+=num2str(ar[str2num(stringfromlist(1,n1,";"))])+"\r"
    endif

    n1+=";"
    n1+=i1
    note      /K curW, n1

end

Function SetVarProc_2filterAge(ctrlName,varNum,varStr,varName) : SetVariableControl
    String ctrlName
    Variable varNum
    String varStr
    String varName

    SVAR f1=root:analysis:filters
    NVAR a1=root:analysis:F_age1
    NVAR a2=root:analysis:F_age2
    string ol=stringfromlist(0, f1,"r")
    string nl=stringfromlist(0,ol,":")+":"+stringfromlist(1,ol,":")+":"+num2str(a1)+":"+num2str(a2)

        f1= replacestring(ol,f1,nl)
        applyfilters()

End

Function ListBoxProc_1(ctrlName,row,col,event) : ListBoxControl
    String ctrlName
    Variable row
    Variable col
    Variable event      //1=mouse down, 2=up, 3=dbl click, 4=cell select with mouse or keys
                        //5=cell select with shift key, 6=begin edit, 7=end

    //print row,event

    if((event==4) || (event==5))
        ModifyGraph/W=CANA#ME lsize=1, rgb=(65000,0,0)
        markT()
    endif

    return 0

End

Function PopMenuProc_1(ctrlName,popNum,popStr) : PopupMenuControl

```

```

String ctrlName
Variable popNum
String popStr
string/G picn1=popstr

GetTracesFromDF()
getvaluesfromdf(popstr,"all")
SetVarProc_Appli1("",0,"","")
End

```

```

Function ButtonProc_selEXP(ctrlName) : ButtonControl
String ctrlName
string el=getEXP()
variable/G whichexp

if(stringmatch(ctrlName,"*4"))//reverse
    if(whichexp>0)
        whichexp-=1
    endif
endif

if(stringmatch(ctrlName,"*5"))//forward
    if(whichexp<itemsinlist(el,";")-1)
        whichexp+=1
    endif
endif

string match=stringfromlist(whichexp,el,";")
//print match
wave/T lw=:lw
wave selw=:selw
variable p1
variable rem=-1
for(p1=0;p1<numelems(lw);p1+=1)
    if(stringmatch(lw[p1],match+"*"))
        if(rem<0)
            rem=p1
        endif
        selw[p1]=1
    else
        selw[p1]=0
    endif
endfor
ModifyGraph/W=CANA#ME lsize=1, rgb=(65000,0,0)
markT()
listbox list0, win=cana,row=(rem)
End

```

```

Function ButtonProc_6browse(ctrlName) : ButtonControl
String ctrlName
dowindow/K canasel
fCanaSel()
PopupMenuProc_1("",0,stringfromlist(0,getwl(),";"))
End

```

```

function appendinfo2exp()
controlinfo/W=canasel popup0
string e1=s_value
    e1=replacestring(" ",e1,"_")
    e1=    replacestring(".tif",e1,"")

string el=getexp()
variable p1
    for(p1=0;p1<itemsinlist(el,";");p1+=1)
string match=stringfromlist(p1,el,";")
        //print match, e1, stringmatch(match,e1+"*")
        if(stringmatch(match,e1+"*"))

                ButtonProc_1append("")
                return 1
        endif

    endfor

end

```

```

Function ButtonProc_1append(ctrlName) : ButtonControl
String ctrlName

```

```

SVAR picN=root:picN1
    make/o/N=0 curTrace
wave/T st=root:$(picN):roiMEAN
    if(!waveexists(st))
        return 0
    endif
    controlinfo/W=canaSel popup0
string e1=S_value
    e1=replacestring(" ",e1,"_")
    e1=    replacestring(".tif",e1,"")

variable p1
    for(p1=0;p1<numpts(st);p1+=1)

wave cw=root:AnaSel:$(e1+"__"+num2str(p1))
        if(waveexists(root:AnaSel:$(e1+"__"+num2str(p1))))
            note/K cw,picN    "+",""+num2str(p1)
            appendInfo(nameofwave(cw))
            //app2ME(root:AnaSel:$(e1+"__"+num2str(p1)),e1,p1)
            if(datafolderexists("root:analysis:allTraces:"))
                duplicate/O root:AnaSel:$(e1+"__"+num2str(p1)), root:analysis:allTraces:$(e1+"__"+num2str(p1))
            endif
        endif

    endfor

end

```

```

Function PopMenuProc_3setapl(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum
    String popStr
    //print ctrlName[5]
    SVAR app=root:$( "app"+num2str(str2num(ctrlName[5])+1))
    app=popstr

```

```

End

```

```

function /S getgeno()

```

```

    string expo=""
    expo+="Csf1r-2A-GCaMP6m"+";"
    expo+="Csf1r-2A-mCherry-2A-GCaMP6m"+";"
    expo+="5xFAD/Csf1r-2A-mCherry-2A-GCaMP6m"+";"
    return expo

```

```

end

```

```

function mergeIT(var, r1,r2, app)

```

```

    variable var //
    variable r1//
    variable r2//range
    string app //num2str app#
    //print var, r1,r2, app
    make/O/N=(var) root:merge1=0

```

```

    wave m1=root:merge1
    dowindow/K w_merge
    display/K=1/N=W_merge as "Average Graph"

```

```

        setdatafolder root:merge
    string wl=wavelist("*,",",","")
    wave scw=root:merge:$(stringfromlist(0,wl,","))
    setscale/p x, (leftx(scw)),(dimdelta(scw,0)), m1

```

```

    variable p1
    for(p1=0;p1<itemsinlist(wl,",");p1+=1)

```

```

        wave curW=root:merge:$(stringfromlist(p1,wl,","))
        mergenans(stringfromlist(p1,wl,","))
        m1+=curw

        appendtograph/W=W_merge curw
        ModifyGraph/W=W_merge rgb($(stringfromlist(p1,wl,",")))=(40000,40000,40000)

```

```

        endfor

        m1=/itemsinlist(wl,";")
        appendtograph/W=W_merge m1
        modifygraph/W=W_merge rgb(merge1)=(0,0,0),lsize(merge1)=2
        SetAxis/W=W_merge left -.5,2
        setdatafolder root:

        SetDrawEnv/W=W_merge xcoord= bottom,ycoord=rel, linethick= 2.00,save

NVAR as=root:$("as"+app)
NVAR ae=root:$("ae"+app)
SVAR na=root:$("app"+app)

        drawline/W=W_merge as,0.1,ae,0.1
        drawtext/W=W_merge as,0.1, na
NVAR nc1=root:nc1
        drawtext/W=W_merge 0,0.1, num2str(itemsinlist(wl,";"))+"/"+num2str(nc1)

        killdatafolder/Z root:merge
        //draw bas and app!

end

function mergenans(waven)
string waven

wave curW=root:merge:$ (waven)
        wavestats/Q curw

        if(V_numNaNs)
variable p1
                for(p1=0;p1<numpts(curw);p1+=1)
                        if(numtype(curw[p1]))
                                curw[p1]=curw[p1-1]
                        endif
                endfor
        endif
end

function flmport()

        killdatafolder/Z root:analysis:import
        newdatafolder/O/S root:analysis:import
string/G pathN
        setdatafolder root:
        dowindow/K W_import

        PauseUpdate; Silent 1           // building window...

        NewPanel/K=1/N=W_import /W=(273,98,604,792) as "Import experiments"
        ModifyPanel cbRGB=(39168,39168,39168)

SVAR p1=root:analysis:import:pathN
        SetVariable setvar0,pos={2,23},size={327,16},bodyWidth=298,title="Path "
        SetVariable setvar0,limits={-inf,inf,0}, value=p1

        CheckBox check0,pos={32,56},size={67,14},title="from folder",value= 0
        CheckBox check0 proc=CheckProc_2importFile
        CheckBox check1,pos={110,56},size={77,14},title="from .pxp file",value= 0

```

```

CheckBox check1 proc=CheckProc_2importFile,value=1

Button button2,pos={213,52},size={100,20},title="Browse"
Button button2 proc=ButtonProc_6BrowseIT

make/O/T/N=0 root:analysis:import:XX44lw
wave/T lw=root:analysis:import:XX44lw
make/O/N=0 root:analysis:import:XX44selw
wave selw=root:analysis:import:XX44selw
ListBox list0,pos={1,84},size={330,515},listwave=lw,selwave=selw
ListBox list0 mode=9
listbox list0 widths={250,80}

Button button0,pos={203,631},size={100,40},title="Load"
Button button0 proc=ButtonProc_6ImportIT
Button button1,pos={23,631},size={100,40},title="Cancel"
Button button1 proc=ButtonProc_6ImportIT

End

Function CheckProc_2importFile(ctrlName,checked) : CheckBoxControl
String ctrlName
Variable checked

variable which=str2num(ctrlName[5])
checkbox $("check"+num2str(!which)), win=w_import, value=(!checked)
ButtonProc_6BrowseIT("AUTO")

End

Function ButtonProc_6ImportIT(ctrlName) : ButtonControl
String ctrlName

if(!str2num(ctrlName[6]))//load
wave/T lw=root:analysis:Import:XX44lw
wave selw=root:analysis:Import:XX44selw
variable p1
for(p1=0;p1<numpts(lw);p1+=1)
    if(selW[p1])
        // print lw[p1]
        if(datafolderExists(lw[p1]))
            killdatafolder root:${lw[p1]}
        endif
        MoveDataFolder root:analysis:Import:${lw[p1]}, root:
    endif
endfor
endif

dowindow/K W_import
killdatafolder root:analysis:import

End

Function ButtonProc_6BrowseIT(ctrlName) : ButtonControl
String ctrlName

killdatafolder/Z root:analysis:import
newdatafolder/S root:analysis:import
string/G pathN
setdatafolder root:

if(stringmatch(ctrlName,"button2"))
    NewPath /M="Select Path..." /O/Q/Z import44

```

```

        if(v_flag) //not succeeded
            return 0
        endif
    endif
    updatell()
    pathinfo import44
svar s1=root:analysis:import:pathN
    s1=s_path
    SetVariable setvar0,limits={-inf,inf,0}, value=s1,win=W_import

End

function updatell()

    controlinfo/W=W_import check1//from pxp
    findana(v_value)//1=pxp,0=datafolder; findana also loads the data into a temp folder

variable n1=CountObjects("root:analysis:import",4)

    make/O/T/N=0 root:analysis:import:XX44lw
wave/T lw=root:analysis:import:XX44lw
    make/O/N=0 root:analysis:import:XX44selw
wave selw=root:analysis:import:XX44selw
    ListBox list0,pos={1,84},size={330,515},listwave=lw,selwave=selw

variable p1
string remov=""

    for(p1=0;p1<n1;p1+=1)
        //print getindexedobjName("root:analysis:import",4,p1)
wave/Z curW=root:analysis:import:${getindexedobjName("root:analysis:import",4,p1)}:RoiMEAN
        //    print (waveexists(curW))
            if(waveexists(curW))
                insertpoints (numpnpts(lw)),1, lw,selw
                lw[numpnpts(lw)-1]=getindexedobjName("root:analysis:import",4,p1)
            else
                remov+=getindexedobjName("root:analysis:import",4,p1)+";"
            endif
        endfor
    for(p1=0;p1<itemsinlist(remov,";");p1+=1)
        killdatafolder root:analysis:import:${stringfromlist(p1,remov,";")}
    endfor

end

function/S findana(var1)
variable var1 //0:datafolder;1:pxp
//load all anafolders into root:analysis/import
//path import44 exists

string dsf=""

    // Store directory list in a free text wave.
    // The free wave is automatically killed when the function returns.

    Make/O/N=1/T dirs

```



```

//edit/K=1 dirs
pathinfo import44
dirs[0]=s_path

variable p1=0,q1
variable np
do
    newpath/O/Q tempi44, (dirs[p1]) //look into the datafolders that were found
    dsf=Indexaddir(tempi44,-1,1)
    if(strlen(dsf))
        np=umpnts(dirs)
        insertpoints (np), (itemsinlist(dsf,";")), dirs
        for(q1=np;q1-np<(itemsinlist(dsf,";"));q1+=1)
            dirs[q1]=stringfromlist(q1-np,dsf,";")
        endfor
    endif
    p1+=1
while(p1<umpnts(dirs))

if(!var1) //data folder
    for (p1=0;p1<umpnts(dirs);p1+=1)
        newpath/O/Q tempi44, (dirs[p1]) //look into the datafolders that were found
        string files=indexedfile(tempi44,-1,"????")
        if(stringmatch (files,"*ROI MEAN*"))
            setdatafolder root:analysis:import
            loaddata/D/O/Q/T/P=tempi44/R dirs[p1]
            setdatafolder root:
        endif
    endfor
else //pxps
//    print "1", umpnts(dirs), dirs[0]
    for (p1=0;p1<umpnts(dirs);p1+=1)
        newpath/O/Q tempi44, (dirs[p1]) //look into the datafolders that were found
        pathinfo tempi44

        string path=S_path
        string pxp=indexedfile(tempi44,-1,".pxp")

        if(strlen(pxp))
            for(q1=0;q1<itemsinlist(pxp,";");q1+=1)
                setdatafolder root:analysis:import
                loaddata/O/Q/P=tempi44/R f1
                setdatafolder root:
            endfor
        endif

    endfor

endif

end

Function ButtonProc_6(ctrlName) : ButtonControl
    String ctrlName

    fimport()

End

Function PopMenuProc(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum

```

```

String popStr
if(popnum==1)
    checkbox check2 win=CANA, disable =2
else
    checkbox check2 win=CANA, disable =0
endif

End

function ButtonProc7_dispSel(lasch):ButtonControl
string lasch

string wl= getSelected()
dowindow/K W_sel
display/N=w_sel/K=1 as "Selected Traces"
variable p1
for(p1=0;p1<itemsinlist(wl,"");p1+=1)
wave curW=root:anasel:$(stringfromlist(p1,wl,";"))
    appendtograph/W=W_sel curW

endfor

end

function distF0(lasch):buttonControl
string lasch
NVAR ThrEx
wave/T lw
    make/O/N=0 res_F0
wave selW
    selw=0
variable p1

for(p1=0;p1<numpts(lw); p1+=1)

    insertpoints (p1),1,res_f0
wave curw=root:anasel:$(lw[p1])
    res_f0[p1]=curw[0]
    if(curw[0]<ThrEx)
        selw[p1]=1
    endif
endfor

end

function fCANA()
globals()
dowindow/K CANA

PauseUpdate; Silent 1 // building window...
NewPanel/N=CANA /K=1/W=(310,153,1578,843) as " "

Button button09,pos={15,2},size={50,30},proc=ButtonProc_6,title="Import\rexp"
Button button8,pos={178,2},size={50,30},proc=ButtonProc_appendall,title="Append\rall exp"
Button button9,pos={229,2},size={50,30},proc=ButtonProc_6browse,title="Browse\rexp"
Button button2,pos={1196,2},size={50,30},proc=ButtonProc_2,title="Clear"
Button button14,pos={747,3},size={70,30},proc=ButtonProc_fSPON,title="Spontaneous"

•variable/G ThrEx=200
CheckBox check1,pos={831,11},size={94,14},proc=CheckProc_corr,title="Correct baseline"
CheckBox check1,value= 0
SetVariable setvar23,pos={1021,8},size={50,16},limits={-inf,inf,0}

```

```

CheckBox check2,pos={831,11},size={94,14},proc=CheckProc_corr,title="Correct baseline"
CheckBox check2,value= 0
SetVariable setvar24 title=" "
SetVariable setvar24,pos={1021,8},size={50,16},limits={-inf,inf,0}, value=ThrEx
Button button7,pos={941,8},size={70,20},title="Mark fluo4-", proc=distF0

Button button13,pos={1116,8},size={50,20},proc=fapp3,title="App3"

PopupMenu popup5,pos={332,4},size={132,21},title="Normalisation"
PopupMenu popup5,mode=1,popvalue="Division",value= #""\Division;Subtraction\""
PopupMenu popup5 proc=PopMenuProc
CheckBox check2,pos={493,9},size={77,14},title="norm to ATP",value= 0
NVAR factor
SetVariable setvar22,pos={611,11},size={79,16},bodyWidth=30,title="SD factor"
SetVariable setvar22,value= factor

GroupBox group0,pos={12,35},size={270,650},labelBack=(39168,39168,39168)
ListBox list0,pos={18,41},size={260,239},proc=ListBoxProc_1,listWave=root:lw
ListBox list0,selWave=root:selw,mode= 4
Button button1,pos={43,290},size={200,20},proc=ButtonProc_3Delete,title="Delete"

GroupBox group15,pos={19,321},size={258,82},title="Display"
CheckBox bckgr4,pos={41,347},size={37,14},proc=CheckProc_1,title="ROI"
CheckBox bckgr4,value= 1,mode=1
CheckBox bckgr2,pos={41,377},size={75,14},proc=CheckProc_1,title="background"
CheckBox bckgr2,value= 0,mode=1
CheckBox bckgr3,pos={156,348},size={86,14},proc=CheckProc_1,title="bg subtracted "
CheckBox bckgr3,value= 1
CheckBox bckgr1,pos={157,377},size={71,14},proc=CheckProc_1,title="normalized "
CheckBox bckgr1,value= 1

GroupBox group16,pos={19,411},size={258,82},title="Filter"
CheckBox check0,pos={40,438},size={43,14},proc=CheckProc_2Filter,title="Filter "
CheckBox check0,value= 0
Button button6,pos={39,462},size={80,20},title="Store Filter"
PopupMenu popup3,pos={166,463},size={80,21},bodyWidth=80,mode=1,value= #""\""

GroupBox group17,pos={19,501},size={258,82},title="Selection"
Button button0,pos={45,523},size={80,20},proc=ButtonProc_5,title="select all"
Button button3,pos={163,522},size={80,20},proc=ButtonProc_5reversel,title="reverse sel"
Button button4,pos={45,556},size={60,20},proc=ButtonProc_selEXP,title="<< exp"
Button button5,pos={115,556},size={60,20},proc=ButtonProc_selEXP,title="exp >>"
Button button12,pos={209,555},size={50,20},proc=ButtonProc7_dispSel,title="Display"

GroupBox group18,pos={19,592},size={258,82},title="Set Offset"
PopupMenu popup4,pos={158,620},size={88,21},bodyWidth=20,proc=PopMenuProc_3byAPP,title="by
application"
PopupMenu popup4,mode=6,popvalue="0.1TLQP21",value= #"getselappl()"

NVAR xOff
SetVariable setvar01,pos={36,622},size={98,16},bodyWidth=60,proc=SetVarProc,title="x offset"
SetVariable setvar01,limits={-inf,inf,10},value= xOff
Button button10,pos={55,647},size={80,20},proc=DP,title="DeletePoints"
Button button11 pos={147,647},size={80,20},title="InsertPoints",proc=IP

GroupBox group1,pos={295,35},size={955,650},labelBack=(39168,39168,39168)

GroupBox group2,pos={306,40},size={450,290},title="application 1"
GroupBox group2,labelBack=(56576,56576,56576),frame=0
SVAR app1
SetVariable setvar0,pos={321,65},size={150,16},bodyWidth=150,title=" "
SetVariable setvar0,limits={-inf,inf,0},value= app1
PopupMenu popup0,pos={472,62},size={20,21},bodyWidth=20,proc=PopMenuProc_3setappl
PopupMenu popup0,mode=8,popvalue="glutamate 1 mM",value= #"getappl()"
GroupBox group3,pos={321,87},size={89,80},title="basal"
GroupBox group3,labelBack=(30464,30464,30464)

```

NVAR bas1,bae1,mb1

```
SetVariable setvar2,pos={330,112},size={32,16},title=" "  
SetVariable setvar2,limits={-inf,inf,0},value= bas1  
SetVariable setvar3,pos={368,112},size={32,16},title=" "  
SetVariable setvar3,limits={-inf,inf,0},value= bae1  
SetVariable setvar1,pos={330,142},size={70,16},title=" "  
SetVariable setvar1,limits={-inf,inf,0},value= mb1
```

```
Button ba1,pos={342,173},size={50,20},proc=getM,title="get basal"
```

```
GroupBox group4,pos={411,87},size={86,81},title="appl."  
GroupBox group4,labelBack={30464,30464,30464}
```

NVAR as1,ae1,ma1

```
SetVariable setvar4,pos={420,112},size={32,16},title=" "  
SetVariable setvar4,limits={-inf,inf,0},value= as1  
SetVariable setvar5,pos={458,112},size={32,16},title=" "  
SetVariable setvar5,limits={-inf,inf,0},value= ae1  
SetVariable setvar6,pos={420,142},size={70,16},title=" "  
SetVariable setvar6,limits={-inf,inf,0},value= ma1
```

```
Button ba2,pos={434,174},size={50,20},proc=getM,title="get appl."
```

```
GroupBox group5,pos={501,87},size={86,81},title="stats"  
GroupBox group5,labelBack={30464,30464,30464}
```

NVAR nc1,ne1

```
SetVariable setvar07,pos={509,111},size={70,16},title=" ",format="%g cells"  
SetVariable setvar07,limits={-inf,inf,0},value= nc1  
SetVariable setvar08,pos={509,141},size={70,16},title=" ",format="%g exp."  
SetVariable setvar08,limits={-inf,inf,0},value= ne1
```

```
GroupBox group6,pos={591,85},size={152,83},title="results"  
GroupBox group6,labelBack={30464,30464,30464}
```

NVAR resp1,amp1

```
SetVariable setvar05,pos={617,111},size={100,16},title=" ",format="%g%"  
SetVariable setvar05,limits={-inf,inf,0},value= resp1  
SetVariable setvar06,pos={618,141},size={100,16},title=" ",format="%g AU"  
SetVariable setvar06,limits={-inf,inf,0},value= amp1
```

```
CheckBox fil1,pos={635,176},size={52,14},title="filtered ",value= 0  
CheckBox fil1 proc=CheckProc_usefilter
```

```
GroupBox group11,pos={331,203},size={149,110},title="Select"  
GroupBox group11,labelBack={34816,34816,34816}  
Button sel_resp1,pos={359,236},size={80,20},proc=ButtonProc_3,title="responding"  
Button non_resp1,pos={359,266},size={80,20},proc=ButtonProc_3,title="non-responding"
```

```
GroupBox group12,pos={493,202},size={149,110},title="Action"  
GroupBox group12,labelBack={34816,34816,34816}  
CheckBox sel1,pos={500,234},size={59,14},title="sel. only ",value= 0  
CheckBox cor1,pos={580,234},size={42,14},title="corr. ",value= 1
```

NVAR range11,range12

```
SetVariable setvar16,pos={499,258},size={62,16},bodyWidth=40,title="bef."  
SetVariable setvar16,limits={-inf,inf,0},value= range11  
SetVariable setvar17,pos={570,258},size={65,16},bodyWidth=40,title="after"  
SetVariable setvar17,limits={-inf,inf,0},value= range12  
SetVariable setvar20,pos={500,289},size={53,16},bodyWidth=30,title="smo"  
SetVariable setvar20,value= smo44  
Button merge1,pos={556,285},size={80,20},proc=ButtonProc_4,title="Merge"
```

```
Button button10e,pos={670,265},size={50,40},proc=ButtonProc_ExportData,title="Export\rbeta"
```

```
GroupBox group7,pos={771,40},size={450,290},title="application 2"  
GroupBox group7,labelBack={56576,56576,56576},frame=0
```

SVAR app2

```
SetVariable setvar7,pos={786,65},size={150,16},bodyWidth=150,title=" "
```

```

SetVariable setvar7,limits={-inf,inf,0},value= app2
PopupMenu popup1,pos={937,62},size={20,21},bodyWidth=20,proc=PopupMenuProc_3setappl
PopupMenu popup1,mode=1,popvalue="ATP 1 mM",value= #"getapp()")

GroupBox group8,pos={786,85},size={89,80},title="basal"
GroupBox group8,labelBack=(30464,30464,30464)

NVAR bas2,bae2,mb2

SetVariable setvar8,pos={795,110},size={32,16},title=" "
SetVariable setvar8,limits={-inf,inf,0},value= bas2
SetVariable setvar9,pos={833,110},size={32,16},title=" "
SetVariable setvar9,limits={-inf,inf,0},value= bae2
SetVariable setvar02,pos={795,140},size={70,16},title=" "
SetVariable setvar02,limits={-inf,inf,0},value= mb2

Button ba3,pos={807,171},size={50,20},proc=getM,title="get basal"

GroupBox group9,pos={876,85},size={86,81},title="appl."
GroupBox group9,labelBack=(30464,30464,30464)

NVAR as2,ae2,ma2

SetVariable setvar09,pos={885,110},size={32,16},title=" "
SetVariable setvar09,limits={-inf,inf,0},value= as2
SetVariable setvar10,pos={923,110},size={32,16},title=" "
SetVariable setvar10,limits={-inf,inf,0},value= ae2
SetVariable setvar11,pos={885,140},size={70,16},title=" "
SetVariable setvar11,limits={-inf,inf,0},value= ma2

Button ba4,pos={894,171},size={50,20},proc=getM,title="get appl."

GroupBox group09,pos={966,85},size={86,81},title="stats"
GroupBox group09,labelBack=(30464,30464,30464)

NVAR nc2,ne2

SetVariable setvar12,pos={974,109},size={70,16},title=" ",format="%g cells"
SetVariable setvar12,limits={-inf,inf,0},value= nc2
SetVariable setvar13,pos={974,139},size={70,16},title=" ",format="%g exp."
SetVariable setvar13,limits={-inf,inf,0},value= ne2

GroupBox group10,pos={1056,83},size={152,83},title="results"
GroupBox group10,labelBack=(30464,30464,30464)

NVAR resp2,amp2

SetVariable setvar14,pos={1083,109},size={100,16},title=" ",format="%g%"
SetVariable setvar14,limits={-inf,inf,0},value= resp2
SetVariable setvar15,pos={1083,139},size={100,16},title=" ",format="%g AU"
SetVariable setvar15,limits={-inf,inf,0},value= amp2

GroupBox group13,pos={786,203},size={149,110},title="Select"
GroupBox group13,labelBack=(34816,34816,34816)
Button sel_resp2,pos={814,236},size={80,20},proc=ButtonProc_3,title="responding"
Button non_resp2,pos={814,266},size={80,20},proc=ButtonProc_3,title="non-responding"

GroupBox group14,pos={948,202},size={149,110},title="Action"
GroupBox group14,labelBack=(34816,34816,34816)
CheckBox sel2,pos={963,234},size={59,14},title="sel. only ",value= 0
CheckBox cor2,pos={1040,234},size={42,14},title="corr. ",value= 1

NVAR range21,range22

SetVariable setvar18,pos={1027,258},size={65,16},bodyWidth=40,title="after"
SetVariable setvar18,limits={-inf,inf,0},value= range22
SetVariable setvar19,pos={956,258},size={62,16},bodyWidth=40,title="bef."
SetVariable setvar19,limits={-inf,inf,0},value= range21
SetVariable setvar21,pos={953,289},size={53,16},bodyWidth=30,title="smo"
SetVariable setvar21,value= smo44
Button merge2,pos={1009,285},size={80,20},proc=ButtonProc_4,title="Merge"

Button button11e,pos={1130,265},size={50,40},proc=ButtonProc_ExportData,title="Export\rbeta"
CheckBox fil2,pos={1115,171},size={52,14},title="filtered ",value= 0
CheckBox fil2 proc=CheckProc_usefilter

```

```

Display/W=(307,335,1221,665)/HOST=#
RenameWindow #,ME
SetActiveSubwindow ##
End

```

```

function /S getpoints()

```

```

string fl=getwl() //folderlist

```

```

variable p1,q1,r1
string expo=""
string poi=""

```

```

    for(p1=0;p1<itemsinlist(fl,";");p1+=1)
        for(q1=1;q1<7;q1+=1)
SVAR/Z curapp=root:${stringfromlist(p1,fl,";")}:points
            if(sVAR_Exists(curapp))
                if(strlen(curapp))
                    for(r1=0;r1<itemsinlist(curapp,"\r");r1+=1)
                        poi=stringfromlist(0,stringfromlist(r1,curapp,"\r"),".")
                        if(!stringmatch(expo,"*" + poi + "*" ))
                            expo+=poi+","
                        endif
                    endfor
                endif
            endif
        endfor
    endfor
    if(strlen(expo))
        return expo
    else
        return "none"
    endif
end

```

```

function fW_filter()

```

```

makelbw()

```

```

PauseUpdate; Silent 1 // building window...
dowindow/K w_filter
NewPanel /K=1/N=W_filter /W=(221,293,829,999) as "Filter"

```

```

GroupBox group6a,pos={0,0},size={290,700},disable=2,title=" "
GroupBox group6a,labelBack=(52224,52224,52224)

```

```

CheckBox check0,pos={261,50},size={21,14},proc=CheckProc_2actFilter,title=" "

```

```

CheckBox check0,value= 0
GroupBox group0,pos={15,22},size={237,60},disable=2,title="Age"
GroupBox group0,labelBack=(39168,39168,39168)

SetVariable setvar0,pos={30,51},size={85,16},bodyWidth=40,disable=2,proc=SetVarProc_2filterAge,title="between"
SetVariable setvar0,format="P %g"
SetVariable setvar0,limits={-inf,inf,0},value= root:Analysis:F_age1
SetVariable setvar1,pos={123,51},size={62,16},bodyWidth=40,disable=2,proc=SetVarProc_2filterAge,title="and"
SetVariable setvar1,format="P %g"
SetVariable setvar1,limits={-inf,inf,0},value= root:Analysis:F_age2

CheckBox check1,pos={261,120},size={21,14},proc=CheckProc_2actFilter,title=" "
CheckBox check1,value= 0
GroupBox group1,pos={15,92},size={237,60},disable=2,title="Sex"
GroupBox group1,labelBack=(39168,39168,39168)
CheckBox check5,pos={64,122},size={44,14},disable=2,proc=CheckProc_2filterSex,title="Male "
CheckBox check5,value= 1
CheckBox check6,pos={144,122},size={55,14},disable=2,proc=CheckProc_2filterSex,title="Female "
CheckBox check6,value= 1

CheckBox check2,pos={261,190},size={21,14},proc=CheckProc_2actFilter,title=" "
CheckBox check2,value= 0
GroupBox group2,pos={15,162},size={235,200},disable=2,title="Region"
GroupBox group2,labelBack=(39168,39168,39168)
ListBox list0,pos={35,194},size={196,150},disable=2,proc=ListBoxProc_2
ListBox list0,listWave=root:Analysis:reg_listw,selWave=root:Analysis:reg_selw
ListBox list0,mode= 4

CheckBox check3,pos={261,396},size={21,14},proc=CheckProc_2actFilter,title=" "
CheckBox check3,value= 0
GroupBox group3,pos={15,368},size={235,120},disable=2,title="Genotype"
GroupBox group3,labelBack=(39168,39168,39168)
ListBox list1,pos={35,400},size={196,70},disable=2,proc=ListBoxProc_2
ListBox list1,listWave=root:Analysis:geno_listw,selWave=root:Analysis:geno_selw
ListBox list1,mode= 4

CheckBox check4,pos={261,526},size={21,14},proc=CheckProc_2actFilter,title=" "
CheckBox check4,value= 0
GroupBox group4,pos={14,488},size={235,200},disable=2,title="Application"
GroupBox group4,labelBack=(39168,39168,39168)
ListBox list2,pos={35,530},size={196,150},disable=2,proc=ListBoxProc_2
ListBox list2,listWave=root:Analysis:app_listw,selWave=root:Analysis:app_selw
ListBox list2,mode= 4

GroupBox group7,pos={312,0},size={290,700},disable=2,title=" "
GroupBox group7,labelBack=(52224,52224,52224)

CheckBox check5a,pos={574,50},size={21,14},proc=CheckProc_2actFilter,title=" "
CheckBox check5a,value= 0
GroupBox group5,pos={328,22},size={239,130},disable=2,title="Location- relative"
GroupBox group5,labelBack=(39168,39168,39168)
PopupMenu popup0,pos={344,54},size={135,21},bodyWidth=100,disable=2,proc=PopupMenuProcpoints,title="point 1"
PopupMenu popup0,mode=1,popvalue="STM",value= #"getpoints()"
PopupMenu popup1,pos={344,84},size={135,21},bodyWidth=100,disable=2,proc=PopupMenuProcpoints,title="point 2"
PopupMenu popup1,mode=2,popvalue="\nREC",value= #"getpoints()"
SetVariable setvar2,pos={475,121},size={72,16},bodyWidth=50,disable=2,title="and"
SetVariable setvar2,format="%g %"
SetVariable setvar2,limits={-inf,inf,0},value= root:Analysis:F_dist2
SetVariable setvar3,pos={342,121},size={95,16},bodyWidth=50,disable=2,title="between"
SetVariable setvar3,format="%g %"
SetVariable setvar3,limits={-inf,inf,0},value= root:Analysis:F_dist1
Button button0,pos={497,54},size={50,50},disable=2,proc=ButtonProc_FilterAna,title="Analyse"

```

```

CheckBox check6a,pos={574,200},size={21,14},proc=CheckProc_2actFilter,title=" "
CheckBox check6a,value= 0
GroupBox group6,pos={328,172},size={239,130},title="Location- absolute"
GroupBox group6,labelBack={39168,39168,39168},disable=2
PopupMenu popup2,pos={371,264},size={149,21},bodyWidth=100,proc=PopupMenuProcpoints,title="from point"
PopupMenu popup2,mode=1,popvalue="STM",value= #"getpoints()",disable=2
SetVariable setvar4,pos={336,202},size={137,16},bodyWidth=50,title="distance (minimal)"
SetVariable setvar4,format="%g px",disable=2
SetVariable setvar4,limits={-inf,inf,0},value= root:Analysis:F_distA
CheckBox check7,pos={503,203},size={32,14},proc=CheckProc_switchunit,title="μm"
CheckBox check7,value= 0,disable=2
SetVariable setvar6,pos={480,222},size={81,16},bodyWidth=50,title="factor"
SetVariable setvar6,format="%g μm/px",disable=1
SetVariable setvar6,limits={-inf,inf,0},value= root:Analysis:F_distF
SetVariable setvar6 bodyWidth=80

```

End

```
#pragma rtGlobals=3          // Use modern global access method and strict wave access.
```

```

function  applyfilters()

SVAR f1=root:analysis:filters

        backuptraces(0)
        //checkLW() is included in backuptraces
        duplicate/O root:selw, root:analysis:filterW
wave fw1=root:analysis:filterW
        fw1=0
wave/T lw=root:lw
variable p1

make/O/N=(numpts(fw1)) root:temp,root:temp1
wave t1=root:temp
wave t2=root:temp1
        for(p1=0;p1<numpts(lw); p1+=1)//all cells 10;p1+=1)//
wave curW=root:anasel:$(lw[p1])
        //print note(curw)
        if(str2num(stringfromlist(1,stringfromlist(0, f1,"\\r"),".")))//apply age filter
string age=stringbykey("age",stringfromlist(2,note(curW),";",""),":","\\r")
variable ag1=-1
                                //print "age", age
                                if(strlen(age))
                                        ag1=str2num(age)
                                endif

NVAR a1=root:analysis:F_age1
NVAR a2=root:analysis:F_age2
//print ag1,a1,a2, numtype(ag1)
        //condition
                                if((ag1<a1)|| (ag1>a2))
                                print ag1,a1,a2,stringfromlist(p1,lw,";"),age,"J"
                                        fw1[p1]=1 //delete later
                                endif
                                if(numtype(ag1))          //age not specified or unreadable

```



```

                                fw1[p1]=1 //delete later
                                endif
                                endif

                                if(str2num(stringfromlist(1,stringfromlist(1, f1,"\\r"),":")))//apply sex filter
                                //print "sex"
                                string sex=stringbykey("sex",stringfromlist(2,note(curW),";"," ":"\\r")
                                //print nameofwave(curw), sex
                                variable match
                                variable fmatch=str2num(stringfromlist(2,stringfromlist(1, f1,"\\r"),":"))
                                if((fmatch<3))// sex filter
                                if(stringmatch(sex, "Male"))
                                match=1
                                elseif (stringmatch(sex, "Female"))
                                match=2
                                endif
                                //condition
                                //print p1, match, fmatch
                                if((match!=fmatch))
                                fw1[p1]=1 //delete later
                                endif
                                endif
                                endif

                                if(str2num(stringfromlist(1,stringfromlist(2, f1,"\\r"),":")))//apply region filter
                                //print "reg"
                                string reg=stringbykey("region",stringfromlist(2,note(curW),";"," ":"\\r")
                                string rl=getfiltersel("reg")
                                if(strlen(rl))
                                //condition
                                if(!stringmatch(rl,"*" + reg + "*"))
                                fw1[p1]=1 //delete later
                                endif
                                endif
                                endif

                                if(str2num(stringfromlist(1,stringfromlist(3, f1,"\\r"),":")))//apply genotype filter
                                //print "gen"
                                string geno=stringbykey("genotype",stringfromlist(2,note(curW),";"," ":"\\r")
                                string gl=getfiltersel("geno")
                                if(strlen(gl))
                                //condition
                                if(!stringmatch(gl,"*" + geno + "*"))
                                fw1[p1]=1 //delete later
                                endif
                                endif
                                endif

                                if(str2num(stringfromlist(1,stringfromlist(4, f1,"\\r"),":")))//apply app filter
                                //print "app"
                                string al=getfiltersel("app")
                                //print al
                                if(strlen(al))
                                //condition
                                string app=stringbykey("app",stringfromlist(2,note(curW),";"," ":"\\r")
                                variable q1, r1, rem1
                                for(q1=0;q1<itemsinlist(al, ";");q1+=1)
                                rem1=0
                                for(r1=0;r1<6;r1+=1)

                                string cw=stringfromlist(r1,app,"\\t")
                                //print cw,"!", app

                                if((strlen(cw))&&(stringmatch(al,"*" + cw + "*")))
                                rem1+=1
                                endif
                                endfor

```

```

                                if(!rem1)
                                    fw1[p1]=1 //delete later
                                endif
                            endfor
                        endif
                    endif

                    if(str2num(stringfromlist(1,stringfromlist(5, f1,"\\r"),":")))//apply loc rel filter
                        //print "loc",stringfromlist(5, f1,"\\r")
                        controlinfo/w=w_filter popup0
                    string poi1=s_value
                        controlinfo/w=w_filter popup1
                    string poi2=s_value
                    string info=note(curw)

//cell location relative
variable cx=str2num(stringfromlist(0, findloc(lw[p1]),",", ""))
variable cy=str2num(stringfromlist(1, findloc(lw[p1]),",", ""))

SVAR/Z poi=root:$(stringfromlist(0,info,";")):points
                                if(SVAR_Exists(poi))
                                    // print lw[p1]+"\\r"+poi
                                variable p1x=str2num(stringfromlist(0,stringbykey(poi1,poi,":", "\\r"),":"))
                                variable p1y=str2num(stringfromlist(1,stringbykey(poi1,poi,":", "\\r"),":"))
                                variable p2x=str2num(stringfromlist(0,stringbykey(poi2,poi,":", "\\r"),":"))
                                variable p2y=str2num(stringfromlist(1,stringbykey(poi2,poi,":", "\\r"),":"))

                                variable d100=sqrt((p1x-p2x)^2+(p1y-p2y)^2)//distance pipettes
                                variable dp=sqrt((p1x-cx)^2+(p1y-cy)^2)//distance point1-cell
                                variable drel=100*dp/d100
                                    t1[p1]=dp
                                    t2[p1]=drel

//print drel
NVAR f_dist1=root:analysis:f_dist1
NVAR f_dist2=root:analysis:f_dist2

                                //print p1x,p1y,cx,cy,d100,nameofwave(curw),drel,f_dist1, f_dist2
                                if((drel>f_dist1)&&(drel<f_dist2))//keep
                                    else
                                        fw1[p1]=1 //delete later
                                    endif
                                else
                                    fw1[p1]=1 //delete later
                                endif

                                else
                                    //print lw[p1],"NOT"
                                endif

                    if(str2num(stringfromlist(1,stringfromlist(6, f1,"\\r"),":")))//apply loc abs filter
                        //print "loc",stringfromlist(5, f1,"\\r")
                        controlinfo/w=w_filter popup2
                    string poi3=s_value

                    string info1=note(curw)

//cell location relative
variable cx1=str2num(stringfromlist(0, findloc(lw[p1]),",", ""))
variable cy1=str2num(stringfromlist(1, findloc(lw[p1]),",", ""))

SVAR/Z poi_=root:$(stringfromlist(0,info1,";")):points

```

```

                if(SVAR_Exists(poi_))
                    //      print lw[p1]+"r"+poi
variable p3x=str2num(stringfromlist(0,stringbykey(poi3,poi_,"","\r"),";"))
variable p3y=str2num(stringfromlist(1,stringbykey(poi3,poi_,"","\r"),";"))

variable dpA=sqrt((p3x-cx1)^2+(p3y-cy1)^2)//distance point3-cell

//print drel
variable distT
NVAR f_distA=root:analysis:f_distA
NVAR f_distF=root:analysis:f_distF

                                controlinfo/W=w_filter check7
                                if(v_value)//µm
                                    distT=f_distA/f_distF
                                else
                                    distT=f_distA
                                endif
                                //print p1x,p1y,cx,cy,d100,nameofwave(curw),drel,f_dist1, f_dist2
                                if((dpA>distT))      //keep
                                    else
                                        fw1[p1]=1 //delete later
                                    endif
                                else
                                    fw1[p1]=1 //delete later
                                endif

                                else
                                    //print lw[p1],"NOT"
                                endif

                                endfor

//duplicate/O fw1,root:selw
//end

                                ButtonProc_3Delete("FILTER")
                                updateCANAME()

end

```

```
Function ButtonProc_ExportData(ctrlName) : ButtonControl
```

```
    String ctrlName
```

```
variable ce
```

```
    if(stringmatch(ctrlName[8],"c") )  
        ce=0 ///cell
```

```
    else
```

```
        ce=1//exp
```

```
    endif
```

```
variable ap
```

```
    if(stringmatch(ctrlName[7],"0") )//app1
```

```
        ap=1
```

```
    elseif(stringmatch(ctrlName[7],"1") )//app2
```

```
        ap=2
```

```
    elseif(stringmatch(ctrlName[7],"2") )//app2
```

```
        ap=3
```

```
    endif
```

```
string expo=""
```

```
    expo+="Experiment"+","
```

```
    if(!ce)//cell
```

```
        expo+="ROI"
```

```
    else
```

```
        expo+="number of ROIs"
```

```
    endif
```

```
    expo+=","
```

```
    expo+="Mouse"+ ","
```

```
    expo+="Age"+ ","
```

```
    expo+="Sex"+ ","
```

```
    expo+="Genotype"+ ","
```

```
    expo+="Region"+ ","
```

```
    expo+="Point1"+ ","
```

```
    expo+="Dist. Pint1"+ ","
```

```
    expo+="Point2"+ ","
```

```
    expo+="Dist. Point2"+","
```

```
    expo+="Application"+ ","
```

```
    expo+="Start App"+ ","
```

```
    expo+="End App"+ ","
```

```
    expo+="Basal tmie1"+","
```

```
    expo+="Basal time2"+","
```

```
    expo+="Basal Avg"+ ","
```

```
    expo+="Basal SD"+ ","
```

```
    expo+="App Avg"+ ","
```

```
    expo+="App Max10"+","
```

```
    expo+="App Avg_bas"+","
```

```
    expo+="App Max10_bas"+ ","
```

```
    expo+="App all_%ATP"+ ","
```

```

expo+="App resp_%ATP"+      ","
        if(ce)//exp
        expo+="% Responding"
    else
        expo+="Responding"
    endif
    expo+=","
    expo+="Filters"+      ","
    dowindow/K Win_export
    killwaves/Z w_expo
    Make/O/T/N=(2,26) W_expo
    w_expo[0][]=stringfromlist(q,expo)

string info
    if(ce)//exp
        info=getexp()

    else //cell
        info=getcells()

    endif

string add1=printFilter()

variable p1
    for(p1=0;p1<itemsinlist(info,"");p1+=1)
string add=getinfo(stringfromlist(p1,info,""),ce,ap)    +add1
        insertpoints (dimsize(W_expo,0)),1, w_expo
        w_expo[dimsize(W_expo,0)-1][]=stringfromlist(q,add,"")

        //w_expo[dimsize(W_expo,0)-1][24]=num2str(getrespE(stringfromlist(p1,info,""),ap))
    endfor

    edit/K=1/N=Win_export W_expo

End

```

```

function  fapp3(lasch):ButtonControl
string lasch

variable/G bas3, bae3
variable/G as3, ae3
variable/G nc3, ne3
variable/G resp3, amp3
variable/G range31, range32
String/G app3

variable/G mb3, ma3
    dowindow/K W_app3

PauseUpdate; Silent 1      // building window...
    NewPanel /K=1/W=(1395,96,1874,403)/N=W_app3 as "Application 3"

```

```

GroupBox group7,pos={11,10},size={450,290},title="application 3"
GroupBox group7,labelBack=(56576,56576,56576),frame=0

SetVariable setvar7,pos={26,35},size={150,16},bodyWidth=150,title=" "
SetVariable setvar7,limits={-inf,inf,0},value= app3

PopupMenu popup1,pos={177,32},size={20,21},bodyWidth=20,proc=PopupMenuProc_3setappl
PopupMenu popup1,mode=1,popvalue="ATP 1 mM",value= #"getapp()"

GroupBox group8,pos={26,55},size={89,80},title="basal"
GroupBox group8,labelBack=(30464,30464,30464)
SetVariable setvar8,pos={35,80},size={32,16},title=" "
SetVariable setvar8,limits={-inf,inf,0},value= bas3
SetVariable setvar9,pos={73,80},size={32,16},title=" "
SetVariable setvar9,limits={-inf,inf,0},value= bae3
SetVariable setvar02,pos={35,110},size={70,16},title=" "
SetVariable setvar02,limits={-inf,inf,0},value= mb3
Button ba5,pos={47,141},size={50,20},proc=getM,title="get basal"

GroupBox group9,pos={116,55},size={86,81},title="appl."
GroupBox group9,labelBack=(30464,30464,30464)
SetVariable setvar09,pos={125,80},size={32,16},title=" "
SetVariable setvar09,limits={-inf,inf,0},value= as3
SetVariable setvar10,pos={163,80},size={32,16},title=" "
SetVariable setvar10,limits={-inf,inf,0},value= ae3
SetVariable setvar11,pos={125,110},size={70,16},title=" "
SetVariable setvar11,limits={-inf,inf,0},value= ma3
Button ba6,pos={134,141},size={50,20},proc=getM,title="get appl."

GroupBox group09,pos={206,55},size={86,81},title="stats"
GroupBox group09,labelBack=(30464,30464,30464)
SetVariable setvar12,pos={214,79},size={70,16},title=" ",format="%g cells"
SetVariable setvar12,limits={-inf,inf,0},value= nc3
SetVariable setvar13,pos={214,109},size={70,16},title=" ",format="%g exp."
SetVariable setvar13,limits={-inf,inf,0},value= ne3
GroupBox group10,pos={296,53},size={152,83},title="results"
GroupBox group10,labelBack=(30464,30464,30464)
SetVariable setvar14,pos={323,79},size={100,16},title=" ",format="%g%"
SetVariable setvar14,limits={-inf,inf,0},value= resp3
SetVariable setvar15,pos={323,109},size={100,16},title=" ",format="%g AU"
SetVariable setvar15,limits={-inf,inf,0},value= amp3
GroupBox group13,pos={26,173},size={149,110},title="Select"
GroupBox group13,labelBack=(34816,34816,34816)
Button sel_resp3,pos={54,206},size={80,20},proc=ButtonProc_3,title="responding"
Button non_resp3,pos={54,236},size={80,20},proc=ButtonProc_3,title="non-responding"
GroupBox group14,pos={188,172},size={149,110},title="Action"
GroupBox group14,labelBack=(34816,34816,34816)
CheckBox sel3,pos={203,204},size={59,14},title="sel. only ",value= 0
CheckBox cor3,pos={280,204},size={42,14},title="corr. ",value= 1
SetVariable setvar18,pos={267,228},size={65,16},bodyWidth=40,title="after"
SetVariable setvar18,limits={-inf,inf,0},value= range32
SetVariable setvar19,pos={196,228},size={62,16},bodyWidth=40,title="bef."
SetVariable setvar19,limits={-inf,inf,0},value= range31
SetVariable setvar21,pos={193,259},size={53,16},bodyWidth=30,title="smo"
SetVariable setvar21,value= smo44
Button merge3,pos={249,255},size={80,20},proc=ButtonProc_4,title="Merge"

CheckBox fil3,pos={345,146},size={52,14},title="filtered ",value= 0
CheckBox fil3 proc=CheckProc_usefilter
Button button12e,pos={370,235},size={50,40},proc=ButtonProc_ExportData,title="Export\rbeta"

```

EndMacro

```

function/S getEXP()
wave/T n1=root:lw
wave fw1=root:fw1
variable expo
string rem=""
variable p1

    for(p1=0;p1<numpts(n1);p1+=1)
        if(fw1[p1])
wave curW=root:anase1:${n1[p1]}
        //print nameofwave(curw), waveexists(curW), n1[p1]
string e1=stringfromlist(0,note(curw),";");//str1[0,strsearch("190304_cc_thalamus_2__3","__",0)]
        e1=replacestring(" ",e1,"_")
        e1=        replacestring(".tif",e1,"__")

        if(findlistitem(e1,rem,";")<0)
            rem+=e1+";"
            expo+=1
        endif
    endif
endfor

return rem

end

```

```

function/S helpmeC(str1,var2)
string str1 /// cell name
variable var2//1=app1;2=app2
string expo=""

wave curw=root:anase1:${str1}
string info=note(curw)
    //1 experiment
    expo+=stringfromlist(0,info,";")+";"
    //2 ROI
    expo+=stringfromlist(1,info,";")+";"
    //3 mouse
SVAR/Z mouse=root:${stringfromlist(0,info,";")}:S_mouse
    expo+=mouse+";"
    //age
SVAR/Z age=root:${stringfromlist(0,info,";")}:s_Age
    expo+=age+";"
    //5 Sex
SVAR/Z sex=root:${stringfromlist(0,info,";")}:s_sex
    expo+=sex+";"
    //6 genotype
SVAR/Z geno=root:${stringfromlist(0,info,";")}:s_geno
    expo+=geno+";"
    //7 Region
SVAR/Z reg=root:${stringfromlist(0,info,";")}:s_region
    expo+=reg+";"

return expo

End

```

```

function/S getcells()
string expo=""
wave fw1=root:fw1
wave/T listw=:lw
variable p1
    for(p1=0;p1<numpts(listW);p1+=1)
        if(fw1[p1])
            expo+=listw[p1]+";"
        endif
    endfor

return expo

end

```

```

function app2ME(w1,s1,v1)

wave w1
string s1
variable v1
wave/T lw=root:lw
wave selw=root:selw

    getwindow CANA#ME, wavelist
wave/T wl=root:w_wavelist
string now=nameofwave(w1)

    findvalue/TEXT=now wl
    if(v_value>=0)/on graph
        removefromgraph/Z/W=CANA#ME $(nameofwave(w1))
        findvalue/TEXT=now lw
        deletepoints v_value,1,lw,selw
    endif

    appendto graph/W=CANA#ME root:AnaSel:$(nameofwave(w1))
    insertpoints (dimsize(lw,0)),1, lw,selw
    lw[dimsize(lw,0) -1] [0]=now

end

```

```

function checkLW()

variable nw=countobjects("root:anasel",1)
    make/O/T/N=(nw) root:lw
wave/T lw=root:lw
    make/O/N=(nw) root:selw=0,root:analysis:filterW=0
variable p1
    for(p1=0;p1<nw;p1+=1)
        lw[p1]=getindexedobjname("root:anasel", 1,p1)
    endfor
    sort/A lw,lw

end

```

```

function/S getapp()

string expo=""
    expo+="ATP 1 mM"+";"
    expo+="ATP 10 μM"+";"
    expo+="ATP 100 μM"+";"

```



```

        expo+="histamine 100 μM"+",";
        expo+="dopamine 100 μM"+",";
        expo+="carbachol 100 μM"+",";
        expo+="serotonine 100 μM"+",";
        expo+="GABA 1 mM"+",";
        expo+="GABA 100 μM"+",";
        expo+="glutamate 1 mM"+",";
        expo+="glutamate 100 μM"+",";

expo=sortlist(expo)

expo+=",";
expo+=sortlist((getwl()))

        return expo

end

function drawIT(l1,r1,lasch)
variable l1
variable r1
string lasch
string n1=""
        if(stringmatch (lasch,"*1")) ///basal1
NVAR l2=bas1
NVAR r2=bae1
                n1="bas1"
                setdrawlayer/K/W=CANA#ME progback
                setdrawenv/W=CANA#ME linethick=0,fillfgc= (0,5224,5224), xcoord=bottom, ycoord=prel,save

        elseif(stringmatch (lasch,"*2")) ///apl1
NVAR l2=as1
NVAR r2=ae1
                n1="app1"
                setdrawlayer/W=CANA#ME progback
                setdrawenv/W=CANA#ME linethick=0,fillfgc= (0,5224,5224), xcoord=bottom, ycoord=prel,save
        elseif(stringmatch (lasch,"*3")) ///basal2
NVAR l2=bas2
NVAR r2=bae2
                n1="bas2"
                setdrawlayer/K/W=CANA#ME userback
                setdrawenv/W=CANA#ME linethick=0,fillfgc= (5224,5224,5224), xcoord=bottom, ycoord=prel,save
        elseif(stringmatch (lasch,"*4")) ///apl2
NVAR l2=as2
NVAR r2=ae2
                n1="app2"
                setdrawlayer/W=CANA#ME userback
                setdrawenv/W=CANA#ME linethick=0,fillfgc= (5224,5224,5224), xcoord=bottom, ycoord=prel,save
        elseif(stringmatch (lasch,"*5")) ///basal3
NVAR l2=bas3
NVAR r2=bae3
                n1="bas3"
                setdrawlayer/K/W=CANA#ME useraxes
                setdrawenv/W=CANA#ME linethick=0,fillfgc= (5224,5224,5224), xcoord=bottom, ycoord=prel,save
        elseif(stringmatch (lasch,"*6")) ///apl3
NVAR l2=as3
NVAR r2=ae3
                n1="app3"
                setdrawlayer/W=CANA#ME useraxes
                setdrawenv/W=CANA#ME linethick=0,fillfgc= (5224,30000,30000), xcoord=bottom, ycoord=prel,save
        endif

```

```

        l2=l1
        r2=r1
//      print n1
        drawrect/W=CANA#ME l1,0,r1,1

        if(!datafolderexists("analysis"))
            newdatafolder analysis
        endif

wave temp43,temp44,temp45,temp46,temp47
wave/T temp42
//print numpts(temp43)
        duplicate/O temp42, root:analysis:$(n1+"_cell")
        duplicate/O temp43, root:analysis:$(n1+"_top10")
        duplicate/O temp44, root:analysis:$(n1+"_max")
        duplicate/O temp45, root:analysis:$(n1+"_min")
        duplicate/O temp46, root:analysis:$(n1+"_avg")
        duplicate/O temp47, root:analysis:$(n1+"_sdev")

End

Function PopMenuProcpoints(ctrlName,popNum,popStr) : PopupMenuControl
    String ctrlName
    Variable popNum
    String popStr

End

Function ButtonProc_FilterAna(ctrlName) : ButtonControl
    String ctrlName

    applyfilters()

End

function/S correctDFN(str1)
string str1//data folder

string df=replacestring(" ",stringbykey("FOLDERS",datafolderdir(1),":",";"))

string cmp=replacestring("_",str1,"")
        cmp=replacestring(" ",cmp,"")

variable p1

        for(p1=0;p1<itemsinlist(df,";");p1+=1)
string curDF=stringfromlist(p1,df,";")
                curDF=replacestring(" ",curDF,"")
                curDF=replacestring("_",curDF,"")
                if(stringmatch(curDF,cmp))
                        return stringfromlist(p1,df,";")
                endif
        endfor
        return "nope"
end

function distP(str1)
string str1//points
variable x1=str2num(stringfromlist(0,stringbykey("STM",str1,":","\r"),";"))
variable y1=str2num(stringfromlist(1,stringbykey("STM",str1,":","\r"),";"))
variable x2=str2num(stringfromlist(0,stringbykey("REC",str1,":","\r"),";"))
variable y2=str2num(stringfromlist(1,stringbykey("REC",str1,":","\r"),";"))

```

```

        return (sqrt((x1-x2)^2+(y1-y2)^2))
end

function/S getinfo(str1,var1,var2)
string str1 ///either cell or exp name
variable var1//1=exp;0=cell
variable var2//1=app1;2=app2

string expo=""
    if(!var1) //cell-wise
        expo=helpmeC(str1,var2)
    else
        //basic
        expo=helpmeE(str1,var2)
        //points
SVAR/Z f1=root:analysis:filters
        if(SVAR_exists(f1))
            endif

        ///stm points
        string str2=correctDFN(str1)
SVAR points=root:$(str2):points
        if(SVAR_exists(points))

            expo+=stringfromlist(0,stringbykey("STM",points,":","\r"),";")+ "/" +stringfromlist(1,stringbykey("STM",points,":","\r"),";")+";"
            expo+=num2str(distP(points)*.625)+";"

            expo+=stringfromlist(0,stringbykey("REC",points,":","\r"),";")+ "/" +stringfromlist(1,stringbykey("REC",points,":","\r"),";")+";"
            expo+=";"
            else
                expo+=";;;"
            endif
            //application
            expo+= getappE(var2)

            //responding
            expo+=getrespE(str1,var2)

            //filters

        endif

    return expo
end

```

```

function/S getappE(var2)

variable var2 //app
string expo=""

SVAR app=root:$("app"+num2str(var2))
    expo+=app+";"
NVAR ae=root:$("ae"+num2str(var2))
NVAR as=root:$("as"+num2str(var2))
NVAR bae=root:$("bae"+num2str(var2))
NVAR bas=root:$("bas"+num2str(var2))
    expo+=num2str(as)+";"
    expo+=num2str(ae)+";"
    expo+=num2str(bas)+";"
    expo+=num2str(bae)+";"

```

```

        return expo

end

function/S getrespE(str1,var2)
string str1 //exp
variable var2 //app

string expo=""

wave/T lw=root:analysis:$("bas"+num2str(var2)+"_cell")
wave basA=root:analysis:$("bas"+num2str(var2)+"_avg")
wave basS=root:analysis:$("bas"+num2str(var2)+"_sdev")
wave appA=root:analysis:$("app"+num2str(var2)+"_avg")
wave app10=root:analysis:$("app"+num2str(var2)+"_top10")
wave appATP=root:analysis:$("app"+num2str(2)+"_top10")//given that App2==ATP
wave basATP=root:analysis:$("bas"+num2str(2)+"_avg")//given that App2==ATP

wave curw=root:analysis:$("app"+num2str(var2)+"_resp")

        make/O/N=0                tempbasAVG,tempbasSD,tempappAVG,tempappmax10,tempappAVG_bas,tempAPPmax10_bas,temp44,
tempAPPmax10_basATP,tempAPPmax10_basATP1
variable p1

        for(p1=0;p1<numpts(lw);p1+=1)
            if(stringmatch(lw[p1],str1+"*"))
                insertpoints                                (numpts(tempbasAVG)),1,
tempbasAVG,tempbasSD,tempappAVG,tempappmax10,tempappAVG_bas,tempAPPmax10_bas,temp44,tempAPPmax10_basATP,tempAPPmax
10_basATP1

                tempbasAVG[numpts(tempbasAVG)-1]=basA[p1]
                tempbasSD[numpts(tempbasSD)-1]=basS[p1]
                tempappAVG[numpts(tempappAVG)-1]=appA[p1]
                tempappmax10[numpts(tempappmax10)-1]=app10[p1]
                tempappAVG_bas[numpts(tempappAVG_bas)-1]=appA[p1]-basA[p1]
                tempAPPmax10_bas[numpts(tempAPPmax10_bas)-1]=app10[p1]-basA[p1]
                if(curw[p1])
                    tempAPPmax10_basATP[numpts(tempAPPmax10_basATP)-1]=(app10[p1]-
basA[p1])/(appATP[p1]-basATP[p1])
                else
                    tempAPPmax10_basATP[numpts(tempAPPmax10_basATP)-1]=nan
                endif
                tempAPPmax10_basATP1[numpts(tempAPPmax10_basATP)-1]=(app10[p1]-basA[p1])/(appATP[p1]-
basATP[p1])

                temp44[numpts(temp44)-1]=curW[p1]

            endif
        endfor
        wavestats/Q        tempbasAVG
        expo+=num2str(v_avg)+";"
        wavestats/Q        tempbasSD
        expo+=num2str(v_avg)+";"
        wavestats/Q        tempappAVG
        expo+=num2str(v_avg)+";"
        wavestats/Q        tempappmax10
        expo+=num2str(v_avg)+";"
        wavestats/Q        tempappAVG_bas
        expo+=num2str(v_avg)+";"
        wavestats/Q        tempappmax10_bas
        expo+=num2str(v_avg)+";"
        wavestats/Q        tempAPPmax10_basATP1//all cells
        expo+=num2str(v_avg)+";"
        wavestats/Q        tempAPPmax10_basATP//responding only

```



```

        if(V_flag)
            return 0
        endif

string sel=getSelected()
variable p1
    for(p1=0;p1<itemsinlist(sel,"");p1+=1)
wave curW=root:anasel:${stringfromlist(p1,sel,";")}
        deletepoints s1,(e1-s1), curW
string n1=note(curw)
string nrem=StringByKey("action",n1,".;")
        nrem+="deletepoints "+num2str(s1)+" "+num2str(e1-s1)+"\r"
string newN=ReplaceStringByKey("action",n1,nrem,".;")
        note/K curw,newN
        //duplicate/O root:anasel:${stringfromlist(p1,sel,";")},root:analysis:AllTraces:${stringfromlist(p1,sel,";")}
    endfor

```

end

```

function IP(lasch)
string lasch

```

```

variable s1
variable e1

```

```

prompt s1,"Start"
prompt e1,"number of points"
doprompt "Delete points from selected traces", s1,e1

```

```

        if(V_flag)
            return 0
        endif

string sel=getSelected()
variable p1
    for(p1=0;p1<itemsinlist(sel,"");p1+=1)
wave curW=root:anasel:${stringfromlist(p1,sel,";")}
variable v1=curw[s1+e1]
        insertpoints s1,e1, curW
        curw[s1, s1+e1-1]=v1
string n1=note(curw)
string nrem=StringByKey("action",n1,".;")
        nrem+="insertpoints "+num2str(s1)+" "+num2str(e1)+"\r"
string newN=ReplaceStringByKey("action",n1,nrem,".;")
        note/K curw,newN
        //duplicate/O root:anasel:${stringfromlist(p1,sel,";")},root:analysis:AllTraces:${stringfromlist(p1,sel,";")}
    endfor

```

end

```

function doaction(str,w1)
string str //actions; \r-separated list
string w1 //wavename

```

```

        if(!strlen(str))
            return 0
        endif

```

```

variable p1

    for(p1=0; p1<itemsinlist(str,"\r");p1+=1)
        //print stringfromlist(p1,str,"\r")+ " root:anase1:""+w1+""
            execute stringfromlist(p1,str,"\r")+ " root:anase1:""+w1+""
        endfor

        //duplicate/O root:anase1:$(w1),root:analysis:allTraces:$(w1)
    end

```

```

Function ButtonProc_3Delete(ctrlName) : ButtonControl

```

```

    String ctrlName
    dowindow/K w_sel
string wl=getsel()
    if(stringmatch(ctrlName,"FILTER"))
        wl=getselF()
    endif

    if(!strlen(wl))
        return 0
    endif

```

```

variable p1
    for(p1=0;p1<itemsinlist(wl,",");p1+=1)
        removefromgraph/Z/W=CANA#ME $(stringfromlist(p1,wl,","))
        killwaves/Z root:anase1:$(stringfromlist(p1,wl,","))
        killwaves/Z root:analysis:$(stringfromlist(p1,wl,","))
    endfor

```

```

        checklw()
//        updateCANAME()
wave/T lw=root:lw
    if(numpts(lw))
        getm("ba2")
        getm("ba4")
    endif
End

```

```

function calibrate()
    if(!checkWIN())
        return 0
    endif
    getmarquee/K/W=W_merge bottom
    if(!v_flag)
        return 0
    endif

```

```

variable l1=v_left
variable r1=v_right

        make/O/N=240 root:merge1=0
wave m1=root:merge1
        setdatafolder root:merge
string wl=wavelist("*",",","")
variable p1

        for(p1=0;p1<itemsinlist(wl,",");p1+=1)
wave curW=root:merge:$(stringfromlist(p1,wl,","))

            wavestats/Q/R=(l1,r1) curw
variable ma1=v_maxloc

```

```

        setscale/P x (-1*ma1+70),1, curw
        duplicate/O/R=(10,250) curw,root:temp44
wave t44=root:temp44
    m1+=t44

```

```

    endfor
    m1/=itemsinlist(wl,";")

    wavestats/Q/R=(l1,r1) m1
    ma1=v_maxloc
    setscale/P x (-1*ma1+70),1, m1
    setdatafolder root:
    setaxis bottom 10,250

```

```

end

```

```

#pragma rtGlobals=3          // Use modern global access method and strict wave access.

```

```

function SPONdoit(lasch):ButtonControl
string lasch

NVAR SPONl1
NVAR SPONr1
    calcSPON(SPONl1,SPONr1,lasch)

```

```

end

```

```

function calcSPON(l1,r1,lasch)
variable l1
variable r1
string lasch
    make/T/O/N=(1,8) Res_spon=""
    Res_spon[0][0]="trace"
    Res_spon[0][1]="basal"
    Res_spon[0][2]="SD"
    Res_spon[0][3]="events #"
    Res_spon[0][4]="trise (20-100)"
    Res_spon[0][5]="tdecay (100-20)"
    Res_spon[0][6]="tmax"
    Res_spon[0][7]="amplitude"

```

```

variable numw
string wl

```

```

    controlinfo/W=W_SA check0
        if(v_value)//selected only
wave selw
            wavestats/Q selw
            numw=v_sum
            wl=getSelected()
        else
            numw=countobjects("anasel",1)
            wl=wList("anasel",numw)
        endif

```

```

    //newdatafolder/O test33
variable p1
    for(p1=0;p1<numw;p1+=1)
wave curW=root:anasel:${stringfromlist(p1,wl,";")}

```



```

        anaSPON(curw,l1,r1,"")
    endfor

end

function anaSPON(curw,l1,r1, str1)
wave curw
variable l1
variable r1
string str1

    Duplicate/O/R=(l1,r1) curw,W_Smooth
wave w_smooth

controlinfo/W=w_sa check1
    if(v_value)//checked
NVAR SPONsmo
        smooth (SPONsmo), W_Smooth
    endif

wave/T rs=:res_spon//to rem the results
variable bas1=getBAS(w_smooth)
variable sd1=getSD(w_smooth)
variable curlItem=dimsize(rs,0)
    if(strlen(str1))
        bas1=str2num(stringfromlist(0,str1,";"))
        sd1=str2num(stringfromlist(1,str1,";"))
        curlItem=str2num(stringfromlist(2,str1,";"))
    endif

    wavestats/Q w_smooth
NVAR SPONthr
variable thr=(sponTHR)*sd1+bas1
//print thr, sponthr, sd1, bas1
    FindLevels/Q/D=destWave/EDGE=1 w_smooth, thr //increasing only
wave destWave //location of events

    if(!strlen(str1))
        insertpoints (dimsize(rs,0)),1, rs
        rs[curlItem][0]=nameofwave(curw) //trace name
    endif
    rs[curlItem][1]=num2str(bas1)//basal
    rs[curlItem][2]=num2str(sd1) //SD
    rs[curlItem][3]=num2str(numpts(destwave)) //number of events

variable p1
    for(p1=0;p1<numpts(destwave); p1+=1)
variable s1=destwave[p1]
        wavestats/Q/R=(s1,s1+5) w_smooth
variable max1=V_max
variable tmax1=v_maxloc

variable thr1=bas1+(max1-bas1)/5///20%
        findlevel/EDGE=1/R=(s1-5,s1+10)/Q w_smooth,thr1//5s before and 10s after signal recognition
variable tr1=v_levelX

variable thr2=max1-(max1-bas1)/5///80%
        findlevel/EDGE=1/R=(s1-5,s1+10)/Q w_smooth,thr2//5s before and 10s after signal recognition
variable tr2=v_levelX

        findlevel/EDGE=2/R=(tmax1,tmax1+60)/Q w_smooth,thr1//max-60s after maximum
variable tdec1=v_levelX

        rs[curlItem][4]=num2str(tr2-tr1)+"\r" //rise 2080
        rs[curlItem][5]=num2str(tdec1-tmax1)+"\r" //decay 100-20
        rs[curlItem][6]=num2str(tmax1)+"\r" //max
    endfor
endfunction

```

```

        rs[curItem][7]+=num2str(max1-bas1)+"\r" //amplitude
    endfor

    //killwaves/Z testdiff,td1,w_smooth,destwave

end

//OLD

variable nods=(r1-l1)/5//every 10 s
variable ints=(r1-l1)/5//5 s
    interpolate2/A=(nods)/N=(ints)/T=1/F=1/Y=testdiff w_smooth
wave testdiff
    duplicate/O testdiff, td1
wave td1
    differentiate/METH=2 td1
    wavestats/Q td1
    make/O/N=0 destwave
    FindLevels/Q/D=destWave/EDGE=1 td1, 3*v_sdev //increasing
wave destWave //location of events

function mergeSPON()
wave/T rs=res_spon

variable l1=20
variable r1=30
    if(!datafolderexists("root:merge"))
        newdatafolder root:merge
    endif
    newdatafolder/O root:merge:mergeSPON
    KillWindow w_SA#g4

    //dowindow/K w_mSPON
    display /HOST=w_SA/W=(0.69,0.15,0.99,0.55)
    ModifyGraph wbRGB=(52224,52224,52224),gbRGB=(56576,56576,56576)
    RenameWindow #,G4
    make/O/N=(l1+r1) mSPON1=0

NVAR l44=:SPONl1,r44=:SPONr1
variable p1=0,q1,s1=0
    for(p1=1;p1<dimsize(rs,0);p1+=1)
string tp=rs[p1][6]
        for(q1=0;q1<itemsinlist(tp,"\r");q1+=1)
variable t1=str2num(stringfromlist(q1,tp,"\r"))
            if( (t1-l1>l44)&&(t1+r1<r44))
                duplicate/O/R=(t1-l1,t1+r1)
root:merge:mergeSPON:$(rs[p1][0]+"_"+num2str(q1))
wave curSPON=root:merge:mergeSPON:$(rs[p1][0]+"_"+num2str(q1))
                controlinfo/W=W_SA check1
                if (V_value)//checked

NVAR smo1=:sponsmo

                    smooth (smo1), curSPON
                endif
                setscale/P x 0,1,curSPON
                wavestats/Q/R=[0,l1/2] curSPON
            endif
        endif
    endfor
    root:anasel:$(rs[p1][0]),

```

```

variable al1=v_avg
                                wavestats/Q/R=[l1+r1/2,l1+r1] curSPON
variable ar1=v_avg
                                curSPON-=(al1+ar1)/2
                                appendtograph/W=W_SA#g4 curSPON
                                ModifyGraph rgb($(nameofwave(curspon)))=(34816,34816,34816)
                                mSPON1+=curSPON
                                s1+=1
                                endif
                                endfor

                                endfor

                                mSPON1/=s1
                                //print s1
                                appendtograph/W=W_SA#g4 mSPON1
                                ModifyGraph/W=W_SA#g4 lsize(mSPON1)=2,rgb(mSPON1)=(0,0,0)
                                SetAxis/W=W_SA#g4 left -5,10
                                ModifyGraph/W=W_SA#g4 zero(left)=4,zeroThick(left)=2
                                Label/W=W_SA#g4 left "amplitude [AU]"
                                Label/W=W_SA#g4 bottom "time [s]"

end

```

```

Function ButtonProc_sponNEXLAS(ctrlName) : ButtonControl
    String ctrlName

    variable/G sponW
    wave/T rs=:res_spon
        if(stringmatch(ctrlName,"*1"))
            sponw-=1
        else
            sponw+=1
        endif
        if(sponw>(dimsize(rs,0)-1))
            sponw=1
        endif
        if(sponw<1)
            sponW=(dimsize(rs,0)-1)
        endif
        SPONdispltrace(sponW)
end

```

```

function SPONdispltrace(sponW)
    variable sponW
    wave/T rs=:res_spon

        TextBox/C/N=text0/W=w_sa "\\s(test) "+rs[sponw][0]
        duplicate/O/R=(0,280) root:anasel:${rs[sponw][0]},test
        controlinfo/W=W_SA check1
            if(v_value)//checked
NVAR SPONsmo
                smooth (SPONsmo), test
            endif
    variable bas=str2num(rs[sponw][1])

        setaxis left bas-20, bas+20
    variable sd1=str2num(rs[sponw][2])
    NVAR sponthr
        setdrawlayer /K userfront

```

```

        setdrawenv xcoord=prel,ycoord=left
        SetDrawEnv dash= 3
        drawline 0, bas+sponthr*sd1,1,bas+sponthr*sd1
        SetDrawEnv dash= 3,linethick= 2.00, xcoord=prel,ycoord=left
        drawline 0, bas,1,bas
variable p1
        setdrawenv xcoord=bottom,ycoord=prel,save
        for(p1=0;p1<str2num(rs[sponw][3]);p1+=1)
                drawline str2num(stringfromlist(p1,rs[sponw][6],"\r")),0,str2num(stringfromlist(p1,rs[sponw][6],"\r")),1
                //print stringfromlist(p1,rs[sponw][6]),sponW,rs[sponw][0]
        endfor

End

function getSD(test)
wave test
duplicate/O test, tempB
sort tempB,tempb

NVAR sponBAS
wavestats /Q/R=[0,(sponBAS/100)*numpts(tempB)] tempB
        return v_sdev
end

function getBAS(test)
wave test
duplicate/O test, tempB
sort tempB,tempb

NVAR sponBAS
wavestats /Q/R=[0,(sponBAS/100)*numpts(tempB)] tempB
        return v_avg
end

function SPONglob()
make/O test
variable/G SPONl1
variable/G SPONr1
variable/G SPONsmo
variable/G SPONbas
variable/G SPONthr

make/O SPONfreq,fit_SPONfreq
make/O SPONamp,fit_SPONamp
make/O SPONrise,fit_SPONrise
make/O SPONdec,fit_SPONdec
end

function f_sa()
        dowindow/K w_SA
        SPONglob()
        dowindow/F W_SA
        PauseUpdate; Silent 1 // building window...
//TAB2
wave test
        Display/K=1 /W=(501.75,88.25,1207.5,659)/N=W_SA test as "Spontaneous activity"
        ModifyGraph/W=W_SA margin(top)=142,margin(right)=235,wbRGB=(34816,34816,34816)
        //SetAxis/W=W_SA left -18.4001112874349,21.5998887125651
        TextBox/C/N=text0/Z=1/A=LT/X=0.00/Y=-25.00 "\s(test)"
        Button button0,pos={427,117},size={200,50},proc=ButtonProc_sponNEXLAS,title=">"
        Button button1,pos={90,117},size={200,50},proc=ButtonProc_sponNEXLAS,title="<"

```

```

Button button3,pos={635,200},size={300,40},proc=ButtonProc_sponsetBas,title="Set basal"
Button button4,pos={635,250},size={300,40},proc=ButtonProc_sponDEL,title="Delete\revent"
Button button5,pos={635,300},size={300,40},proc=ButtonProc_SPONcre,title="Create\new event"


TabControl tab0,pos={1,1},size={938,38},labelBack=(56576,56576,56576),fSize=30
TabControl tab0,tabLabel(0)="    Automatic    "
TabControl tab0,tabLabel(1)="    Manual    "
TabControl tab0,tabLabel(2)="    Statistics    ", value=0
TabControl tab0 proc=TabProc_fSA

//TAB1
GroupBox group0,pos={27,76},size={600,658},title="Analyze spontaneous activity"
GroupBox group0,labelBack=(47872,47872,47872),fSize=24
NVAR SPONl1
NVAR SPONr1
    GroupBox group2,pos={56,126},size={540,104},labelBack=(34816,34816,34816)
    GroupBox group2,fSize=18
    SetVariable setvar0,pos={92,150},size={191,28},bodyWidth=150,title="from"
    SetVariable setvar0,fSize=18,limits={-inf,inf,0}, value=sponL1, format="%g s"
    SetVariable setvar1,pos={379,150},size={169,28},bodyWidth=150,title="to"
    SetVariable setvar1,fSize=18,limits={-inf,inf,0}, value=sponR1, format="%g s"
    CheckBox check0,pos={215,192},size={197,24},title=" Selected traces only "
    CheckBox check0,fSize=18,value= 0


NVAR    SPONsmo
NVAR    SPONbas
NVAR    SPONthr
    GroupBox group1,pos={56,243},size={540,250},title="Parameters"
    GroupBox group1,labelBack=(34816,34816,34816),fSize=18
    CheckBox check1,pos={80,308},size={143,24},title=" Smooth traces",fSize=18
    CheckBox check1,value= 1
    SetVariable setvar2,pos={292,306},size={50,28},fSize=18,limits={9,0,1}, value=SPONsmo
    SetVariable setvar2 title=" "
    SetVariable setvar2 bodyWidth=40
    SetVariable setvar3,pos={77,363},size={266,28},title=" auto baseline",fSize=18
    SetVariable setvar3,limits={0,100,10}, value=SPONbas, format="%g %"
    SetVariable setvar4,pos={78,423},size={266,28},title="threshold ",fSize=18
    SetVariable setvar4,limits={0,100,1}, value=sponTHR, format="%g*SD"


Button button2,pos={65,522},size={529,62},title="Do it",fColor=(65280,0,0),proc=SPONdoit


//tab3
Button button6,pos={725,492},size={130,62},proc=SPOneditRes,title="Edit results"
Button button6,fColor=(0,6400,26112)
Button button7,pos={725,604},size={130,62},proc=SPONexpGraph,title="Export graphs"
Button button7,fColor=(0,13056,0)


//showStats()

TabProc_fSA("",0)

End


function showstats()

    getwindow/Z w_sa#g0, active

```

```

    if(!v_flag)
        return 0
    endif

wave SPONfreq,fit_SPONfreq
NVAR l1=SPONl1,r1=SPONr1
string t1=num2str(r1-l1)
    if(waveexists(SPONfreq))
        Display/W=(0.01,0.15,0.31,0.55)/HOST=# SPONfreq
        appendtograph fit_SPONfreq
        ModifyGraph wbRGB=(52224,52224,52224),gbRGB=(56576,56576,56576)
        ModifyGraph mode(SPONfreq)=5
        ModifyGraph hbFill(SPONfreq)=2,rgb(SPONfreq)=(34816,34816,34816);DelayUpdate
        ModifyGraph lsize(fit_SPONfreq)=2,rgb(fit_SPONfreq)=(0,0,0)
        ModifyGraph useBarStrokeRGB(SPONfreq)=1
        Setaxis bottom, 0,11
    else
        Display/W=(0.01,0.15,0.31,0.55)/HOST=#
    endif
    Label left "number of cells"
    Label bottom "frequency (events/" + t1 + " s)"
    RenameWindow #,G0
    SetActiveSubwindow ##

wave SPONamp,fit_SPONamp
    if(waveexists(SPONamp))
        Display/W=(0.35,0.15,0.65,0.55)/HOST=# SPONamp
        appendtograph fit_SPONamp
        ModifyGraph mode(SPONamp)=5,hbFill(SPONamp)=2,rgb(SPONamp)=(34816,34816,34816);DelayUpdate
        ModifyGraph useBarStrokeRGB(SPONamp)=1,lsize(fit_SPONamp)=2;DelayUpdate
        ModifyGraph rgb(fit_SPONamp)=(0,0,0)
        setaxis bottom 0,25
    else
        Display/W=(0.35,0.15,0.65,0.55)/HOST=#
    endif
    Label bottom "amplitude [AU]"
    Label left "number of events"
    ModifyGraph wbRGB=(52224,52224,52224),gbRGB=(56576,56576,56576)
    RenameWindow #,G1
    SetActiveSubwindow ##

wave SPONrise,fit_SPONrise
    if(waveexists(SPONrise))
        Display/W=(0.01,0.59,0.31,0.99)/HOST=# SPONrise
        appendtograph fit_SPONrise
        ModifyGraph mode(SPONrise)=5,hbFill(SPONrise)=2,rgb(SPONrise)=(34816,34816,34816);DelayUpdate
        ModifyGraph useBarStrokeRGB(SPONrise)=1,lsize(fit_SPONrise)=2;DelayUpdate
        ModifyGraph rgb(fit_SPONrise)=(0,0,0)
        setaxis bottom, 0,25
    else
        Display/W=(0.01,0.59,0.31,0.99)/HOST=#
    endif

    Label bottom "rise time [s]"
    Label left "number of events"
    ModifyGraph wbRGB=(52224,52224,52224),gbRGB=(56576,56576,56576)
    RenameWindow #,G2
    SetActiveSubwindow ##

wave SPONdec,fit_SPONdec
    if(waveexists(spondec))
        Display/W=(0.35,0.59,0.65,0.99)/HOST=# SPONdec
        appendtograph fit_SPONdec
    
```

```

        ModifyGraph mode(SPONdec)=5,hbFill(SPONdec)=2,rgb(SPONdec)=(34816,34816,34816);DelayUpdate
        ModifyGraph useBarStrokeRGB(SPONdec)=1,lsz(fit_SPONdec)=2;DelayUpdate
        ModifyGraph rgb(fit_SPONdec)=(0,0,0)
        setaxis bottom, 0,25
    else
        Display/W=(0.35,0.59,0.65,0.99)/HOST=#
    endif
    Label bottom "decay time [s]"
    Label left "number of events"
    ModifyGraph wbRGB=(52224,52224,52224),gbRGB=(56576,56576,56576)
    RenameWindow #,G3
    SetActiveSubwindow ##

    Display/W=(0.69,0.15,0.99,0.55)/HOST=#
    ModifyGraph wbRGB=(52224,52224,52224),gbRGB=(56576,56576,56576)
    RenameWindow #,G4
    SetActiveSubwindow ##

    updatestats()
end

```

```
Function TabProc_fSA(ctrlName,tabNum) : TabControl
```

```

    String ctrlName
    Variable tabNum
    setdrawlayer/K/W=w_SA userfront

```

```
//TAB1
```

```

    GroupBox group0,win=w_sa, disable=(tabnum!=0)
    GroupBox group2,win=w_sa, disable=(tabnum!=0)
    SetVariable setvar0,win=w_sa, disable=(tabnum!=0)
    SetVariable setvar1,win=w_sa, disable=(tabnum!=0)
    CheckBox check0,win=w_sa, disable=(tabnum!=0)
    GroupBox group1,win=w_sa, disable=(tabnum!=0)
    CheckBox check1,win=w_sa, disable=(tabnum!=0)
    SetVariable setvar2,win=w_sa, disable=(tabnum!=0)
    SetVariable setvar3,win=w_sa, disable=(tabnum!=0)
    SetVariable setvar4,win=w_sa, disable=(tabnum!=0)

    Button button2,win=w_sa, disable=(tabnum!=0)

```

```
//TAB2
```

```

    Button button0,win=w_sa, disable=(tabnum!=1)
    Button button1,win=w_sa, disable=(tabnum!=1)
    Button button3,win=w_sa, disable=(tabnum!=1)
    Button button4,win=w_sa, disable=(tabnum!=1)
    Button button5,win=w_sa, disable=(tabnum!=1)

```

```

    if((tabnum!=1))
        textbox/K/N=text0
        modifygraph margin=1000
    else

```

```
nvar sponw
```

```
wave/T rs=res_spon
```

```

    ModifyGraph/W=w_sa margin(top)=142,margin(right)=235, margin(left)=0, margin(bottom)=0
    TextBox/C/N=text0/W=w_sa/Z=1/A=LT/X=0.00/Y=-25.00 "\\s(test) "+rs[sponw][0]

```

```
endif
```

```
//TAB3
```

```

        Button button6,win=w_sa, disable=(tabnum!=2)
        Button button7,win=w_sa, disable=(tabnum!=2)
        if((tabnum!=2))
            hidestats()
        else
            setactivesubwindow w_sa
            showstats()
        endif

        return 0
    End

```

```

function hidestats()
    getwindow/Z W_SA#G0,active
    if(!V_flag)
        killwindow W_SA#G0
    endif
    getwindow/Z W_SA#G1,active
    if(!V_flag)
        killwindow W_SA#G1
    endif
    getwindow/Z W_SA#G2,active
    if(!V_flag)
        killwindow W_SA#G2
    endif
    getwindow/Z W_SA#G3,active
    if(!V_flag)
        killwindow W_SA#G3
    endif
    getwindow/Z W_SA#G4,active
    if(!V_flag)
        killwindow W_SA#G4
    endif
end

```

```

function ButtonProc_fSPON(lasch):ButtonControl
string lasch

```

```

variable/G SPONl1
variable/G SPONr1
getmarquee/Z/K/W=CANA#ME bottom
    if(V_flag)
        SPONl1=v_left
        SPONr1=v_right
    else
        SPONl1=0
        SPONr1=300
    endif
variable/G SPONsmo=1
variable/G SPONbas=80
variable/G SPONthr=3

```

```

        f_sa()

    end

```



```

Function ButtonProc_sponsetBas(ctrlName) : ButtonControl
    String ctrlName
    getmarquee/Z/K/W=W_SA bottom
    if(V_flag)
variable    l1=v_left
variable    r1=v_right
    else
        return 0
    endif
wave test
    wavestats /Q/R=[l1,r1] test

NVAR sponW
NVAR sponthr
wave/T res_spon
    res_spon[sponw][1]=num2str(v_avg)
    res_spon[sponw][2]=num2str(v_sdev)

NVAR sponl1,sponr1
    anaSPON(test,sponl1,sponr1, res_spon[sponw][1]+",""+res_spon[sponw][2]+",""+num2str(sponw))
    SPONdispltrace(sponW)

End

```

```

Function ButtonProc_sponDEL(ctrlName) : ButtonControl
    String ctrlName
    getmarquee/Z/K/W=W_SA bottom
    if(V_flag)
variable    l1=v_left
variable    r1=v_right
    else
        return 0
    endif
NVAR sponW
wave/T rs=:res_spon

string curR=rs[sponw][4]
string curD=rs[sponw][5]
string curT=rs[sponw][6]
string curA=rs[sponw][7]
string rem=""
    rs[sponw][4]=""
    rs[sponw][5]=""
    rs[sponw][6]=""
    rs[sponw][7]=""
variable p1
    for(p1=0;p1<itemsinlist(curT,"\r");p1+=1)
variable v1=str2num(stringfromlist(p1,curT,"\r" ))
        if((v1<l1) || (v1>r1)) //event out of marquee
            rs[sponw][4]+=stringfromlist(p1,curR,"\r")+""\r"
            rs[sponw][5]+=stringfromlist(p1,curD,"\r")+""\r"
            rs[sponw][6]+=stringfromlist(p1,curT,"\r")+""\r"
            rs[sponw][7]+=stringfromlist(p1,curA,"\r")+""\r"

        endif

    endfor

    rs[sponw][3]=num2str(itemsinlist(rs[sponw][4],"\r"))
    SPONdispltrace(sponW)

```

End

```
Function ButtonProc_SPONcre(ctrlName) : ButtonControl
    String ctrlName
    getmarquee/Z/K/W=W_SA bottom
    if(V_flag)
        variable l1=v_left
        variable r1=v_right
        else
            return 0
        endif

    wave test
        wavestats/Q/R=(l1,r1) test
    variable maxl1=v_maxloc
    variable max1=V_max

    NVAR sponW
    wave/T rs=:res_spon
    variable bas1=str2num(rs[sponw][1])

    variable thr20=bas1+(max1-bas1)/5///20%
        findlevel/EDGE=1/R=(l1,maxl1)/Q test,thr20//5s before and 10s after signal recognition
    variable tr1=v_levelX

    variable thr80=max1-(max1-bas1)/5///80%
        findlevel/EDGE=1/R=(l1, maxl1)/Q test,thr80//5s before and 10s after signal recognition
    variable tr2=v_levelX

        findlevel/EDGE=2/R=(maxl1,r1)/Q test,thr20//max-60s after maximum
    variable tdec1=v_levelX

        rs[sponw][4]+=num2str(tr2-tr1)+"\r" //rise 2080
        rs[sponw][5]+=num2str(tdec1-maxl1)+"\r" //decay 100-20
        rs[sponw][6]+=num2str(maxl1)+"\r" //max
        rs[sponw][7]+=num2str(max1-bas1)+"\r" //amplitude

        rs[sponw][3]=num2str(itemsinlist(rs[sponw][7],"\r"))

    SPONdispltrace(sponW)
```

End

```
function updatestats()

    wave/T rs=:res_spon
        dowindow/F W_SA
        SetActiveSubwindow w_sa

    //frequency
        make/O/N=(dimsize(rs,0)) temp22
        temp22=str2num(rs[p][3])
        Make/N=11/O SPONfreq
        Histogram/B={0,1,11} temp22,SPONfreq
        CurveFit/NTHR=0/Q exp SPONfreq /D

    wave w_coef
    wave w_sigma
    variable sig1=round(100*abs(1/(w_coef[2]+w_sigma[2])-1/w_coef[2]))/100

    string/G v_freq=num2str(round(100/w_coef[2])/100)+" ± "+num2str(sig1)
```

```

setdrawlayer/K/W=W_SA#G0 userfront
DrawText/W=W_SA#G0 0.4,0.3,v_freq

//amplitude
    SPON_Tval("temp23", 7)
wave temp23
Make/N=50/O SPONamp
    Histogram/B={0,0.5,50} temp23,SPONamp
    CurveFit/NTHR=0/Q gauss SPONamp /D
string/G v_amp=num2str(round(100*w_coef[2])/100)+" ± "+num2str(round(100*w_sigma[2])/100)
    setdrawlayer/K/W=W_SA#G1 userfront
    DrawText/W=W_SA#G1 0.4,0.3,v_amp

//merge
    mergeSPON()

///trise
    SPON_Tval("temp24", 4)
wave temp24
    Make/N=25/O SPONrise
    Histogram/B={0,1,25} temp24,SPONrise
    CurveFit/NTHR=0/Q gauss SPONrise /D
wave w_coef
wave w_sigma
variable sig2=round(100*abs(1/(w_coef[2]+w_sigma[2])-1/w_coef[2]))/100

string/G v_rise=num2str(round(100*w_coef[2])/100)+" ± "+num2str(sig2)+" s"
    setdrawlayer/K/W=W_SA#G2 userfront
    DrawText/W=W_SA#G2 0.4,0.3,v_rise

///tdec
    SPON_Tval("temp25", 5)
wave temp25
    Make/N=25/O SPONdec
    Histogram/B={-1,2,25} temp25,SPONdec
    CurveFit/NTHR=0/Q gauss SPONdec[2,25] /D
wave w_coef
wave w_sigma
variable sig3=round(100*abs(1/(w_coef[2]+w_sigma[2])-1/w_coef[2]))/100

string/G v_dec=num2str(round(100*w_coef[2])/100)+" ± "+num2str(sig3)+" s"
    setdrawlayer/K/W=W_SA#G3 userfront
    DrawText/W=W_SA#G3 0.4,0.3,v_dec

    setdrawlayer/K/W=w_SA userfront
    setdrawenv/W=W_SA xcoord=rel,ycoord=rel, fstyle= 1,save
    DrawText/W=W_SA 0.25,0.264,"Experiments: " +num2str(itemsinlist(getexp()))

    DrawText/W=W_SA 0.25,0.272,"Cells: " + num2str ( (dimsize(rs,0)-1) )
wavestats/Q SPONamp
    DrawText/W=W_SA 0.25,0.280,"Events: " +num2str(v_sum)

//xx cells, xx events"

killwaves/Z temp22, temp23,temp24, temp25
end

```

```

function SPON_Tval(str1,var)
string str1 //wavename
variable var

    make/O/N=0 $(str1)
wave w1=$(str1)
wave/T rs=:res_spon

variable p1, q1
    for(p1=0;p1<dimsize(rs,0);p1+=1)
        if(strlen(rs[p1]))
string curstr=rs[p1][var]
                for(q1=0;q1<itemsinlist(curstr,"\r");q1+=1)
                    insertpoints (numpts(w1)), 1, w1
                    w1[numpts(w1)-1]=str2num(stringfromlist(q1,curstr,"\r"))
                endfor
            endif
        endfor

end

```

```

Make/N=11/O SPONfreq
Make/O fit_SPONfreq
string/G v_freq
Make/N=50/O SPONamp
Make/O fit_SPONamp
string/G v_amp

```

```

Function SPoneditRes(ctrlName) : ButtonControl
    String ctrlName

```

```

string exp1=getEXP()

```

```

    make/O/N=(itemsinlist(exp1,";"),8)/T SPONres=""
    SetDimLabel 1,0,Experiment,ponres
    SetDimLabel 1,1,Number_Cells,ponres
    SetDimLabel 1,2,Perc_Responding_Cells,ponres
    SetDimLabel 1,3,Avg_number_Events_perCell,ponres
    SetDimLabel 1,4,Avg_Amplitude,ponres
    SetDimLabel 1,5,Avg_AMP_ATPnorm,ponres
    SetDimLabel 1,6,Avg_riseT,ponres
    SetDimLabel 1,7,Avg_decayT,ponres
    ponres[][0]=stringfromlist(p,exp1,";")

```

```

variable p1
    for(p1=0;p1<itemsinlist(exp1,";");p1+=1)
string r1=SPONgetexpEV(stringfromlist(p1,exp1,";"))
        ponres[p1][1]= stringfromlist(0, r1) //number of cells
        ponres[p1][2]= stringfromlist(2,r1)//% responding
        ponres[p1][3]= stringfromlist(1,r1 ) //avg number of events

string r2
        ponres[p1][4]= stringfromlist(0,SPONgetexpAMP(stringfromlist(p1,exp1,";")) ) //avg amplituds of events
        ponres[p1][5]= stringfromlist(1,SPONgetexpAMP(stringfromlist(p1,exp1,";")) ) //avg amplituds of events

        ponres[p1][6]= stringfromlist(2,SPONgetexpAMP(stringfromlist(p1,exp1,";")) ) //avg rise times

```

```

        sponres[p1][7]= stringfromlist(3,SPONgetexpAMP(stringfromlist(p1,exp1,";")) )           //avg decay times
    endfor

    dowindow/K SPONexp
    Edit/N=SPONexp/W=(36.75,249.5,970.5,736.25)/K=1  sponres.Id as "Summary spontaneous Ca2+ elevations"
    ModifyTable format(Point)=1,width(Point)=0,width(SPONres.l)=20,alignment(SPONres.d)=1
    ModifyTable width(SPONres.d)=150
End

```

```

function/S SPONgetexpEV(str1)
string str1 //exp
wave/T rs=res_SPON
variable expo,n1
variable p1
variable c1//count
    for(p1=0;p1<dimsize(rs,0);p1+=1)
        if(stringmatch(rs[p1][0],str1+"*"))
            expo+=str2num(rs[p1][3])
            n1+=1
            if(str2num(rs[p1][3]))
                c1+=1//responding cells
            endif
        endif
    endfor

    return num2str(n1)+";"+num2str(expo/n1)+";"+num2str(100*c1/n1)
end

```

```

function/S SPONgetexpAMP(str1)
string str1 //exp

wave/T rs=res_SPON
variable expo,n1, expo1,expo2,expo3,n2,n3
variable p1,q1
    for(p1=0;p1<dimsize(rs,0);p1+=1)
        variable ATPa=getATPamp(str1)
        if(stringmatch(rs[p1][0],str1+"*"))
            for(q1=0;q1<itemsinlist((rs[p1][7]));q1+=1)
                expo+=str2num(stringfromlist(q1,rs[p1][7],"\\r"))

                expo1+=100*str2num(stringfromlist(q1,rs[p1][7],"\\r"))/ATPa

                if(!numtype(str2num(stringfromlist(q1,rs[p1][4],"\\r"))))
                    expo2+=str2num(stringfromlist(q1,rs[p1][4],"\\r"))
                    n2+=1
                endif
                if(!numtype(str2num(stringfromlist(q1,rs[p1][5],"\\r"))))
                    expo3+=str2num(stringfromlist(q1,rs[p1][5],"\\r"))
                    n3+=1
                endif
            endfor
            n1+=1
        endif
    endfor

    return num2str(expo/n1)+";"+num2str(expo1/n1)+";"+num2str(expo2/n2)+";"+num2str(expo3/n3)
end

```

end

function getATPamp(str1)

string str1//cell

wave/T lw=root:analysis:\${"bas"+num2str(2)+"_cell"}

wave appATP=root:analysis:\${"app"+num2str(2)+"_top10"}//given that App2==ATP

wave basATP=root:analysis:\${"bas"+num2str(2)+"_avg"}//given that App2==ATP

variable p1, i1

for(p1=0;p1<numpts(lw);p1+=1)

if(stringmatch(lw[p1],str1))

i1=p1

endif

endfor

return (appATP[i1]-basATP[i1])

end

Function SPONexpGraph(ctrlName) : ButtonControl

String ctrlName

SavePICT/EF=1/E=-3

End

function fromPrev(lasch)

string lasch

controlinfo/W=canasel popup0

string curE=s_value

string prevE=stringfromlist(v_value-1-1,gettext(),",")

getvaluesfromdf(prevE,lasch) //copies values to root:

SetVarProc_1("",1,"", "")

SetVarProc_Appli1("",0,"", "")

end

Function ButtonProc_showZIMG(ctrlName) : ButtonControl

String ctrlName

dowindow/K W_curlIMG

controlinfo/W=canasel popup0

string df=s_value

wave curlIMG=root:\${df}:ZIMG

if(waveexists(curlimg))

fW_curlIMG()//opens window

else

sRD(df)

endif

End

function crop(lasch):buttoncontrol

string lasch

```

wave zimg=root:$(getPicN()):zimg

tempdata()
wave xrw,yrw
imageboundarytomask width=dimsize(zimg,0), height=dimsize(zimg,1), xwave=xrw,ywave=yrw,seedx=xrw[1],seedy=yrw[1]
wave MR=M_roidmask
if(stringmatch(lasch,"*crop"))
    MR=(MR[p][q]==0)
endif

```

```

wave/T wC=root:$(getpicN()):$("roiCOOR")

```

```

variable p1

```

```

for(p1=0; p1<numpts(WC); p1+=1)
string curC=wc[p1]//current roi in df
string curCoord=stringfromlist(0,curc,";")//first set of points in roi
variable x1=str2num(stringfromlist(0,curcoord," "))
variable y1=str2num(stringfromlist(1,curcoord," "))

```

```

    if(!MR[x1][y1])//roi outside
        Markdel(p1)
    else //roi inside region

    endif

```

```

endfor
deletemarked()

```

```

removefromgraph/W=W_curing/Z xrw,yrw
checkbox check2, win=w_curIMG, value=0
killwaves /Z MR,xrw,yrw

```

```

drawallrois(0)

```

```

end

```

```

function deletemarked()
wave ra=root:roiArea
wave/T rbc=root:roibcoor
wave/T rbm=root:roibmean
wave/T rc=root:roicoor
wave/T rl=root:roiLoc
wave/T rm=root:roimean
wave/T rn=root:roinum
wave r0=root:roires_0
wave r1=root:roires_1
wave r2=root:roires_2
wave r3=root:roires_3
wave r4=root:roires_4
wave r5=root:roires_5

```

```

variable p1=0
do
    if(strlen(rc[p1]))//not delete
        p1+=1
    else
        deletepoints p1,1,rbc,rbm,rc,rl,rm,rn
        deletepoints p1,1,r0,r1,r2,r3,r4,r5,ra
    endif

```

```

        while(p1<numpnts(rc))

end

function markdel(var)
variable var
wave ra=root:roiArea
wave/T rbc=root:roibcoor
wave/T rbm=root:roibmean
wave/T rc=root:roicoor
wave/T rl=root:roiLoc
wave/T rm=root:roimean
wave/T rn=root:roinum
wave r0=root:roires_0
wave r1=root:roires_1
wave r2=root:roires_2
wave r3=root:roires_3
wave r4=root:roires_4
wave r5=root:roires_5

        ra[var]=nan
        rbc[var]=""
        rbm[var]=""
        rc[var]=""
        rl[var]=""
        rm[var]=""
        rn[var]=""
        r0[var]=nan
        r1[var]=nan
        r2[var]=nan
        r3[var]=nan
        r4[var]=nan
        r5[var]=nan

end

function tempdata()

wave ra=root:$(getPicN()):roiArea
wave/T rbc=root:$(getPicN()):roibcoor
wave/T rbm=root:$(getPicN()):roibmean
wave/T rc=root:$(getPicN()):roicoor
wave/T rl=root:$(getPicN()):roiLoc
wave/T rm=root:$(getPicN()):roimean
wave/T rn=root:$(getPicN()):roinum
wave r0=root:$(getPicN()):roires_0
wave r1=root:$(getPicN()):roires_1
wave r2=root:$(getPicN()):roires_2
wave r3=root:$(getPicN()):roires_3
wave r4=root:$(getPicN()):roires_4
wave r5=root:$(getPicN()):roires_5

        duplicate/O ra, root:roiArea
        duplicate/O rbc,root:roibcoor
        duplicate/O rbm,root:roibmean
        duplicate/O rc,root:roicoor
        duplicate/O rl,root:roiLoc
        duplicate/O rm,root:roimean
        duplicate/O rn,root:roinum
        duplicate/O r0,root:roires_0
        duplicate/O r1,root:roires_1
        duplicate/O r2,root:roires_2

```



```

        duplicate/O r3,root:roires_3
        duplicate/O r4,root:roires_4
        duplicate/O r5,root:roires_5
end

```

```

function /S getPicN()
    controlinfo/W=canasel popup0
    return s_value
end

```

```

function drawallrois(var1)
variable var1

```

```

    string picN=getPicN()
    if(var1)///df
    wave/T wC=root:${picN}:${"roiCOOR"}
    else //root:
    wave/T wC=root:${"roiCOOR"}
    endif
    variable p1

    make/O/N=0 xA,yA
    for (p1=0; p1<numpts(wC);p1+=1)

        makeCoordWfromStr(p1, var1)

    wave xw,yw

        concatenate/NP {xw},xA
        concatenate/NP {yw},yA

        insertpoints (numpts(xA)),1, xA,yA
        xA[numpts(xA)-1]=NaN
        yA[numpts(xA)-1]=NaN

    endfor
    setdrawlayer/W=W_curing/K progfront
    setdrawenv/W=W_curing xcoord=top,ycoord=left,linefgc=(65000,65000,0), linethick=1, fillpat=0,save
    drawpoly/W=W_curing/ABS 0,0,1,1,xA,yA

    controlinfo/W=fval check01
end

```

```

function makeCoordWfromStr(num,var1)
variable num
variable var1

    make/O/N=0 root:xW,root:yW
    wave xw=root:xw,yw=root:yw
    string picN=getPicN()
    if(var1)///df
    wave/T wC=root:${picN}:${"roiCOOR"}
    else //root:
    wave/T wC=root:${"roiCOOR"}
    endif
    string curC=wc[num]

    variable p1
    for(p1=0; p1<itemsinlist(curC,""); p1+=1)
    string curCoord=stringfromlist(p1,curC,"")
        insertpoints p1, 1, xw,yw
        xw[p1]=str2num(stringfromlist(0,curcoord,""))
        yw[p1]= str2num(stringfromlist(1,curcoord,""))
    endfor

```

end

```
function CheckProc_2selREG(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked
    if(checked)
        make/O/N=0 xrw,yrw
        graphWavedraw/O/L/T/W=W_curimg yrW,xrW
    else
        removefromgraph/Z/W=W_curimg xrw,yrw
    endif
```

End

```
function fW_curlIMG()
dowindow/K W_curimg
    PauseUpdate; Silent 1 // building window...
    Display/N=W_curimg /W=(58.5,64.25,826.5,832.25)/K=1 as getpicn()
    AppendImage/T root:${getpicn()}:zIMG
    ModifyImage zIMG ctab= {*,*,Green,0}
    ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1,wbRGB=(39936,39936,39936)
    ModifyGraph gbRGB=(39936,39936,39936)
    ModifyGraph mirror=0
    ModifyGraph nticks=0
    ModifyGraph noLabel=2
    ModifyGraph standoff=0
    ModifyGraph axThick=0
    ModifyGraph axisEnab(left)={0,0.9}
    SetAxis/A/R left

    CheckBox check2,pos={157,50},size={80,14},proc=CheckProc_2selREG,title="Draw Region"
    CheckBox check2,value= 0
    Button button0,pos={29,27},size={70,20},proc=ButtonProc_restore,title="Show ROIs"
    Button button1,pos={29,65},size={30,20},proc=ButtonProc_restoremc,title="ROIs"
    Button button1,fColor=(0,13056,0)
    Button button2,pos={69,65},size={30,20},proc=ButtonProc_restoremc,title="ROIs"
    Button button2,fColor=(39168,13056,0)

    GroupBox group0,pos={128,12},size={476,75},title="delete ROIs"
    Button button1crop,pos={274,36},size={70,40},proc=crop,title="Crop"
    Button button1delete,pos={350,36},size={70,40},proc=crop,title="Delete"
    Button button3,pos={931,32},size={70,40},title="Store",proc=ButtonProc_roistore
    Button button1delete1,pos={500,36},size={70,40},proc=ButtonProc_restore,title="Restore"

    Button button4,pos={638,47},size={70,20},proc=ButtonProc_appendMC,title="mC IMG"
    Button button4,fColor=(39168,13056,0)
```

End

```
Function ButtonProc_restore(ctrlName) : ButtonControl
    String ctrlName
    drawallrois(1)
End
```

```
Function ButtonProc_roistore(ctrlName) : ButtonControl
    String ctrlName
    wave ra=root:roiArea
```

```

wave/T rbc=root:roibcoor
wave/T rbm=root:roibmean
wave/T rc=root:roicoor
wave/T rl=root:roiLoc
wave/T rm=root:roimean
wave/T rn=root:roinum
wave r0=root:roires_0
wave r1=root:roires_1
wave r2=root:roires_2
wave r3=root:roires_3
wave r4=root:roires_4
wave r5=root:roires_5

```

```

duplicate/O ra, root:${getpicn()}:roiArea
duplicate/O rbc, root:${getpicn()}:roibcoor
duplicate/O rbm, root:${getpicn()}:roibmean
duplicate/O rc, root:${getpicn()}:roicoor
duplicate/O rl, root:${getpicn()}:roiLoc
duplicate/O rm, root:${getpicn()}:roimean
duplicate/O rn, root:${getpicn()}:roinum
duplicate/O r0, root:${getpicn()}:roires_0
duplicate/O r1, root:${getpicn()}:roires_1
duplicate/O r2, root:${getpicn()}:roires_2
duplicate/O r3, root:${getpicn()}:roires_3
duplicate/O r4, root:${getpicn()}:roires_4
duplicate/O r5, root:${getpicn()}:roires_5

```

```

PopupMenuProc_1("", 1, getpicn())

```

```

killwaves/Z r1,rbc,rbm,rc,rl,rm,rn,r0,r1,r2,r3,r4,r5

```

End

```

Function CheckProc_2actFilter(ctrlName, checked) : CheckBoxControl

```

```

String ctrlName
Variable checked

```

```

groupbox $("group"+ctrlName[5]), win=W_filter, disable=2*(checked==0)
SVAR f1=root:analysis:filters
string ol=stringfromlist(str2num(ctrlName[5]), f1, "r")
string nl=stringfromlist(0, ol, ".")+"."+num2str(checked)+":"+stringfromlist(2, ol, ".")

```

```

f1= replacestring(ol, f1, nl)

```

```

if(!strlen(ol))/newfilter
    f1+="locA:"+num2str(checked)+":\r"
endif

```

```

if(str2num(ctrlName[5])==0)//age
    GroupBox group0, win=W_filter, disable=2*(checked==0)
    SetVariable setvar0, win=W_filter, disable=2*(checked==0)
    SetVariable setvar1, win=W_filter, disable=2*(checked==0)
endif
if(str2num(ctrlName[5])==1)//sex
    GroupBox group1, win=W_filter, disable=2*(checked==0)
    CheckBox check5, win=W_filter, disable=2*(checked==0)
    CheckBox check6, win=W_filter, disable=2*(checked==0)
endif
if(str2num(ctrlName[5])==2)//region
    GroupBox group2, win=W_filter, disable=2*(checked==0)
    ListBox list0, win=W_filter, disable=2*(checked==0)
endif
if(str2num(ctrlName[5])==3)//genotype
    GroupBox group3, win=W_filter, disable=2*(checked==0)

```

```

        ListBox list1, win=W_filter, disable=2*(checked==0)
    endif
    if(str2num(ctrlName[5])==4)//application
        GroupBox group4,win=W_filter, disable=2*(checked==0)
        ListBox list2, win=W_filter, disable=2*(checked==0)
    endif
    if(str2num(ctrlName[5])==5)//location relative
        GroupBox group5,win=W_filter, disable=2*(checked==0)
        PopupMenu popup0,win=W_filter, disable=2*(checked==0)
        PopupMenu popup1,win=W_filter, disable=2*(checked==0)
        SetVariable setvar2,win=W_filter, disable=2*(checked==0)
        SetVariable setvar3,win=W_filter, disable=2*(checked==0)
        Button button0, win=W_filter, disable=2*(checked==0)
    endif
    if(str2num(ctrlName[5])==6)//location relative
        GroupBox group6, disable=2*(checked==0),win=W_filter
        PopupMenu popup2, disable=2*(checked==0),win=W_filter
        SetVariable setvar4, disable=2*(checked==0),win=W_filter
        CheckBox check7, disable=2*(checked==0),win=W_filter
        SetVariable setvar6, disable=2*(checked==0),win=W_filter
    endif

    //applyfilters()

End

Function ButtonProc_applyfilter(ctrlName) : ButtonControl
    String ctrlName
    applyfilters()
End

Function CheckProc_switchunit(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked

    if(checked)
        SetVariable setvar4 format="%g µm"
        setvariable setvar6 disable=0
    else
        SetVariable setvar4 format="%g px"
        setvariable setvar6 disable=1
    endif

End

Function ListBoxProc_2(ctrlName,row,col,event) : ListBoxControl
    String ctrlName
    Variable row
    Variable col
    Variable event //1=mouse down, 2=up, 3=dbl click, 4=cell select with mouse or keys
                  //5=cell select with shift key, 6=begin edit, 7=end

    if(event==4)
        //applyfilters()
    endif
    return 0

End

function filteredYES(str1)

```

```

string str1 //wavename

wave curW=root:anasel:${str1}

SVAR f1=root:analysis:filters

//print note(curw)
if(str2num(stringfromlist(1,stringfromlist(0, f1,"r"),":")))//apply age filter
string age=stringbykey("age",stringfromlist(2,note(curW),":"),":","r")
variable ag1=-1

//print "age", age
if(strlen(age))
    ag1=str2num(age)
endif

NVAR a1=root:analysis:F_age1
NVAR a2=root:analysis:F_age2
//print ag1,a1,a2, numtype(ag1)
//condition
//
if((ag1<a1) || (ag1>a2))
    print ag1,a1,a2,stringfromlist(p1,lw,":"),age,"J"
    return 0 //not included
endif
if(numtype(ag1)) //age not specified or unreadable
    return 0 //not included
endif
endif

if(str2num(stringfromlist(1,stringfromlist(1, f1,"r"),":")))//apply sex filter
//print "sex"
string sex=stringbykey("sex",stringfromlist(2,note(curW),":"),":","r")
//print nameofwave(curw), sex
variable match
variable fmatch=str2num(stringfromlist(2,stringfromlist(1, f1,"r"),":"))
if((fmatch<3))// sex filter
    if(stringmatch(sex, "Male"))
        match=1
    elseif (stringmatch(sex, "Female"))
        match=2
    endif
endif
//condition
//print p1, match, fmatch
if((match!=fmatch))
    return 0 //not included
endif
endif
endif

if(str2num(stringfromlist(1,stringfromlist(2, f1,"r"),":")))//apply region filter
//print "reg"
string reg=stringbykey("region",stringfromlist(2,note(curW),":"),":","r")
string rl=getfiltersel("reg")
if(strlen(rl))
//condition
if(!stringmatch(rl,"*" + reg + "*"))
    return 0 //not included
endif
endif
endif

if(str2num(stringfromlist(1,stringfromlist(3, f1,"r"),":")))//apply genotype filter
//print "gen"
string geno=stringbykey("genotype",stringfromlist(2,note(curW),":"),":","r")
string gl=getfiltersel("geno")
if(strlen(gl))
//condition

```

```

                                if(!stringmatch(gl,"*" + geno + "*" ))
                                    return 0 //not included
                                endif
                            endif
                        endif

                        if(str2num(stringfromlist(1,stringfromlist(4, f1,"\\r"),":.")))//apply app filter
                            //print "app"

string al=getfilterse("app")
//print al

                                if(strlen(al))

                                    //condition
string app=stringbykey("app",stringfromlist(2,note(curw),",",".", "\\r")
variable q1,r1, rem1

                                for(q1=0;q1<itemsinlist(al, ",");q1+=1)
                                    rem1=0
                                    for(r1=0;r1<6;r1+=1)

string cw=stringfromlist(r1,app,"\\t")
//print cw,"!", app

                                if((strlen(cw))&&(stringmatch(al,"*" + cw + "*" )))
                                    rem1+=1
                                endif
                            endfor

                                if(!rem1)
                                    return 0 //not included
                                endif
                            endfor
                        endif
                    endif

                        if(str2num(stringfromlist(1,stringfromlist(5, f1,"\\r"),":.")))//apply loc filter
                            //print "loc",stringfromlist(5, f1,"\\r")
                                controlinfo/w=w_filter popup0

string poi1=s_value

                                controlinfo/w=w_filter popup1

string poi2=s_value
string info=note(curw)

//cell location
variable cx=str2num(stringfromlist(0, findloc(str1),","))
variable cy=str2num(stringfromlist(1, findloc(str1),","))

SVAR/Z poi=root:${stringfromlist(0,info,":")}:points
                                if(SVAR_Exists(poi))
                                    // print lw[p1]+"\\r"+poi
variable p1x=str2num(stringfromlist(0,stringbykey(poi1,poi,":",".", "\\r"),":."))
variable p1y=str2num(stringfromlist(1,stringbykey(poi1,poi,":",".", "\\r"),":."))
variable p2x=str2num(stringfromlist(0,stringbykey(poi2,poi,":",".", "\\r"),":."))
variable p2y=str2num(stringfromlist(1,stringbykey(poi2,poi,":",".", "\\r"),":."))

variable d100=sqrt((p1x-p2x)^2+(p1y-p2y)^2)//distance pipettes
variable dp=sqrt((p1x-cx)^2+(p1y-cy)^2)//distance point1-cell
variable drel=100*dp/d100
                                //t1[p1]=dp
                                //t2[p1]=drel

//print drel
NVAR f_dist1=root:analysis:f_dist1
NVAR f_dist2=root:analysis:f_dist2

                                //print p1x,p1y,cx,cy,d100,nameofwave(curw),drel,f_dist1, f_dist2
                                if((drel>f_dist1)&&(drel<f_dist2))//keep
                                    else

```

```

                                return 0 //not included
                                endif
                                else
                                return 0 //not included
                                endif

                                else
                                //print lw[p1],"NOT"
                                endif

                                if(str2num(stringfromlist(1,stringfromlist(6, f1, "\r"),":")))//apply loc abs filter
                                //print "loc",stringfromlist(5, f1, "\r")
                                controlinfo/w=w_filter popup2

                                string poi3=s_value

                                string info1=note(curw)

                                //cell location relative
                                variable cx1=str2num(stringfromlist(0, findloc(str1),","))
                                variable cy1=str2num(stringfromlist(1, findloc(str1),","))

                                SVAR/Z poi_=root:(stringfromlist(0,info1,";")):points
                                if(SVAR_Exists(poi_))
                                // print lw[p1)+"\r"+poi
                                variable p3x=str2num(stringfromlist(0,stringbykey(poi3,poi_,";\r"),";"))
                                variable p3y=str2num(stringfromlist(1,stringbykey(poi3,poi_,";\r"),";"))

                                variable dpA=sqrt((p3x-cx1)^2+(p3y-cy1)^2)//distance point3-cell

                                //print drel
                                variable distT
                                NVAR f_distA=root:analysis:f_distA
                                NVAR f_distF=root:analysis:f_distF

                                controlinfo/W=w_filter check7
                                if(v_value)//µm
                                distT=f_distA/f_distF
                                else
                                distT=f_distA
                                endif
                                //print p1x,p1y,cx,cy,d100,nameofwave(curw),drel,f_dist1, f_dist2
                                if((dpA>distT)) //keep
                                else
                                return 0 //not included
                                endif
                                else
                                return 0 //not included
                                endif

                                else
                                //print lw[p1],"NOT"
                                endif

                                //duplicate/O fw1,root:selw
                                //end
                                return 1
                                end

```

```

function calcIT(l1,r1,lasch)
variable l1
variable r1
string lasch

variable numw
string wl

//      controlinfo/W=cana check3
//      if(v_value)//selected only
//wave selw
//      wavestats/Q selw
//      numw=v_sum
//      wl=getSelected()
//      else

      numw=countobjects("anasel",1)
      wl=wList("anasel",numw)

//      endif

      make/O/N=(0)/T temp42=""
      make/O/N=(0) temp43=nan
      make/O/N=(0) temp44=nan
      make/O/N=(0) temp45=nan
      make/O/N=(0) temp46=nan
      make/O/N=(0) temp47=nan
      //newdatafolder/O test33
variable p1
      for(p1=0;p1<numw;p1+=1)
wave curW=root:anasel:$(stringfromlist(p1,wl,";"))
      wavestats/Q/R=(l1,r1) curW
      //duplicate/O/R=(l1,r1) curw, root:test33:$(nameofwave(curw))
variable top10=round(abs(l1-r1)/10)
      insertpoints (p1),1, temp42, temp43, temp44, temp45, temp46, temp47
      temp42[p1]=nameofwave(curW)
      temp43[p1]=top100(nameofwave(curw),l1,r1,top10)
      temp44[p1]=v_max
      temp45[p1]=v_min
      temp46[p1]=v_avg
      temp47[p1]=v_sdev
      //curW_e[W_FindLevels[p1],W_FindLevels[p1+1]-1]=v_min-1
      endfor

      if(strlen(lasch))      //filtered
      makeFW(wl)
wave fw1
      temp43*=fw1
      temp44*=fw1
      temp45*=fw1
      temp46*=fw1
      temp47*=fw1
      //killwaves/Z fw1
      endif
end

Function CheckProc_2filterSex(ctrlName,checked) : CheckBoxControl
String ctrlName
Variable checked
variable res=0
controlinfo/W=W_filter check5//male
res+=V_Value

```



```

controlinfo/W=W_filter check6//female
    res+=2*V_Value
SVAR f1=root:analysis:filters

string ol=stringfromlist(1, f1, "\r")
string nl=stringfromlist(0,ol,":")+":"+stringfromlist(1,ol,":")+":"+num2str(res)

    f1= replacestring(ol,f1,nl)
    //applyfilters()
End

Function CheckProc_usefilter(ctrlName,checked) : CheckBoxControl
    String ctrlName
    Variable checked
    if(checked)
        prepfiler()
    else
        wave fw1=root:fw1
        fw1=1
        dowindow/K w_filter
    endif

End

Function ButtonProc_2(ctrlName) : ButtonControl
    String ctrlName

//clear
    CheckProc_2Filter("check0",0)
    checkbox check0, win=cana, value=0
string wl=tracenamelist("cana#me",",",1)

variable p1
    for(p1=0;p1<itemsinlist(wl,",";);p1+=1)
        removefromgraph/W=cana#me $(stringfromlist(p1,wl,",";))
    endfor
    make/O/T/N=0 root:lw
    make/O/N=0 root:selw, root:fw1
    killdatafolder/Z root:anase1
    killdatafolder/Z root:analysis
    newdatafolder/O root:anase1
    newdatafolder/O root:Analysis

End

Function ButtonProc_appendall(ctrlName) : ButtonControl
    String ctrlName

string wl=getwl()//removefromlist("packages",getwl(),",",0)

variable p1,q1
    for(p1=0;p1<itemsinlist(wl,",";);p1+=1)
SVAR picN=root:picN1
        picN=stringfromlist(p1,wl,",";)
        make/o/N=0 curTrace
wave/T st=root:$(picN):roiMEAN
        if(waveexists(st))

            killdatafolder/Z root:curanase1

string e1=picN
        e1=replacestring(" ",e1,"_")
        e1= replacestring(".tif",e1,"")
        newdatafolder/O root:curAnaSel

```

```

                if(!datafolderExists("AnaSel"))
                    newdatafolder/O root:AnaSel
                endif

                for(q1=0;q1<numpts(st);q1+=1)
                    getcurTrace1(q1)

wave curtrace
                    duplicate/O root:curtrace, root:AnaSel:$(e1+"__"+num2str(q1))
wave cw=root:AnaSel:$(e1+"__"+num2str(q1))
                    note cw,picN          +";"+num2str(q1)
                    appendInfo(nameofwave(cw))
                    app2ME(root:AnaSel:$(e1+"__"+num2str(q1)),e1,q1)
                endfor
            endif
        endfor
wave lw
        make/O/N=(numpts(lw)) fw1=1
End

```

```

Function ButtonProc_1posneg(ctrlName) : ButtonControl
    String ctrlName

```

```

SVAR picN=root:picN1
    make/o/N=0 curTrace
wave/T st=root:$(picN):roiMEAN
    if(!waveexists(st))
        return 0
    endif

wave curtype=root:$(picN):roitype
    if(!waveexists(curtype))
        definecelltypes("")
    endif
wave curtype=root:$(picN):roitype

variable mc1=0
    if(stringmatch(ctrlName,"*9"))//positive for mC
        mc1=1
    endif

    controlinfo/W=canaSel popup0
string e1=S_value
    e1=replacestring(" ",e1,"_")
    e1=    replacestring(".tif",e1,"")
    if(!datafolderExists("AnaSel"))
        newdatafolder/O root:AnaSel
    endif

variable p1
    for(p1=0;p1<numpts(st);p1+=1)
        //print p1, curtype[p1]
        if(curtype[p1]==mc1)
            getcurTrace1(p1)

wave curtrace
                duplicate/O root:curtrace, root:AnaSel:$(e1+"__"+num2str(p1))
wave cw=root:AnaSel:$(e1+"__"+num2str(p1))
                note cw,picN          +";"+num2str(p1)
                appendInfo(nameofwave(cw))

```

```

                app2ME(root:AnaSel:$(e1+"__"+num2str(p1)),e1,p1)
                if(datafolderexists("root:analysis:allTraces:"))
                    duplicate/O root:AnaSel:$(e1+"__"+num2str(p1)), root:analysis:allTraces:$(e1+"__"+num2str(p1))
                else
                    backuptraces(1)
                endif
            endif
        endfor

        setdatafolder root:anasel:
string wl=wavelist(";",",","")
        SETDATAFOLDER root:
        make/O/N=(itemsinlist(wl,",")) root:fw1=1
    end

```

```

Function ButtonProc_appendMC(ctrlName) : ButtonControl
    String ctrlName

        getwindow W_curing, wavelist
wave/T w_wavelist

FindValue/TEXT="mcimg" w_wavelist
        if(v_value>=0)
            RemoveImage mCIMG
        else
SVAR picn=root:picn1
            AppendImage/T root:$(picn):mCIMG
            ModifyImage mCIMG ctab= {*,*,Gold,0}
        endif

End

```

```

function ButtonProc_restoremc(lasch)
string lasch

variable mc
        if(stringmatch(lasch,"*2"))//mC-positive
            mc=1
        endif

string picN=getPicN()
wave/T wC=root:$(picN):$("roiCOORD")
wave wT=root:$(picN):$("roiTYPE")
variable p1

make/O/N=0 xA,yA
        for (p1=0; p1<numpts(wC);p1+=1)
            if(wt[p1]==mc)
                makeCoordWfromStr(p1, 1)
wave xw,yw
                    concatenate/NP {xw},xA
                    concatenate/NP {yw},yA

                    insertpoints (numpts(xA)),1, xA,yA
                    xA[numpts(xA)-1]=NaN
                    yA[numpts(xA)-1]=NaN
                endif
            endif
        endfor
        setdrawlayer/W=W_curing/K progfront
        setdrawenv/W=W_curing xcoord=top,ycoord=left,linefgc=(65000,65000,0), linethick=1, fillpat=0,save

```

```

drawpoly/W=W_curing/ABS 0,0,1,1,xA,yA

controlinfo/W=fval check01

end

function getcelltype(str1,var1)
string str1 //data folder
variable var1 //cell number

wave mC=root:$(str1):mCIMG
wave/T owC=root:$(str1):roicoor
wave/T owB=root:$(str1):roibCOOR

dowindow/K tempDR
newimage/HIDE=1/N=tempDR/K=1 mc
setdrawlayer/W=tempDR/K progfront
setdrawenv/W=tempDR xcoord=top, ycoord=left,save

variable q1

make/O/N=0 root:xW,root:yW
wave xw=root:xw,yw=root:yw

string curC=owc[var1]
for(q1=0; q1<itemsinlist(curC,";"); q1+=1)
string curCoord=stringfromlist(q1,curc,";")
insertpoints q1, 1, xw,yw
xw[q1]=round(str2num(stringfromlist(0,curcoord,";")))
yw[q1]=round(str2num(stringfromlist(1,curcoord,";")))
endfor
drawpoly/W=tempDR/ABS 0,0,1,1, xw,yw
ImageGenerateROIMask/W=tempDR/E=1/l=0 mCIMG

wave M_ROIMask

Imagettransform/R=M_ROIMask roiTo1D MC

wave W_roi_to_1d
sort/R W_roi_to_1d,W_roi_to_1d
duplicate/O/R=[0,49] W_roi_to_1d, ana1
wavestats/Q ana1

variable VAL=V_avg

make/O/N=0 root:xW,root:yW
curC=owB[var1]
for(q1=0; q1<itemsinlist(curC,";"); q1+=1)
curCoord=stringfromlist(q1,curc,";")
insertpoints q1, 1, xw,yw
xw[q1]=round(str2num(stringfromlist(0,curcoord,";")))
yw[q1]=round(str2num(stringfromlist(1,curcoord,";")))
endfor
drawpoly/W=tempDR/ABS 0,0,1,1, xw,yw
ImageGenerateROIMask/W=tempDR/E=1/l=0 mCIMG

wave M_ROIMask

Imagettransform/R=M_ROIMask roiTo1D MC

sort/R W_roi_to_1d,W_roi_to_1d
duplicate/O/R=[0,49] W_roi_to_1d, ana1
wavestats/Q ana1

variable BCK=V_avg

return (VAL-BCK)

```

end

```
function definecelltypes(lasch):buttoncontrol
string lasch
```

```
controlinfo/W=canasel popup0
```

```
string str1=s_value
```

```
wave/T owC=root:${str1}:roicoor
```

```
make/O/N=(numpts(owC)) root:${str1}:roitype=-1
wave rt= root:${str1}:roitype
```

```
variable p1
```

```
for(p1=0;p1<numpts(owC);p1+=1)
```

```
rt[p1]=getcelltype(str1,p1)
```

```
endfor
```

```
Make/O/N=20/O rthist;DelayUpdate
```

```
Histogram/B={-20,2,100} rt,rthist;DelayUpdate
```

```
//K0 = 0;K2 = 0
```

```
//CurveFit/Q/H="1010"/NTHR=0 gauss rthist /D
```

```
//wave w_coef
```

```
setdatafolder root:${str1}
```

```
NVAR thr=root:mcTHR//ceil(w_coef[3]*2)
```

```
variable/G MCthr=thr
```

```
//print MCthr
```

```
setdatafolder root:
```

```
NVAR mcth= root:${str1}:MCthr
```

```
rt=(rt[p]>thr) ///0=neg;1=pos
```

```
wave rthist,fit_rthist,fit_roitype_hist
```

```
//killwaves/Z rthist,w_coef,fit_rthist,fit_roitype_hist//, fit_
```

end

Package 3: IHC-GABABR in microglia (17.8.21)

```
#pragma rtGlobals=3
```

```
// Use modern global access method and strict wave access.
```

```
function multipleanalysis(lasch)
```

```
string lasch
```

```
make/O/T/N=(2,11) MAres=""
```

```
wave/T res=root:MAres
```

```
res[0][0]="cell"
```

```
res[0][1]="image"
```

```
res[0][2]="pos[x,y]"
```

```
res[0][3]="threshold/filter"
```

```
res[0][4]="cell size (voxels)"
```

```
res[0][5]="background avg signal"
```

```
res[0][6]="background SD"
```

```
res[0][7]="cell avg signal"
```

```
res[0][8]="# voxels above threshold"
```

```
res[0][9]="DAPI size"
```

```
res[0][10]="DAPI excluded"
```

```
newpath/O/Q MApath
```

```
if(V_flag)
```

```
return 0
```

```
endif
```

```
pathinfo MApath
```

```
dowindow/K w_MAres
```

```
edit/W=(5.25,43.25,1013.25,237.5)/K=1/N=w_MAres MAres as "All cells from "+stringfromlist(itemsinlist(s_path,":-1,s_path,":-1)
```

```
string fl=IndexedFile(MApath, -1, ".pyp" )// [, creatorStr ])
```

```
string obj="W_iba;w_dapi;w_vglut"
```

```
variable p1
```

```
for(p1=0;p1<itemsinlist(fl,"");p1+=1)
```

```
LoadData/O/Q/P=MApath/J=obj (stringfromlist(p1,fl,""))// [/D/I/J=objectNamesStr /L=loadFlags /O[=overwriteMode ]  
/P=pathName /Q/R/S=subDataFolderStr /T[=topLevelName ] ] fileOrFolderNameStr
```

```
LoadData/O/Q/P=MApath/S="Cells"/R/T (stringfromlist(p1,fl,""))
```

```
ButtonProc_2reanalyse("")
```

```
wave /T resC
```

```
variable rem=(dimsize(MAres,0))
```

```
variable n1=dimsize(resc,0)-2
```

```
insertpoints/M=0 (rem),(n1),MAres
```

```
MAres[rem,rem-1+n1][]=resc[p+2-rem][q]
```

```
endfor
```

```
end
```

```
function checkScale(lasch)
```

```
string lasch
```

```
wave cell2 //threshold 3D
```

```
wave glu2 ///gabbr signal 3D
```

```
wave dapi=:mask_dapi//dapi mask 3D
```

```

duplicate/O glu2, glu2o, glu2c//outside, cell

glu2o*=(cell2==0)

controlinfo/W=panel0 check0

if( (v_value)&&(waveexists(dapi)) )///dapi exclusion
    glu2o*=(dapi==1)//nuclear region is no background
endif

glu2c*=(cell2==1)

Controlinfo/W=panel0 check2

if(v_value)
    onllyomatic()//generates a rectangular mask
wave cell2=:tmpOS
wave tmp44
    glu2o*=(tmp44[p][q]==0)
    glu2c*=(tmp44[p][q]==0)
    corrCS()
else
wave cell2=:cell2
endif

//imagetransform/METH=1 zProjection glu2o
//wave M_zProjection
//    duplicate/O M_zProjection, co,co1
    duplicate/O glu2o,glu2or
wave co=glu2or
    redimension/N=(numpnts(co))/S co
    co/=(co[p]!=0)
    co/=(co[p]>=6)

```



```

//      wavestats/Q co

      make/O h_c,h_o

      Histogram/B={0,1,255} co,h_o

variable mo=wavemax(h_o)

      h_o/=mo

      wavestats/Q co

variable/G v_extra=v_avg

variable/G v_thr=2*v_sdev

//print "outside:",v_extra,v_thr


//      imagetransform/METH=1 zProjection glu2c

//      duplicate/O M_zProjection, cc,cc1

      duplicate/O glu2c,glu2cr

wave cc=glu2cr


      redimension/N=(numpnts(cc))/S cc

      cc/=(cc[p]!=0)

//      wavestats/Q cc

//      print "cell:",v_npnts,v_avg,v_sdev

      Histogram/B={0,1,255} cc,h_c

variable mc=wavemax(h_c)

      h_c/=mc


k0 = 0;K1 = 1

      CurveFit/Q/H="1100"/NTHR=0 gauss h_c /D

wave w_coef

variable/G v_cell=w_coef[2]

      wavestats/Q/R=[round(v_extra+v_thr)+1,255] h_c

variable/G v_PoT=V_sum*mc

```

```

setdrawlayer/K/W=panel0#g5 userfront

SetDrawEnv/W=panel0#g5 xcoord= bottom,ycoord= prel,linefgc= (0,0,0), linethick=2

drawline/W=panel0#g5 (V_extra+v_thr),0,(V_extra+v_thr),1

SetDrawEnv/W=panel0#g5 xcoord= bottom,ycoord= prel,linefgc= (40000,0,0), linethick=2

drawline/W=panel0#g5 (V_cell),0,(V_cell),1

removefromgraph/Z/W=panel0#g5 fit_h_o,fit_h_c

```

```

end

```

```

function corrCS()

wave tmp44//rect mask

wave cell2 //cell mask

duplicate/O cell2, tmp45

tmp45*=(tmp44==0)

wavestats/Q tmp45

NVAR res_size

res_size=v_sum

killwaves/Z tmp45

```

```

end

```

```

function onlysomatic()

wave mask_dapi

if(!waveexists(mask_dapi))

```

```

        return 0

    endif

variable xyr=10

variable zr=2

    duplicate/O mask_dapi, tmp44

    if(mean(mask_dapi)==1)

wave cell2

        duplicate/O cell2 tmpOS

        return 0

    endif


wave tmp44

    tmp44*=255

    imageanalyzeparticles/A=500 stats tmp44


wave M_3DParticleInfo

variable x1=M_3DParticleInfo[0][0]-xyr

    if(x1<0)

        x1=0

    endif

variable x2=M_3DParticleInfo[0][1]+xyr

    if(x2>(dimsize(mask_dapi,0)-1))

        x2=dimsize(mask_dapi,0)-1

    endif

variable y1=M_3DParticleInfo[0][2]-xyr

    if(y1<0)

        y1=0

    endif

variable y2=M_3DParticleInfo[0][3]+xyr

    if(y2>dimsize(mask_dapi,1)-1)

        y2=dimsize(mask_dapi,1)-1

```

```

endif
variable z1=M_3DParticleInfo[0][4]-zr
    if(z1<0)
        z1=0
    endif
variable z2=M_3DParticleInfo[0][5]+zr
    if(z2>dimsize(mask_dapi,2)-1)
        z2=dimsize(mask_dapi,2)-1
    endif

```

```

//print x1,x2,y1,y2,z1,z2

```

```

    tmp44[x1,x2][y1,y2][z1,z2]=0
wave cell2

```

```

duplicate/O cell2 tmpOS
tmpOS*=(tmp44==0)
//newimage/S=0/K=1 tmpOS
killwaves/Z M_3DParticleInfo
end

```

```

function storedata(vl,vr,vt,vb)
variable vl,vr,vt,vb
string str=replacestring(", ",stringbykey("FOLDERS",DataFolderDir(1),":",""),";")
string num=num2str(itemsinlist(str))

```

NVAR res_size

NVAR glut_fl

NVAR glut_sp

NVAR gaba_fl

NVAR gaba_sp

wave w_iba

```
print "file:\t", "cell size:\t", "glu spots:\t", "glu fluorescence:\t", "gab spots:\t", "gab fluorescence:\t"
```

```
print note(W_iba), "\t", vl, vr, vt, vb, "\t", res_size, "\t", glut_sp, "\t", glut_fl, "\t", gaba_sp, "\t", gaba_fl
```

end

function globals()

variable/G thr=30

variable/G smo=10

variable /G ibachan

variable/G res_size

variable /G imgF=3

variable/G xyr// xy range; odd number

variable/G zr//z range; odd number

variable /G rminA,rmaxA=255

variable /G gminA,gmaxA=255

variable /G bminA,bmaxA=255

String/G path1

variable/G somatic=0

make/O h_c=nan

make/O h_o=nan

make/O/N=(2,2)/B/U allcell=0

make/O/N=(2,2)/B/U cell=0

make/O/N=(2,2)/B/U cellsmod=0

make/O/N=(2,2)/B/U cell2d=0

make/O/N=(2,2)/B/U all_vglut=0

make/O/N=(2,2)/B/U s_vglut1=0

make/O/N=(2,2)/B/U s_vglut2=0

make/O/N=(2,2)/B/U all_dapi=0

make/O/N=(2,2)/B/U all_dapiP2=1

make/O/N=(2,2)/B/U all_dapi2MP=1

end

```

function g2gray(wn, var)

string wn

variable var //0=red,1=green, 2=blue

wave curl=${wn}

variable nl=dimsize(curl,3)

    if(!nl)

        duplicate/O curl, output

        return 0

    endif

    make/O/N=(dimsize(curl,0),dimsize(curl,1),dimsize(curl,3)) output

variable p1

    for(p1=0;p1<nl;p1+=1)

        imagetransform/CHIX=(p1) getChunk curl

wave M_Chunk

        imagetransform/P=(var) getplane M_Chunk

wave M_imageplane

        imagetransform/P=(p1)/D=M_imageplane setplane output

    endfor

    killwaves/Z m_chunk,M_imageplane

//newimage/K=1/S=0 output

end

```

Function CheckProc(ctrlName,checked) : CheckBoxControl

```
    String ctrlName
    Variable checked
//    print ctrlName
string n1
    if(stringmatch (ctrlName,"glut*"))
        n1="glut"
    elseif(stringmatch (ctrlName,"gab*"))
        n1="gab"
    endif
string n2
variable m1,ms,c1,c2,c3
    if(stringmatch (ctrlName,"*N"))
        n2="n"
        m1=8
        ms=0.5
        c1=34816
        c2=34816
        c3=34816
    elseif(stringmatch (ctrlName,"*A"))
        n2="o"
        m1=19
        ms=3
        c1=0
        c2=26112
        c3=26112
```



```

elseif(stringmatch (ctrlname,"*i"))

    n2="i"

    m1=16

    ms=3

    c1=0

    c2=52224

    c3=52224

endif

if(checked)

    AppendToGraph/L/T $(n1+"y"+n2) vs $(n1+"X"+n2)

    ModifyGraph marker($(n1+"y"+n2))=m1, mode=3

    ModifyGraph msize($(n1+"y"+n2))=ms

    ModifyGraph rgb($(n1+"y"+n2))=(c1,c2,c3)

else

    removefromGraph/Z $(n1+"Y"+n2)

endif

```

End

```
Function ButtonProc_proj(ctrlName) : ButtonControl
```

```
    String ctrlName
```

```
    controlinfo/W=panel0 check44
```

```
    if(v_value)
```

```
        buttonproc_restore("")
```

```
    endif
```

```
    fw_proj()
```

```
End
```

```
function fw_proj()
```

```
    dowindow/K w_proj
```

```
    PauseUpdate; Silent 1          // building window...
```

```
    NewPanel /W=(671,133,1443,905)/K=1 /N=w_proj as "Projection"
```

```
    SetDrawLayer UserBack
```

```
    SetDrawEnv arrow= 1
```

```
    DrawLine 385,370,385,20
```

```
    SetDrawEnv arrow= 1
```

```
    DrawLine 400,385,750,385
```

```
    SetDrawEnv fsize= 18
```

```
    DrawText 755,395,"x"
```

```
    SetDrawEnv fsize= 18
```

```
    DrawText 381,770,"z"
```

```
    SetDrawEnv dash= 1,arrow= 1
```

```
    DrawLine 370,385,20,385
```

```
    SetDrawEnv dash= 1,arrow= 1
```

```
    DrawLine 385,400,385,750
```

```

SetDrawEnv fsize= 18

DrawText 10,395,"z"

SetDrawEnv fsize= 18

DrawText 381,17,"y"

GroupBox group0,pos={20,400},size={350,350},labelBack={34816,34816,34816}

//CheckBox check0,pos={36,418},size={43,14},proc=CheckProc_1_setw,title="Crop "

//CheckBox check0,value= 1

Button button0,pos={100,450},size={150,40},proc=ButtonProc_crop,title="Delete"

Button button1,pos={100,500},size={150,40},proc=ButtonProc_crop,title="Crop"

Button button2,pos={100,600},size={150,40},proc=ButtonProc_restore,title="Restore"

Button button3,pos={100,650},size={150,40},proc=ButtonProc_useit,title="Use"

SetVariable setvar0,pos={121,417},size={108,16},bodyWidth=40,proc=SetVarProc_filter,title="Avg Img Filter"

SetVariable setvar0,limits={3,inf,2},value= imgF

```

```

wave cellZ, cellX,cellY

```

```

Display/W=(400,20,750,370)/HOST=#

AppendImage/T cellZ

ModifyImage cellZ ctab= {5,255,Grays,0}

ModifyGraph margin=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,Gxy

SetActiveSubwindow ##

```

```

Display/W=(20,20,370,370)/HOST=#

AppendImage/T cellX

ModifyImage cellX ctab= {5,255,Grays,0}

```

ModifyGraph margin=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

SetAxis/A top

RenameWindow #,Gyz

SetActiveSubwindow ##

Display/W=(400,400,750,750)/HOST=#

AppendImage/T cellY

ModifyImage cellY ctab= {10,255,Grays,0}

ModifyGraph margin=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,Gxz

SetActiveSubwindow ##

setwindow W_proj, hook=hook_crop,hookevents=7

make/O/N=0 xw,yw

GraphWaveDraw /W=\${"W_proj"}#\$("Gyz") /L/T/O yW, xW

End

```
function gwd(lasch1, lasch2)
```

```
string lasch1 //window
```

```
string lasch2 //subwindow
```

```
removefromgraph/W=${lasch1}#Gxy/Z yw
```

```
removefromgraph/W=${lasch1}#Gxz/Z yw
```

```
removefromgraph/W=${lasch1}#Gyz/Z yw
```

```
GraphWaveDraw /W=${lasch1}#${lasch2} /L/T/O yW, xW
```

```
wave xw
```

```
note/K xw, lasch2
```

```
end
```

```
Function hook_crop (infoStr)
```

```
String infoStr
```

```
String ev= StringByKey("event",infoStr)
```

```
String mo= StringByKey("modifiers",infoStr)
```

```
String Hw= StringByKey("HCSPEC",infoStr)
```

```
variable dec=0
```

```
if((str2num(mo)==5)&&(stringmatch(ev,"mousedown")))//case Graphmormal
```

```
dec+=1
```

```
endif
```

```
if((str2num(mo)==4)&&(stringmatch(ev,"mouseup")))//case graphwavedraw
```

```
dec+=1
```

```
endif
```

```

if(dec)

    if(stringmatch(hw,"**")) //projection win

        gwd(stringfromlist(0,hw,"#"),stringfromlist(1,hw,"#"))

    else

        graphnormal/W=$(hw)#gyz

    endif

endif

```

End

Function CheckProc_1_setw(ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

```

if(checked)

    setwindow W_proj, hook=hook_crop,hookevents=7

    make/O/N=0 xw,yw

else

    setwindow W_proj, hook=""

    killwaves/Z xw,yw

endif

```

End

Function ButtonProc_crop(ctrlName) : ButtonControl

```

String ctrlName

string rem="\t"

wave cellz

string o1=note(cellz)

variable it1=itemsinlist(o1,"\t")

    rem+="Action_"+num2str(it1)+"\r"

wave temp44

    if(!waveexists(temp44))

wave cell1

        duplicate/o cell1, temp44

    endif

wave xw,yw

string pr=note(xw)

    rem+=pr+"\r"        //1st row:window

    if(stringmatch(pr,"Gxy"))

wave curw=root:cellz

    endif

    if(stringmatch(pr,"Gxz"))

wave curw=root:cellY

    endif

    if(stringmatch(pr,"Gyz"))

wave curw=root:cellX

    endif

    ImageBoundaryToMask                                width=dimsize(curw,0),
height=dimsize(curw,1),scalingWave=curw,seedy=dimoffset(curw,1),seedx=dimoffset(curw,0), xwave=xw, ywave=yw

wave M_ROIMask

    if(stringmatch (ctrlName,"*1"))//crop

        M_Roimask=(M_Roimask[p][q]==0)

```

```

        rem+="crop_1\r"

else ///delete, as it is

        rem+="delete_2\r"

endif

if(stringmatch(pr,"Gxy"))

        temp44*=m_roimask[p][q]

endif

if(stringmatch(pr,"Gxz"))

        temp44[][]*=m_roimask[p][r]

endif

if(stringmatch(pr,"Gyz"))

        imagetransform fliprows m_roimask

        temp44[][]*=m_roimask[r][q]

endif

        imagetransform zprojection temp44

wave M_zprojection

        imagetransform xprojection temp44

wave M_xprojection

        imagetransform yprojection temp44

wave M_yprojection

        duplicate/O M_zprojection, cellZ

        duplicate/O M_xprojection, cellX

        duplicate/O M_yprojection, cellY

        ImageRotate/A=90/O cellx

        killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh

rem+="xw:"+w2str(xw)+"\r"

rem+="yw:"+w2str(yw)+"\r"

```



```
note cellz, o1+rem
```

```
End
```

```
function rest_crop(crop)
```

```
string crop
```

```
if(!strlen(crop))
```

```
return 0
```

```
endif
```

```
variable p1
```

```
for(p1=1;p1<itemsinlist(crop,"\t");p1+=1)
```

```
string/G act=stringfromlist(p1,crop,"\t")
```

```
restore_xyw(act)
```

```
wave xw
```

```
note/K xw, stringfromlist(1,act,"\r")
```

```
ButtonProc_crop(stringfromlist(2,act,"\r"))
```

```
endfor
```

```
end
```

```

function/S w2str(w1)

wave w1

string expo=""

variable p1

    for(p1=0; p1<numpnts(w1); p1+=1)

        expo+=num2str(w1[p1])+";"

    endfor

    return expo

end

```

```

Function SetVarProc_filter(ctrlName,varNum,varStr,varName) : SetVariableControl

```

```

    String ctrlName

```

```

    Variable varNum

```

```

    String varStr

```

```

    String varName

```

```

wave temp44

```

```

    ImageFilter/N=(varNum)/O gauss3d temp44

```

```

    imagettransform zprojection temp44

```

```

wave M_zprojection

```

```

        imagetransform xprojection temp44
wave M_xprojection
        imagetransform yprojection temp44
wave M_yprojection
        duplicate/O M_zprojection, cellZ
        duplicate/O M_xprojection, cellX
        duplicate/O M_yprojection, cellY
        ImageRotate/A=90/O cellx
        killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh
End

```

```

Function PopMenuProc_col(ctrlName,popNum,popStr) : PopupMenuControl

```

```

    String ctrlName
    Variable popNum
    String popStr

    if(str2num(ctrlName[5])==0)
NVAR g1
        g1=popnum
    endif
    if(str2num(ctrlName[5])==1)
NVAR i1
        i1=popnum
    endif
    if(str2num(ctrlName[5])==2)
NVAR ga1
        ga1=popnum
    endif
    if(str2num(ctrlName[5])==3)

```

NVAR gl1

gl1=popnum

endif

End

Function ButtonProc_rgbDoit(ctrlName) : ButtonControl

String ctrlName

doRGBproj()

End

function ButtonProc_expPICK(lasch):buttoncontrol

string lasch

dowindow/K w_projRGB0

display/K=1 /W=(335,75,1107,847)/N=w_projRGB0 as "RGB projection"

SetDrawLayer userfront

SetDrawEnv arrow= 1

SetDrawEnv xcoord=abs, ycoord=abs, save

DrawLine 385,370,385,20

SetDrawEnv arrow= 1

DrawLine 400,385,750,385

SetDrawEnv fsize= 18

DrawText 755,395,"x"

SetDrawEnv fsize= 18

DrawText 381,770,"z"

SetDrawEnv dash= 1,arrow= 1

DrawLine 370,385,20,385

SetDrawEnv dash= 1,arrow= 1

DrawLine 385,400,385,750

SetDrawEnv fsize= 18

DrawText 10,395,"z"

SetDrawEnv fsize= 18

DrawText 381,17,"y"

wave cellZRGB, cellXRGB,cellYRGB

Display/W=(400,20,750,370)/HOST=#

AppendImage/T cellZRGB

ModifyImage cellZRGB ctab= {5,50,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

RenameWindow #,Gxy

SetActiveSubwindow ##

Display/W=(20,20,370,370)/HOST=#

AppendImage/T cellXRGB

ModifyImage cellXRGB ctab= {5,100,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

RenameWindow #,Gyz

SetActiveSubwindow ##

```

Display/W=(400,400,750,750)/HOST=#

AppendImage/T cellYRGB

ModifyImage cellYRGB ctab= {10,100,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,Gxz

SetActiveSubwindow ##

string rz=calcS()

//print rz

variable relx=str2num(stringfromlist(0,rz,";"))

variable rely=str2num(stringfromlist(1,rz,";"))

variable relz=str2num(stringfromlist(2,rz,";"))

//Gyz

ModifyGraph/W=w_projRGB0#Gyz axisEnab(left)={{(1-relx),1} //x

ModifyGraph/W=w_projRGB0#Gyz axisEnab(top)={{(1-relz),1} //z

//gxz

ModifyGraph/W=w_projRGB0#Gxz axisEnab(left)={{(1-relz),1} //z

ModifyGraph/W=w_projRGB0#Gxz axisEnab(top)={{(1-rely),1} //y

//Gxy

ModifyGraph/W=w_projRGB0#Gxy axisEnab(top)={{(1-rely),1} //y

ModifyGraph/W=w_projRGB0#Gxy axisEnab(left)={{(1-relx),1} //x

```

```
//      DolgorMenu "file", "Save Graphics"

end
```

```
function/S calcS()

//cut segment size

wave cell

string n1=note(cell)

variable vl1=str2num(stringfromlist(0,n1,";"))

variable vr1=str2num(stringfromlist(1,n1,";"))

variable vt1=str2num(stringfromlist(2,n1,";"))

variable vb1=str2num(stringfromlist(3,n1,";"))
```

```
variable vsy=abs(vr1-vl1)

variable vsx=abs(vb1-vt1)
```

```
//wholeimg

wave allcell

variable vwx=dimsize(allcell,0)

variable vwy=dimsize(allcell,1)
```

```
//stack:50  $\mu$ m

//whole pic: 160X160
```

```
make/O/N=3 res

res[0]=(vsx/vwx)*160

res[1]=(vsy/vwy)*160

res[2]=50

variable wm=wavemax(res)

res/=wm
```

```
return num2str(res[0])+";"+num2str(res[1])+";"+num2str(res[2])+";"
```

```
end
```

```
function f_projRGB()
```

```
Dowindow/K w_projRGB
```

```
    rgbglob()
```

```
PauseUpdate; Silent 1          // building window...
```

```
newpanel/K=1 /W=(335,75,1107,847)/N=w_projRGB as "RGB projection"
```

```
SetDrawLayer UserBack
```

```
SetDrawEnv arrow= 1
```

```
DrawLine 385,370,385,20
```

```
SetDrawEnv arrow= 1
```

```
DrawLine 400,385,750,385
```

```
SetDrawEnv fsize= 18
```

```
DrawText 755,395,"x"
```

```
SetDrawEnv fsize= 18
```

```
DrawText 381,770,"z"
```

```
SetDrawEnv dash= 1,arrow= 1
```

```
DrawLine 370,385,20,385
```

```
SetDrawEnv dash= 1,arrow= 1
```

```
DrawLine 385,400,385,750
```

```
SetDrawEnv fsize= 18
```

```
DrawText 10,395,"z"
```

```
SetDrawEnv fsize= 18
```

```
DrawText 381,17,"y"
```


GroupBox group0,pos={20,400},size={350,350},labelBack={34816,34816,34816}

NVAR g1,gz1,gz2,gcon

GroupBox group1,pos={39,417},size={319,58},title="Background"

CheckBox check2,pos={124,413},size={24,14},title="F",value= 0

PopupMenu popup0,pos={49,442},size={60,21},bodyWidth=60, proc=PopupMenuProc_col

PopupMenu popup0,mode=(g1),popvalue="Gray",value= #""\ "Gray;Red;Green;Blue;Purple\ ""

SetVariable setvar6,pos={114,445},size={45,16},bodyWidth=30,title="z1"

SetVariable setvar6,limits={-inf,inf,0}, value=gz1

SetVariable setvar7,pos={174,445},size={45,16},bodyWidth=30,title="z2"

SetVariable setvar7,limits={-inf,inf,0}, value=gz2

SetVariable setvar0,pos={229,443},size={119,16},bodyWidth=60,title="contribution"

SetVariable setvar0,limits={-inf,inf,10}, value=gcon

NVAR i1,iz1,iz2,icon

GroupBox group2,pos={39,477},size={319,67},title="Iba1"

PopupMenu popup1,pos={49,506},size={60,21},bodyWidth=60, proc=PopupMenuProc_col

PopupMenu popup1,mode=(i1),popvalue="Green",value= #""\ "Gray;Red;Green;Blue;Purple\ ""

SetVariable setvar8,pos={114,509},size={45,16},bodyWidth=30,title="z1"

SetVariable setvar8,limits={-inf,inf,0}, value=iz1

SetVariable setvar9,pos={174,509},size={45,16},bodyWidth=30,title="z2"

SetVariable setvar9,limits={-inf,inf,0}, value=iz2

SetVariable setvar1,pos={229,509},size={119,16},bodyWidth=60,title="contribution"

SetVariable setvar1,limits={-inf,inf,10}, value=icon

NVAR ga1,gaz1,gaz2,gacon

GroupBox group3,pos={39,549},size={319,68},title="Vgat"

PopupMenu popup2,pos={49,577},size={60,21},bodyWidth=60, proc=PopupMenuProc_col

PopupMenu popup2,mode=(ga1),popvalue="Blue",value= #""\ "Gray;Red;Green;Blue;Purple\ ""

SetVariable setvar4,pos={114,580},size={45,16},bodyWidth=30,title="z1"

```

SetVariable setvar4,limits={-inf,inf,0}, value=gaz1

SetVariable setvar5,pos={174,580},size={45,16},bodyWidth=30,title="z2"

SetVariable setvar5,limits={-inf,inf,0}, value=gaz2

SetVariable setvar2,pos={229,573},size={119,16},bodyWidth=60,title="contribution"

SetVariable setvar2,limits={-inf,inf,10},value= gacon

CheckBox check0,pos={289,595},size={43,14},title="ideal ",value= 0,proc=CheckProc_1ideal

```

```

NVAR gl1,glz1,glz2,glcon

```

```

GroupBox group4,pos={39,622},size={319,67},title="Vglut"

PopupMenu popup3,pos={49,652},size={60,21},bodyWidth=60, proc=PopupMenuProc_col

PopupMenu popup3,mode=(gl1),popvalue="Red",value= #"\Gray;Red;Green;Blue;Purple\ ""

SetVariable setvar05,pos={114,655},size={45,16},bodyWidth=30,title="z1"

SetVariable setvar05,limits={-inf,inf,0}, value=glz1

SetVariable setvar06,pos={174,655},size={45,16},bodyWidth=30,title="z2"

SetVariable setvar06,limits={-inf,inf,0}, value=glz2

SetVariable setvar3,pos={229,645},size={119,16},bodyWidth=60,title="contribution"

SetVariable setvar3,limits={-inf,inf,10},value= glcon

CheckBox check1,pos={289,668},size={43,14},title="ideal ",value= 0,proc=CheckProc_1ideal

```

```

Button button0,pos={295,700},size={60,40},proc=ButtonProc_rgbDoit,title="Doit"

Button button1,pos={40,700},size={60,40},proc=ButtonProc_expPICT,title="Export"

```

```

doRGBproj()

```

```

wave cellZRGB, cellXRGB,cellYRGB

```

```

Display/W=(400,20,750,370)/HOST=#

```

AppendImage/T cellZRGB

ModifyImage cellZRGB ctab= {5,50,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

RenameWindow #,Gxy

SetActiveSubwindow ##

Display/W=(20,20,370,370)/HOST=#

AppendImage/T cellxRGB

ModifyImage cellXRGB ctab= {5,100,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

RenameWindow #,Gyz

SetActiveSubwindow ##

Display/W=(400,400,750,750)/HOST=#

AppendImage/T cellyRGB

ModifyImage cellYRGB ctab= {10,100,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

```
ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,Gxz

SetActiveSubwindow ##

End
```

```
function rgbglob()
```

```
variable/G g1=1 //gray
```

```
variable/G gz1=0
```

```
variable/G gz2=255
```

```
variable/G gcon=100
```

```
variable/G i1=3//green
```

```
variable/G iz1=0
```

```
variable/G iz2=255
```

```
variable/G icon=100
```

```
variable/G ga1=4//blue
```

```
variable/G gaz1=0
```

```
variable/G gaz2=255
```

```
variable/G gacon=100
```

```
variable/G gl1=2//red
```

```
variable/G glz1=0
```

```
variable/G glz2=255
```

```
variable/G glcon=100
```

```
end
```

```
function tempproj(w1,var1)
```

```
wave w1
```

```
variable var1 //threshold
```

```
    w1*=(w1[p][q][r]>=var1)
```

```
    imagetransform xprojection w1
```

```
wave M_xprojection
```

```
    imagetransform zprojection w1
```

```
wave M_zprojection
```

```
    imagetransform yprojection w1
```

```
wave M_yprojection
```

```
    duplicate/O M_xprojection, tempX
```

```
    duplicate/O M_yprojection, tempY
```

```
    duplicate/O M_zprojection, tempZ
```

```
    ImageRotate/A=90/O tempX
```

```
variable xmax=wavemax(tempX)/255
```

```
    tempX/=xmax
```

```
variable ymax=wavemax(tempY)/255
```

```
    tempY/=ymax
```

```
variable zmax=wavemax(tempZ)/255
```

```
    tempZ/=zmax
```

```
    //print wavemax(tempX),wavemax(tempY),wavemax(tempZ)
```

```
killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh
```

```
end
```

```
function doRGBproj()
```

```
wave cell1 //iba cell stack
```

```
wave cellsmo //filtered iba cell stack
```

```
Make/O/B/U/N=(dimsize(cell1,0),dimsize(cell1,1),3) cellZRGB=0
```

```
Make/O/B/U/N=(dimsize(cell1,2),dimsize(cell1,1),3) cellXRGB=0
```

```
Make/O/B/U/N=(dimsize(cell1,0),dimsize(cell1,2),3) cellYRGB=0
```

```
wave cellZRGB, cellXRGB,cellYRGB
```

```
///background, gray, lba1 channel
```

```
NVAR g1
```

```
NVAR gz1
```

```
NVAR gz2
```

```
NVAR gcon
```

```
controlinfo/W=w_projRGB check2
```

```
if(!v_value)
```

```
duplicate/O cell1, temp444///cell1== lba stack
```

```
else
```

```

        duplicate/O cellsmo, temp444///cellsmo==filtered iba stack

    endif

    tempproj(temp444,0)

wave tempx,tempy,tempz


    scaletemp(gz1,gz2)


variable s1=gcon

string col1=getcol(g1)

variable p1

    for(p1=0;p1<itemsinlist(col1);p1+=1)

        cellxRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempx[p][q]/(255/s1)

        cellyRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempy[p][q]/(255/s1)

        cellzRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempz[p][q]/(255/s1)

    endfor


///Iba thresh, green

wave cell2 ///cropped iba image

NVAR i1

NVAR iz1

NVAR iz2

NVAR icon

    duplicate/O cell2, temp445

    tempproj(temp445,0)

    scaletemp(iz1,iz2)

    s1=icon

    col1=getcol(i1)

    for(p1=0;p1<itemsinlist(col1);p1+=1)

        cellxRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempx[p][q]/(255/s1)

        cellyRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempy[p][q]/(255/s1)

```

```

        cellZRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempz[p][q]/(255/s1)

    endfor

killwaves/Z temp444,temp445,temp446,temp447


///vglut/vgat thresh, red


NVAR gl1
NVAR glz1
NVAR glz2
NVAR glcon
NVAR thrGL

wave gluRGB=root:glu1 //vglut

    duplicate/O gluRGB, temp447

    tempproj(temp447,thrGL)

    scaletemp(glz1,glz2)

    s1=glcon

    col1=getcol(gl1)

    for(p1=0;p1<itemsinlist(col1);p1+=1)

        cellxRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempx[p][q]/(255/s1)

        cellyRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempy[p][q]/(255/s1)

        cellZRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempz[p][q]/(255/s1)

    endfor

```



```
wave gabRGB=root:gab1 //vgat
```

```
NVAR ga1
```

```
NVAR gaz1
```

```
NVAR gaz2
```

```
NVAR gacon
```

```
NVAR thrGA
```

```
duplicate/O gabRGB, temp446
```

```
tempproj(temp446,thrGA)
```

```
scaletemp(gaz1,gaz2)
```

```
s1=gacon
```

```
col1=getcol(ga1)
```

```
for(p1=0;p1<itemsinlist(col1);p1+=1)
```

```
    cellxRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempz[p][q]/(255/s1)
```

```
    cellyRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempz[p][q]/(255/s1)
```

```
    cellzRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempz[p][q]/(255/s1)
```

```
endfor
```

```
setscale/P x (dimoffset(cell1,0)),1,cellzRGB
```

```
setscale/P y (dimoffset(cell1,1)),1,cellzRGB
```

```
setscale/P y (dimoffset(cell1,1)),1,cellxRGB
```

```
setscale/P x (dimoffset(cell1,0)),1,cellyRGB
```

```
end
```

```
function/S getcol(var)
```

```
variable var
```

```
if(var==1)//gray
```

```

        return "0;1;2"
    endif
    if(var==2)//red
        return "0"
    endif
    if(var==3)
        return "1"
    endif
    if(var==4)
        return "2"
    endif
    if(var==5)
        return "0;2"
    endif

end
end

```

```
function scaletemp(gz1,gz2)
```

```
variable gz1,gz2
```

```
wave tempx,tempy,tempz
```

```
redimension/S tempx
```

```
redimension/S tempy
```

```
redimension/S tempz
```

```
tempx-=gz1
```

```
tempx*=(tempx[p][q]>=0)
```

```
tempx*=255/(gz2)
```

```
tempx/=(tempx[p][q]<=255)
```

```

    tempy-=gz1

    tempy*=(tempy[p][q]>=0)

    tempy*=255/gz2

    tempy/=(tempy[p][q]<=255)


    tempz-=gz1

    tempz*=(tempz[p][q]>=0)

    tempz*=255/gz2

    tempz/=(tempz[p][q]<=255)

    redimension/B/U tempx

    redimension/B/U tempy

    redimension/B/U tempz

end

```

```

#pragma rtGlobals=3           // Use modern global access method and strict wave access.

```

```

function autoana(lasch)

```

```

string lasch

    getmarquee /W=panel0#G0 left,top

    if(!V_flag)

        return 0

    endif

    quantify("")

    ButtonProc_4lbamask("")

    ButtonProc_useit("werDAPI")

    Checkscale("")

    setdatafolder root:cells

string cl=replacestring(";",stringbykey("FOLDERS",datafolderdir(1),":",";")) //data folders

variable  num=itemsinlist(cl,";")+1

string    nn=("cell"+num2str(num))

    setdatafolder root:

    ButtonProc_saveCell("auto_"+nn)

    CheckProc_2chowAna("",1)

end

```

```

menu "analysis"

"_"

"Quantify IHC signals", f_pruning()

end

```

```

function f_Pruning ()

dowindow/K Panel0

    globals()

    PauseUpdate; Silent 1          // building window...

    NewPanel/K=1 /N=Panel0/W=(133,109,1630,894) as "Pruning..."

    modifypanel/W=Panel0 fixedsize=1


    GroupBox group3,pos={21,0},size={747,75},title=" "

    GroupBox group3,labelBack=(39168,39168,39168)


    Button button1,pos={34,22},size={50,20},proc=loadIMG,title="Iba1"

    Button button1,fColor=(6400,13056,0)

    Button button2,pos={100,22},size={50,20},proc=loadIMG,title="GABBR"

    Button button2,fColor=(62976,0,5888)

    Button button03,pos={34,49},size={50,20},proc=loadIMG,title="DAPI"

    Button button03,fColor=(0,12800,52224)


SVAR p_1=:path1

    SetVariable setvar15,pos={160,36},size={570,16},bodyWidth=544,title="Path"

    SetVariable setvar15,value= p_1,noedit= 1


    GroupBox group4,pos={1330,0},size={153,75},title=" "

    GroupBox group4,labelBack=(39168,39168,39168)

    Button button10,pos={1097,29},size={60,30},proc=ButtonProc_2editresults,title="Exp Data"

```

GroupBox group5,pos={780,0},size={539,75},title=" "

GroupBox group5,labelBack={39168,39168,39168}

Button button9,pos={946,26},size={60,30},proc=ButtonProc_saveCell,title="Save Cell"

Button button11,pos={1035,26},size={60,30},proc=ButtonProc_2reanalyse,title="Reanalyse"

PopupMenu popup0,pos={789,31},size={128,21},bodyWidth=80,title="Open cell"

PopupMenu popup0,mode=1,value= getcells()

PopupMenu popup0 proc=PopupMenuProc_loadcell

CheckBox check1,pos={1142,38},size={113,14},proc=CheckProc_2chowAna,title="show Analysed cells"

CheckBox check1,value=0

Button button09,pos={1413,26},size={60,30},proc=multipleanalysis,title="Multi Ana"

Button button8,pos={1343,26},size={60,30},proc=ButtonProc_2editresults,title="Data Table"

GroupBox group0,pos={780,85},size={705,240},title="Iba Mask"

GroupBox group0,labelBack={39168,39168,39168}

Button button0,pos={798,293},size={50,20},proc=autoana,title="AutoANA"

Button button02,pos={868,293},size={50,20},proc=quantify,title="Copy cell"

variable /G smo=5

SetVariable setvar0,pos={943,297},size={78,16},bodyWidth=40,title="smooth"

SetVariable setvar0,value= smo

variable /G thr=30

SetVariable setvar1,pos={1034,297},size={87,16},bodyWidth=40,title="threshold"

SetVariable setvar1,value= thr, limits={0,inf,5}

CheckBox check_iba,pos={1137,299},size={57,14},title="dynamic",value= 0,proc=CheckProc_1_setdyn

SetVariable setvar14,pos={1328,289},size={121,16},bodyWidth=80,title="cell size"

SetVariable setvar14,format="%g voxels"

SetVariable setvar14,limits={-inf,inf,0},value= res_size,noedit= 1

```

Button button6,pos={1344,155},size={50,50},proc=ButtonProc_proj,title="Proj"

Button button7,pos={1354,220},size={100,50},proc=ButtonProc_useit,title="Mask"

CheckBox check44,pos={1347,131},size={43,14},title="New ",value= 1

```

```

GroupBox group1,pos={780,330},size={705,225},title="GABBR"

GroupBox group1,labelBack=(39168,39168,39168)

```

```

Button button3,pos={1375,487},size={50,50},proc=ButtonProc_2,title="Overlay"

Button button4,pos={1351,430},size={100,50},proc=checkScale,title="Check"

CheckBox check2,pos={1357,399},size={76,14},title="only somatic",value= 1

Button button5,pos={1410,155},size={50,50},proc=ButtonProc_projDAPI

Button button5 title="Crop\rNucleus"

CheckBox check0,pos={1356,374},size={97,14},proc=CheckProc_1,title="exclude Nucleus"

CheckBox check0,value= 1

```

NVAR gminA,gmaxA

```

GroupBox group6,pos={780,560},size={344,63},title="Iba1"

GroupBox group6,labelBack=(47872,47872,47872)

SetVariable setvar2,pos={788,578},size={60,16},bodyWidth=40,proc=SetVarProc_scaleA,title="min"

SetVariable setvar2,value= gminA

SetVariable setvar3,pos={785,599},size={63,16},bodyWidth=40,proc=SetVarProc_scaleA,title="max"

SetVariable setvar3,value= gmaxA

Slider slider0,pos={860,578},size={254,19},proc=SliderProc_scaleA

Slider slider0,limits={0,255,1},variable= gminA,vert= 0,ticks= 0

Slider slider1,pos={860,598},size={254,19},proc=SliderProc_scaleA

Slider slider1,limits={0,255,1},variable= gmaxA,vert= 0,ticks= 0

```

NVAR rminA,rmaxA

GroupBox group7,pos={780,628},size={344,80},title="GABBR"

GroupBox group7,labelBack=(47872,47872,47872)

SetVariable setvar4,pos={788,644},size={60,16},bodyWidth=40,proc=SetVarProc_scaleA,title="min"

SetVariable setvar4,value= rminA

SetVariable setvar5,pos={785,666},size={63,16},bodyWidth=40,proc=SetVarProc_scaleA,title="max"

SetVariable setvar5,value= rmaxA

Slider slider2,pos={860,645},size={254,19},proc=SliderProc_scaleA

Slider slider2,limits={0,255,1},variable= rminA,vert= 0,ticks= 0

Slider slider3,pos={860,665},size={254,19},proc=SliderProc_scaleA

Slider slider3,limits={0,255,1},variable= rmaxA,vert= 0,ticks= 0

Button button01,pos={909,684},size={100,20},proc=ButtonProc_3A,title="Threshold"

NVAR bminA,bmaxA

GroupBox group8,pos={780,710},size={344,63},title="DAPI"

GroupBox group8,labelBack=(47872,47872,47872)

SetVariable setvar2d,pos={789,731},size={60,16},bodyWidth=40,proc=SetVarProc_scaleA,title="min"

SetVariable setvar2d,value= bminA

SetVariable setvar3d,pos={786,751},size={63,16},bodyWidth=40,proc=SetVarProc_scaleA,title="max"

SetVariable setvar3d,value= bmaxA

Slider slider2d,pos={860,730},size={254,19},proc=SliderProc_scaleA

Slider slider2d,limits={0,255,1},variable= bminA,vert= 0,ticks= 0

Slider slider3d,pos={860,750},size={254,19},proc=SliderProc_scaleA

Slider slider3d,limits={0,255,1},variable= bmaxA,vert= 0,ticks= 0

wave allcell,cm1all

```
    if(!waveexists(allcell))
        make/O/N=(2,2) allcell
    endif

    if(!waveexists(cm1all))
        make/O/N=(2,2,3) cm1all
        redimension/B/U cm1all
    endif

    Display/W=(17,86,768,785)/HOST=#

    AppendImage/T cm1all


    ModifyGraph margin=-1

    ModifyGraph mirror=0

    ModifyGraph nticks=0

    ModifyGraph noLabel=2

    ModifyGraph standoff=0

    ModifyGraph axThick=0

    SetAxis/A/R left

    RenameWindow #,G0

    SetActiveSubwindow ##
```

wave cell

```
    if(!waveexists(cell))
        make/O/N=(2,2) cell
    endif

    Display/W=(795,125,945,275)/HOST=# ///mask, 1st

    AppendImage/T cell

    ModifyImage cell ctab= {*,*,Green,0}

    ModifyGraph margin=-1

    ModifyGraph mirror=0

    ModifyGraph nticks=0

    ModifyGraph noLabel=2
```

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,G1

SetActiveSubwindow ##

wave cellsmod

if(!waveexists(cellsmod))

make/O/N=(2,2) cellsmod

wave cellsmod

endif

Display/W=(975,125,1125,275)/HOST=# //mask,2nd

AppendImage/T cellsmod

ModifyImage cellsmod ctab= {10,200,Green,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,G3

SetActiveSubwindow ##

wave cell2d

if(!waveexists(cell2d))

make/O/N=(2,2) cell2d

wave cell2d

endif

Display/W=(1155,125,1305,275)/HOST=# //mask,3rd

AppendImage/T cell2d

```

ModifyImage cell2d ctab= {*,*,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,G2

SetActiveSubwindow ##

///

wave all_vglut

    if(!waveexists(all_vglut))

        make/O/N=(2,2) all_vglut

    endif


Display/W=(795,360,945,510)/HOST=#

AppendImage/T all_vglut

ModifyImage all_vglut ctab= {0,*,Red,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,G4

SetActiveSubwindow ##


wave s_vglut1,s_vglut2

    if(!waveexists(s_vglut1))

```

```

        make/O/N=(2,2) s_vglut1

endif

Display/W=(975,360,1125,510)/HOST=#

AppendImage/T S_vglut1

ModifyImage S_vglut1 ctab= {10,100,Red,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,G6

SetActiveSubwindow ##


Display/W=(1155,360,1305,510)/HOST=#

AppendImage/T S_vglut2

ModifyImage S_vglut2 ctab= {10,100,Red,0}

ModifyImage S_vglut2 maxRGB=NaN

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,G9

SetActiveSubwindow ##

wave h_o,h_c

Display/W=(1136,567,1483,778)/HOST=# h_o,h_c

ModifyGraph wbRGB=(43520,43520,43520)

```

```

ModifyGraph mode(h_o)=5

ModifyGraph lSize(h_c)=2

ModifyGraph rgb(h_o)=(34816,34816,34816),rgb(h_c)=(49152,1280,256)

ModifyGraph hbFill=2

ModifyGraph nticks(left)=0

ModifyGraph noLabel(left)=2

SetAxis bottom 0,70

Legend/C/N=text0/J "\\s(h_o) GABBR1 extracell.\r\\s(h_c) GABBR1 intracell."

SetDrawLayer UserFront

SetDrawEnv xcoord= bottom,ycoord= prel,linethick= 2

DrawLine 25.6613517618439,0,25.6613517618439,1

SetDrawEnv xcoord= bottom,ycoord= prel,linethick= 2,linefgc= (40000,0,0)

DrawLine 23.7168012957503,0,23.7168012957503,1

RenameWindow #,G5

SetActiveSubwindow ##

wave w_iba

if(waveexists(w_iba))

    if(dimsz(W_iba,0)>5)

        imagetransform zprojection w_iba

wave m_zprojection

        duplicate/O m_zprojection, allcell

    endif

endif

End

Function ButtonProc(ctrlName) : ButtonControl

    String ctrlName

    f_projRGB()

End

```

```
//////////
```

```
function analyse(lasch):ButtonControl
```

```
string lasch
```

```
//      quantify("")
```

```
getmarquee /W=panel0#G0 left,top
```

```
variable vl1
```

```
variable vt1
```

```
variable vr1
```

```
variable vb1
```

```
if(!V_flag)//no marquee
```

```
wave cell
```

```

string n1=note(cell)

        vl1=str2num(stringfromlist(0,n1,","))

        vt1=str2num(stringfromlist(2,n1,","))

        vr1=str2num(stringfromlist(1,n1,","))

        vb1=str2num(stringfromlist(3,n1,","))

    else

        vl1=v_left

        vt1=v_top

        vr1=v_right

        vb1=v_bottom

    endif

    if(stringmatch(lasch,"*4"))//vglut

        qSpots(vl1,vt1,vr1,vb1,"glu")

    else

        qSpots(vl1,vt1,vr1,vb1,"gab")

    endif

end

```

```

function addspots(s_win,gg)

```

```

string s_win

```

```

string gg

```

```

        if(stringmatch(gg,"glu"))

```

```

wave Ox=:glutXo

```

```

wave Oy=:glutYo

```

```

wave Oz=:glutZo

```

```

wave Oz1=:glutZo1

```

```

wave lx=:glutXi

```

```

wave ly=:glutYi

```

```
wave lz=:glutZi  
wave lz1=:glutZi1  
wave nx=:glutxn  
wave ny=:glutYn  
wave nz=:glutZn  
wave nz1=:glutZn1
```

```
elseif(stringmatch(gg,"gab"))
```

```
wave Ox=:gabXo  
wave Oy=:gabYo  
wave Oz=:gabZo  
wave Oz1=:gabZo1  
wave lx=:gabXi  
wave ly=:gabYi  
wave lz=:gabZi  
wave lz1=:gabZi1  
wave nx=:gabxn  
wave ny=:gabYn  
wave nz=:gabZn  
wave nz1=:gabZn1
```

```
endif
```

```
variable r1=0  
variable g1=26112  
variable b1=65535  
variable r2=8960  
variable g2=39680  
variable b2=51456
```



```

appendtograph/W=$(s_win)#Gyz/L/T Oy vs Oz1

appendtograph/W=$(s_win)#Gyz/L/T ly vs lz1

appendtograph/W=$(s_win)#Gyz/L/T ny vs nz1

ModifyGraph/W=$(s_win)#Gyz rgb($(nameofwave(iy)))=(r1,g1,b1)

ModifyGraph/W=$(s_win)#Gyz rgb($(nameofwave(oy)))=(r2,g2,b2)

ModifyGraph/W=$(s_win)#Gyz mode=3,marker=19

ModifyGraph/W=$(s_win)#Gyz rgb($(nameofwave(ny)))=(56576,56576,56576)

ModifyGraph/W=$(s_win)#Gyz marker($(nameofwave(ny)))=8,msize($(nameofwave(ny)))=0.5

```

```

appendtograph/W=$(s_win)#Gxy/L/T Oy vs Ox

appendtograph/W=$(s_win)#Gxy/L/T ly vs lx

appendtograph/W=$(s_win)#Gxy/L/T ny vs nx

ModifyGraph/W=$(s_win)#Gxy rgb($(nameofwave(iy)))=(r1,g1,b1)

ModifyGraph/W=$(s_win)#Gxy rgb($(nameofwave(oy)))=(r2,g2,b2)

ModifyGraph/W=$(s_win)#Gxy mode=3,marker=19

ModifyGraph/W=$(s_win)#Gxy rgb($(nameofwave(ny)))=(56576,56576,56576)

ModifyGraph/W=$(s_win)#Gxy marker($(nameofwave(ny)))=8,msize($(nameofwave(ny)))=0.5

```

```

appendtograph/W=$(s_win)#Gxz/L/T Oz vs Ox

appendtograph/W=$(s_win)#Gxz/L/T iz vs ix

appendtograph/W=$(s_win)#Gxz/L/T nz vs nx

ModifyGraph/W=$(s_win)#Gxz rgb($(nameofwave(iz)))=(r1,g1,b1)

ModifyGraph/W=$(s_win)#Gxz rgb($(nameofwave(oz)))=(r2,g2,b2)

ModifyGraph/W=$(s_win)#Gxz mode=3,marker=19

ModifyGraph/W=$(s_win)#Gxz rgb($(nameofwave(nz)))=(56576,56576,56576)

ModifyGraph/W=$(s_win)#Gxz marker($(nameofwave(nz)))=8,msize($(nameofwave(nz)))=0.5

```

end

```

//display results

sp=umpnts$(spX)

redimension/D w_3dsm

w_3dsm/=(w_3dsm[p][q][r]>0)

wavestats/Q w_3dsm

fl=V_avg

end

//NVAR smoga1

//smooth (smoga1), gab1

wave w_vGat

duplicate/O/R=[vl1,vr1][vt1,vb1] w_vGat, gab1

redimension/B/U gab1

gab1*=cell2

imagetransform zprojection gab1

wave M_zProjection

duplicate/O M_zProjection, S_vgat1

//GABA

NVAR gab_min,gab_max

NVAR thrGA

imagethreshold/Q/M=(m1)/T=(thrGa)/I s_vgat1

```

```

wave M_imagethresh

    duplicate/O M_imagethresh, gabThR

    imageanalyzeParticles/A=(gab_min)/MAXA=(gab_max)/Q stats    M_imagethresh

wave w_spotx,w_spoty

    w_spotx+=dimoffset(s_vgat1,0)

    w_spoty+=dimoffset(s_vgat1,1)

    duplicate/O w_spotx, gabx

    duplicate/O w_spoty, gaby

NVAR gaba_fl

NVAR gaba_sp

gaba_sp=numpnts(gabX)

redimension/D gab1

    gab1/=(gab1[p][q][r]>0)

    wavestats/Q gab1

    gaba_fl=V_avg

    storeDATA(vl1,vr1,vt1,vb1)

    killwaves/Z M_imagethresh,M_zProjection

end

```

```

Function ButtonProc_1b(ctrlName) : ButtonControl

    String ctrlName

wave w_iba=:glu1

nvar thr

make/O/N=(dimsize(w_iba,2)) dx1,dy1

variable p1

    for(p1=0; p1<(dimsize(w_iba,2)); p1+=1)

        imagertransform/P=(p1) getplane w_iba

wave M_ImagePlane

```

```

// imgethreshold/M=0/T=(thr)/O M_ImagePlane

wvstats/Q M_ImagePlane

dx1[p1]=p1

dy1[p1]=v_sum


endfor


dowindow/K w_distr

display /K=1/N=w_distr dy1 vs dx1

// edit dx1, dy1

End

```

```

function restore_xyw(str1)

string str1

string x1=stringbykey("xw", str1, ":", "\r")

string y1=stringbykey("yw", str1, ":", "\r")

```

```

make/O/N=0 xw,yw

variable p1

for(p1=0;p1<itemsinlist(x1,";");p1+=1)

    insertpoints (numpts(xw)),1, xw

    insertpoints (numpts(yw)),1, yw

    xw[numpts(xw)-1]=str2num(stringfromlist(p1,x1,";"))

    yw[numpts(yw)-1]=str2num(stringfromlist(p1,y1,";"))

endfor

end

function/S getcells()

    if(!datafolderexists("cells"))

        return ""

    endif

    setdatafolder root:cells

    string cl=replacestring(" ",stringbykey("FOLDERS",datafolderdir(1),":",";")) //data folders

    setdatafolder root:

    return cl

end

```

```

function fGraph0(lasch) :buttoncontrol

```

string lasch

string n1,n2

```
if(stringmatch (lasch,"glu*"))
    n1="glut"
    n2="S_vglut1"
elseif (stringmatch (lasch,"gab*"))
    n1="gab"
    n2="S_vgat1"
endif
```

dowindow/K \$(lasch)

PauseUpdate; Silent 1 // building window...

Display/K=1 /W=(177,71.75,963.75,761.75)/T/N=\$(lasch) \$(n1+"yo") vs \$(n1+"Xo") as n1

appendtograph/T \$(n1+"yi") vs \$(n1+"Xi")

appendtograph/T \$(n1+"yn") vs \$(n1+"Xn")

AppendImage/T \$(n2)

ModifyImage \$(n2) ctab= {*,*,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mode=3

ModifyGraph marker=19

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

```

ModifyGraph msize($(n1+"yn"))=0.5

ModifyGraph marker($(n1+"yn"))=8

ModifyGraph rgb($(n1+"yn"))=(34816,34816,34816)

ModifyGraph msize($(n1+"yi"))=3

ModifyGraph marker($(n1+"yi"))=16

ModifyGraph rgb($(n1+"yi"))=(0,52224,52224)

ModifyGraph marker( $(n1+"yo"))=19

ModifyGraph msize( $(n1+"yo"))=3

ModifyGraph rgb( $(n1+"yo"))=(0,26112,26112)

```

```

SetAxis/A/R left

```

```

CheckBox $(n1+"A"),pos={21,31},size={68,14},proc=CheckProc,title=n1+ " spot A "

CheckBox $(n1+"A"),value= 1

CheckBox $(n1+"N"),pos={21,11},size={69,14},proc=CheckProc,title=n1+ " spot N "

CheckBox $(n1+"N"),value= 1

CheckBox $(n1+"I"),pos={21,51},size={64,14},proc=CheckProc,title=n1+ " spot I "

CheckBox $(n1+"I"),value= 1

```

```

EndMacro

```

```

function fillit1(var,x,y,z,r)

```

```

variable var

```

```

variable x,y,r

```

variable zr

wave M_3DParticleInfo

variable x1=M_3DParticleInfo[var][0]-xyr

variable x2=M_3DParticleInfo[var][1]+xyr

variable y1=M_3DParticleInfo[var][2]-xyr

variable y2=M_3DParticleInfo[var][3]+xyr

variable z1=M_3DParticleInfo[var][4]-zr

variable z2=M_3DParticleInfo[var][5]+zr

wave cell2//idealised (thresholded) 3d iba img

duplicate/O/R=[x1,x2][y1,y2][z1,z2] cell2, ts

end

function fillit(x1,y1,z1,xyr,zr)

variable x1//midpoint of a spot x

variable y1//midpoint of a spot y


```
variable z1//midpoint of a spot z
```

```
variable xyr//range xy
```

```
variable zr//range z
```

```
//print x1,y1,z1,xyr,zr
```

```
wave ts
```

```
wave W_iba=root:cellsmo//W_iba//cell2
```

```
variable xoff=dimoffset(w_iba,0)
```

```
variable yoff=dimoffset(w_iba,1)
```

```
variable p1,q1,r1
```

```
    for(p1=0; p1<xyr;p1+=1)
```

```
        for(q1=0; q1<xyr;q1+=1)
```

```
            for(r1=0; r1<zr;r1+=1)
```

```
                ts[p1][q1][r1]=w_iba[x1-(xyr-1)/2+p1-xoff][y1-(xyr-1)/2+q1-yoff][z1-(zr-1)/2+r1]
```

```
                //print x1-(xyr-1)/2+p1-xoff,y1-(xyr-1)/2+q1-yoff,z1-(zr-1)/2+r1
```

```
            endfor
```

```
        endfor
```

```
    endfor
```

```
end
```

Function CheckProc_1ideal(ctrlName,checked) : CheckBoxControl

String ctrlName//0=gaba,1=glut

Variable checked

if(checked)

if(stringmatch(ctrlName[5],"0"))//GABA

addspots("w_projRGB","gab")

else

addspots("w_projRGB","glu")

endif

else

if(stringmatch(ctrlName[5],"1"))//glut

removefromgraph/Z/W=w_projRGB#Gyz glutyo,glutyi,glutyn

removefromgraph/Z/W=w_projRGB#Gxy glutyo,glutyi,glutyn

removefromgraph/Z/W=w_projRGB#Gxz glutzo,glutzi,glutzn

else //GABA

removefromgraph/Z/W=w_projRGB#Gyz gabyo,gabyi,gabyn

removefromgraph/Z/W=w_projRGB#Gxy gabyo,gabyi,gabyn

removefromgraph/Z/W=w_projRGB#Gxz gabzo,gabzi,gabzn

endif

endif

End

```
Function CheckProc_1_setdyn(ctrlName,checked) : CheckBoxControl
```

```
    String ctrlName
```

```
    Variable checked
```

```
    string setvar
```

```
        if(stringmatch(ctrlName,"*iba"))
```

```
    NVAR thr=root:thr
```

```
        setvar="setvar1"
```

```
        elseif(stringmatch(ctrlName,"*glu"))
```

```
    NVAR thr=root:thrGL
```

```
        setvar="setvar3"
```

```
        elseif(stringmatch(ctrlName,"*gab"))
```

```
    NVAR thr=root:thrGA
```

```
        setvar="setvar4"
```

```
    endif
```

```
    if(checked)
```

```
        thr=3
```

```
        SetVariable $(setvar), win=Panel0 ,format="%g *avg ",limits={0,20,1}
```

```
    else
```

```
        thr=50
```

```
        SetVariable $(setvar), win=Panel0 ,format="",limits={0,255,5}
```

```
    endif
```

```
End
```

```

function showsignal(w1,var)

wave w1

variable var

variable p1

make/O/N=(dimsize(w1,2)) signal

    for(p1=0; p1<(dimsize(w1,2)); p1+=1)

        imagetransform/P=(p1) getplane w1

wave M_ImagePlane

        wavestats/Q M_ImagePlane

        signal[p1]=v_avg

    endfor

killwaves/Z m_imageplane

if(var)

    dowindow/K w_signal

    display/K=1/N=W_signal      signal as "z signal of "+nameofwave(w1)

endif

end

function thresh3d (w1,GG)

wave w1

```

```

string GG// glu or GABA

//print nameofwave(w1),dimsize(w1,0),dimsize(w1,1),gg,"Y"

string spX=""

string spY=""

string spz=""

        if(stringmatch (GG, "glu")) //vglut staining

NVAR thr=:thrGL //manual thr value

NVAR min1=:glu_min

NVAR max1=:glu_max

                spX="glutx"

                spY="glutY"

                spz="glutz"

wave W_gg=:w_vglut

        elseif(stringmatch (GG, "gab")) //vgat staining

NVAR thr=:thrGA //manual thr value

NVAR min1=:gab_min

NVAR max1=:gab_max

                spX="gabx"

                spY="gabY"

                spz="gabz"

wave W_gg=:w_vgat

        endif


controlinfo/W=panel0 $("check_" +gg)

variable dyn=0

        if(v_value)//checked

                dyn=1


wave signal=:$("signal_" +gg)

        else

```

```

        //print gg,"nope"

    endif

//print thr,gg

    duplicate/O w1, test44,test45

    test44=0

    test45=0

variable p1

    for(p1=0; p1<(dimsize(w1,2)); p1+=1)

        imagertransform/P=(p1) getplane w1

wave M_ImagePlane

wave signal

        wavestats/Q W_gg

        if(dyn)

            //print p1, thr,signal[p1]

            imagerthreshold/Q/T=(thr*signal[p1])/M=0/I M_ImagePlane

            //print p1,thr,signal[p1],thr*signal[p1]

        else

            thr=3*ceil(v_avg)

            imagerthreshold/Q/T=(thr)/M=0/I M_ImagePlane

        endif

wave M_imagerthresh

    M_ImagePlane*=M_imagerthresh

    imagertransform/P=(p1)/D=M_ImagePlane setplane test44

    imagertransform/P=(p1)/D=M_imagerthresh setplane test45

endfor

end

```

```

Function ButtonProc_1(ctrlName) : ButtonControl

    String ctrlName

    wave w_iba=root:w_iba

    nvar thr

    make/O/N=(dimsize(w_iba,2)) dx1,dy1

    variable p1

        for(p1=0; p1<(dimsize(w_iba,2)); p1+=1)

            imagetransform/P=(p1) getplane w_iba

        wave M_ImagePlane

            imagethreshold/Q/M=0/T=(thr)/O M_ImagePlane

            wavestats/Q M_ImagePlane

            dx1[p1]=p1

            dy1[p1]=v_sum/255

        endfor

    dowindow/K w_distr

    display /K=1/N=w_distr dy1 vs dx1

    dowindow/K t_distr

    edit/K=1/N=t_distr dx1, dy1

```

End

```
function getsignal(cell1)
```

```
wave cell1
```

```
duplicate/O cell1,test
```

```
wave test
```

```
redimension/S test
```

```
wavestats/Q test
```

```
variable t1=ceil(V_avg)
```

```
test/=(test[p][q]>3*t1)
```

```
showsignal(test,0)
```

```
wave signal
```

```
CurveFit/Q/L=(numpnts(signal))/NTHR=0 poly 3, signal /D
```

```
wave fit_signal
```

```
end
```

```
#pragma rtGlobals=3           // Use modern global access method and strict wave access.
```

```
function qSpots(vl1,vt1,vr1,vb1,GG)
```

```
variable vl1
```

```
variable vt1
```

```
variable vr1
```

```
variable vb1
```

```
string GG //either glu or gab
```



```
//variable m1=0 //method threshold; manual
```

```
NVAR xyr// xy range; odd number
```

```
NVAR zr//z range; odd number
```

```
string s_3DspotNN="" // name of cell-filtered 3D spot wave
```

```
string s_spotNN="" ///max projection of s_3DspotNN
```

```
string spX=""
```

```
string spY=""
```

```
string spz=""
```

```
if(stringmatch (GG, "glu")) //vglut staining
```

```
wave w_3Dspot=:w_vGlut//3d wave containing all spots
```

```
NVAR thr=:thrGL //manual thr value
```

```
s_3DspotNN="glu1" // name of cell-filtered 3D spot wave
```

```
s_spotNN="S_vglut1" ///max projection of s_3DspotNN
```

```
NVAR min1=:glu_min
```

```
NVAR max1=:glu_max
```

```
spX="glutx"
```

```
spY="gluty"
```

```
spz="glutz"
```

```
NVAR smo1=:smogl1
```

```
make/O/N=0 glutXo,glutYo,glutZo,glutZo1
```

```
make/O/N=0 glutXi,glutYi,glutZi,glutZi1
```

```
make/O/N=0 glutXn,glutYn,glutZn,glutZn1
```

```
wave Ox=:glutXo
```

```
wave Oy=:glutYo
```

```
wave Oz=:glutZo
```

```
wave Oz1=:glutZo1
```

```
wave lx=:glutXi  
wave ly=:glutYi  
wave lz=:glutZi  
wave lz1=:glutZi1  
NVAR fl=:glut_fl  
NVAR sp=:glut_sp  
NVAR spN=:glut_spN  
NVAR spA=:glut_spA  
NVAR spl=:glut_spl  
wave nx=:glutxn  
wave ny=:glutyn  
wave nz=:glutzn  
wave nz1=:glutzn1
```

```
elseif(stringmatch (GG, "gab")) //vgat staining  
wave w_3Dspot=:w_vGat//3d wave containing all spots  
NVAR thr=:thrGA //manual thr value  
s_3DspotNN="gab1" // name of cell-filtered 3D spot wave  
s_spotNN="S_vgat1" ///max projection of s_3DspotNN
```

```
NVAR min1=:gab_min  
NVAR max1=:gab_max  
spX="gabx"  
spY="gaby"  
spz="gabz"  
NVAR smo1=:smoga1
```

```

make/O/N=0 gabXo,gabYo,gabZo,gabZo1

make/O/N=0 gabXi,gabYi,gabZi,gabZi1

make/O/N=0 gabXn,gabYn,gabZn,gabZn1

wave Ox=:gabXo

wave Oy=:gabYo

wave Oz=:gabZo

wave Oz1=:gabZo1

wave lx=:gabXi

wave ly=:gabYi

wave lz=:gabZi

wave lz1=:gabZi1

NVAR fl=:gaba_fl

NVAR sp=:gaba_sp

NVAR spN=:gaba_spN

NVAR spA=:gaba_spA

NVAR spl=:gaba_spl

wave nx=:gabxn

wave ny=:gabyn

wave nz=:gabzn

wave nz1=:gabzn1

endif


//      print      gg,dimoffset(w_3dsm,0), dimoffset(w_3dsm,1),nameofwave (w_3dsm)//seedx

wave cell2 ///mask; IBA1-based; 3D, single cell

duplicate/O/R=[vl1,vr1][vt1,vb1] w_3Dspot, $(s_3DspotNN)      //W_vglut into glu1(w_3dsm)

wave w_3dsm=$(s_3DspotNN) ///3D wave spots; mask; glu1

//dynamic

controlinfo/W=panel0 $("check_" +gg)

variable dyn=0

```

```

        if(v_value)//checked

            dyn=1

            showsignal(w_3dsm,0)

wave signal

            duplicate/O signal, $("signal_"+gg)

        endif

variable scale=wavemax(w_3dsm)

//      w_3dsm/=(scale/255)

        redimension/B/U w_3dsm //glu1

            //smooth

        if(smo1)

            //print smo1,"F"

            imagefilter/N=(smo1)/O gauss3d w_3dsm

        endif

        thresh3d (w_3dsm,GG)

wave test45

        imageanalyzeParticles/A=(min1)/MAXA=(max1) stats test45

wave M_3DParticleInfo

        w_3dsm*=cell2 //masking

        imagetransform zprojection w_3dsm

wave M_zProjection

        duplicate/O M_zProjection, $(s_spotNN) //s_vglut1

wave w_sm1=$(s_spotNN)      //max proj of 3D masked spot wave

//      setscale/P x 0,1, w_sm1

//      setscale/P y 0,1, w_sm1

```

```

make/O/N=(dimsize(M_3DParticleInfo,0)) $(spx),$(spy),$(spz)

wave spX1=$(spX)//midpoint of spot

wave spy1=$(spY)//midpoint of spot

wave spz1=$(spZ)//midpoint of spot

spX1=M_3DParticleInfo[p][0]+(M_3DParticleInfo[p][1]-M_3DParticleInfo[p][0])/2

spY1=M_3DParticleInfo[p][2]+(M_3DParticleInfo[p][3]-M_3DParticleInfo[p][2])/2

spZ1=M_3DParticleInfo[p][4]+(M_3DParticleInfo[p][5]-M_3DParticleInfo[p][4])/2

spX1+=dimoffset(w_3dsm,0)

spY1+=dimoffset(w_3dsm,1)


//check spots

variable p1

for(p1=0; p1<numpts(spz1); p1+=1)

    fillIT1(p1,x,y,z)

wave ts

wavestats/q ts

if(!v_avg) ///no touch

    insertpoints (numpts(nx)),1, nx,ny,nz

    nx[numpts(nx)-1]=spX1[p1]

    ny[numpts(ny)-1]=spy1[p1]

    nz[numpts(nz)-1]=spz1[p1]

elseif(v_avg<0.8) ///out

    insertpoints (numpts(Ox)),1, ox,oy,oz

    ox[numpts(ox)-1]=spX1[p1]

    oy[numpts(oy)-1]=spy1[p1]

    oz[numpts(oz)-1]=spz1[p1]

elseif(v_avg>=0.8) //in

    insertpoints (numpts(ix)),1, ix,iy,iz

    ix[numpts(ix)-1]=spX1[p1]

    iy[numpts(iy)-1]=spy1[p1]

```

```

            iz[numpnts(iz)-1]=spz1[p1]

        endif

    endfor

//scaling stuff

    duplicate/o oz, $(nameofwave(oz1))

    duplicate/o iz, $(nameofwave(iz1))

    duplicate/o nz, $(nameofwave(nz1))

    if(numpnts(oz1))

        oz1=dimsize(w_3dsm,2)-oz

    endif

    if(numpnts(iz1))

        iz1=dimsize(w_3dsm,2)-iz

    endif

    if(numpnts(nz1))

        nz1=dimsize(w_3dsm,2)-nz

    endif

variable xoff=dimoffset(cell2,0)

variable yoff=dimoffset(cell2,1)


//results

sp=numpnts$(spX)

spN=numpnts(nX)

spA=numpnts(oX)

spl=numpnts(iX)

```

duplicate/O w_3dsm, test

redimension/D test

test/=(test[p][q][r]>0)

wavestats/Q test

fl=V_avg

killwaves/Z test

end

Function ButtonProc_restore(ctrlName) : ButtonControl

String ctrlName

if(stringmatch(ctrlName,"*dapi"))

wave cell1:=all_DAPI2

else

wave cell1:=cell1

endif

imagetransform/METH=1 xprojection cell1

wave M_xprojection

 imagetransform/METH=1 zprojection cell1

wave M_zprojection

 imagetransform/METH=1 yprojection cell1

wave M_yprojection

 if(stringmatch(ctrlName,"*dapi"))

 duplicate/O M_xprojection, dapiX

 duplicate/O M_yprojection, dapiY

 duplicate/O M_zprojection, dapiZ

 ImageRotate/A=90/O dapiX

 else

 duplicate/O M_xprojection, cellX

 duplicate/O M_yprojection, cellY

 duplicate/O M_zprojection, cellZ

 ImageRotate/A=90/O cellX

 endif

killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh

 if(stringmatch(ctrlName,"*dapi"))

 duplicate/O cell1, temp44dapi

 else

 duplicate/O cell1, temp44//temp44=cell1*mask

 endif

End


```
#pragma rtGlobals=3          // Use modern global access method and strict wave access.
```

```
function genMask(cell1)
```

```
wave cell1 ///n layers
```

```
variable nl=dimsize(cell1,2)//number of layers
```

```
NVAR thr
```

```
NVAR smo
```

```
variable p1
```

```
duplicate/O cell1, cell2,cellsmo
```

```
cell2=0
```

```
if(smo>0)
```

```
for(p1=0;p1<smo;p1+=1)
```

```
Imagefilter/N=(3)/O gauss3d cellsmo
```

```
endfor
```

```
endif
```

```
controlinfo/W=panel0 check_iba
```

```
variable dyn=0
```

```
if(v_value)//checked
```

```
dyn=1
```

```
wave W_iba
```

```
showsignal(w_iba,0)
```

```
wave signal
```

```

endif

for(p1=0;p1<nl;p1+=1)

    imagetransform/P=(p1) getplane cellsmo

wave M_ImagePlane

    if(dyn)

        imagethreshold/Q/T=(thr*signal[p1])/M=0 M_ImagePlane

    else

        imagethreshold/Q/T=(thr)/M=0 M_ImagePlane

    endif

wave m_imagethresh

    imagetransform/P=(p1)/D=m_imagethresh setplane cell2

endfor

killwaves/Z signal

//ImageSeedFill/B=0 seedX=526,seedY=327,seedZ=25,target=255,min=30,max=255,fillnumber=1,srcWave=cell1

cell2/=255

controlinfo/W=Panel0 check0

if(V_value)//nucleus excluded

wave mask_dapi

    if(waveexists(mask_dapi))

        cell2*=(mask_dapi)

        wavestats/Q          mask_dapi

variable/G dapi_ex=1

variable/G nuc_size= v_npnts-v_sum

```

```

else

variable/G dapi_ex=0

variable/G nuc_size=nan

endif

else

variable/G dapi_ex=0

variable/G nuc_size=nan

endif


imagetransform zprojection cell2

wave M_zProjection

duplicate/O M_zProjection, cell2d


imagetransform zprojection cellsmo

wave M_zProjection

duplicate/O M_zProjection, cellsmod


wave glu1

imagetransform zprojection glu1

wave M_zProjection

duplicate/O M_zProjection, all_vglut

redimension/B/U all_vglut

//middle img

NVAR smo

duplicate/O glu1, glu2


wave temp44

```

```

        if(waveexists(temp44))

            glu2*=(temp44>0)

        endif

        if(smo>0)

            for(p1=0;p1<smo;p1+=1)

                Imagefilter/N=(3)/O gauss3d glu2

            endfor

        endif

        imagetransform zprojection glu2

        duplicate/O M_zProjection, S_vglut1


//right img

        duplicate/O glu2, glu2M

wave cell2

        glu2m*=cell2

        imagetransform zprojection glu2m

        duplicate/O M_zProjection, S_vglut2


//////////DAPI

wave celldapi

        imagetransform zprojection celldapi


        duplicate/O M_zProjection, all_dapi

        redimension/B/U all_dapi


//middle img

NVAR smo

        duplicate/O celldapi, all_dapi2

```

```
wave mask44
```

```
    if(waveexists(mask44))
```

```
        all_dapi2*=mask44
```

```
    endif
```

```
    if(smo>0)
```

```
        for(p1=0;p1<smo;p1+=1)
```

```
            Imagefilter/N=(3)/O gauss3d all_dapi2
```

```
        endfor
```

```
    endif
```

```
    imagetransform zprojection all_dapi2
```

```
    duplicate/O M_zProjection, all_dapiP2
```

```
//Right
```

```
//wave mask44dapi
```

```
    duplicate/O all_dapi2, all_dapi2m
```

```
    if(waveexists(mask_dapi))
```

```
        all_dapi2m*=(mask_dapi==0)
```

```
    endif
```

```
    if(smo>0)
```

```
        for(p1=0;p1<smo;p1+=1)
```

```
            Imagefilter/N=(3)/O gauss3d all_dapi2m
```

```
        endfor
```

```
    endif
```

```
    imagetransform zprojection all_dapi2m
```

```
    duplicate/O M_zProjection, all_dapi2mp
```

end

Function ButtonProc_useit(ctrlName) : ButtonControl

String ctrlName

wave temp44, temp44dapi

if(stringmatch(ctrlName,"*dapi"))

wave dapi=:celldapi

duplicate/O dapi, temp45dapi

NVAR smo

if(smo)

variable p1

for(p1=0;p1<9;p1+=1)

imagefilter/N=3/O gauss temp45dapi

endfor

endif

imagethreshold/M=1/O/Q temp45dapi

temp45dapi/=255

temp44dapi*=temp45dapi

temp44dapi=(temp44dapi>0)

temp44dapi=(temp44dapi==0)

killwaves/Z temp45dapi

duplicate/O temp44dapi,mask44dapi

// mask44dapi=(mask44dapi==0)

```

        duplicate/O mask44dapi,mask_dapi

        checkbox check0,win=panel0, value=1

    else

        duplicate/O temp44,mask44

        mask44=(mask44>0)

    endif

    genMask(temp44)

NVAR res_size

wave cell2

    wavestats/Q cell2

    res_size=v_sum

End

```

```

Function ButtonProc_4Ibmask(ctrlName) : ButtonControl

```

```

    String ctrlName

    wave mask=:cell2//iba mask

    wave curw=:all_DAPI2//without mask

```

```

        duplicate/O curw,temp44dapi

    wave cell1=:temp44dapi

```

```

    cell1*=(mask>0)

```

```

    imagertransform/METH=1 xprojection cell1

```

```
wave M_xprojection
```

```
    imagetransform/METH=1 zprojection cell1
```

```
wave M_zprojection
```

```
    imagetransform/METH=1 yprojection cell1
```

```
wave M_yprojection
```

```
    duplicate/O M_xprojection, dapiX
```

```
    duplicate/O M_yprojection, dapiY
```

```
    duplicate/O M_zprojection, dapiZ
```

```
    ImageRotate/A=90/O dapiX
```

```
killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh
```

```
End
```

```
function sc1()
```



```
wave cm1
```

```
wave cm1
```

```
wave pl0,pl1,pl2
```

```
controlinfo/W=panel0 check2
```

```
if(V_value)
```

```
wave tmp44
```

```
    if(waveexists(tmp44))
```

```
        imagetransform/METH=3 zprojection tmp44
```

```
wave m_zProjection
```

```
    pl0*=(m_zProjection==0)
```

```
    pl1*=(m_zProjection==0)
```

```
    pl2*=(m_zProjection==0)
```

```
    killwaves/Z M_zprojection
```

```
    endif
```

```
endif
```

```
imagetransform/P=0/D=pl0 setplane cm1
```

```
imagetransform/P=1/D=pl1 setplane cm1
```

```
imagetransform/P=2/D=pl2 setplane cm1
```

```
end
```

```
function makeRGB(doit)
```

```
variable doit
```

```
wave gc=cellsmod///IBA
```

```
controlinfo/W=W_scale check0///GABBR
```

```
    if(v_value)
```

```
    wave rc=s_vglut2
```

```
    else
```

```
    wave rc=s_vglut1
```

```
    endif
```

```
controlinfo/W=W_scale check1//DAPI
```

```
    if(v_value)
```

```
    wave bc=all_dapi2mp
```

```
    else
```

```
    wave bc=all_dapip2
```

```
    endif
```

```
duplicate/O rc,pl0
```

```
duplicate/o gc,pl1
```

```
duplicate/o bc,pl2
```

```
if(doit>0)
```

```
make/O/N=(dimsize(pl0,0),dimsize(pl0,1),3)/b/U cm1=0
```

```
imagehistogram rc
```

```
wave W_imageHist
```

```
duplicate/O W_imageHist,h_rc
```

```
imagehistogram gc
```

```
duplicate/O W_imageHist,h_gc
```

```
        imagehistogram bc

        duplicate/O W_imageHist,h_bc

    endif

    scaleIT(0)

    scaleIT(1)

    scaleIT(2)

    sc1()

end
```

```
function scale_glob()

variable/G gmin,gmax=255

variable/G rmin,rmax=255

variable/G bmin,bmax=255

end
```

```
function scaleIT(plane)

variable plane

    if(plane==0)//red channel==GABBR

wave curw=:pl0

NVAR min1=root:rmin

NVAR max1=root:rmax
```

```

        elseif(plane==1)//green channel==iba1

wave curw=:pl1

NVAR min1=root:gmin

NVAR max1=root:gmax

        elseif(plane==2)//blue channel==DAPI

wave curw=:pl2

NVAR min1=root:bmin

NVAR max1=root:bmax

        endif

        //print min1,max1,nameofwave(curw),plane

redimension/S curw

curw-=min1

curw*=(curw[p][q]>=0)


curw/=(max1-min1)

curw=curw^(curw<1)

curw*=255


redimension/B/U curw

end

```

```
Function SliderProc_scale(ctrlName,sliderValue,event) : SliderControl
```

```
    String ctrlName
```

```
    Variable sliderValue
```

```
    Variable event          // bit field: bit 0: value set, 1: mouse down, 2: mouse up, 3: mouse moved
```

```
    makeRGB(0)
```

```
    return 0
```

```
End
```

```
Function SetVarProc_scale(ctrlName,varNum,varStr,varName) : SetVariableControl
```

```
    String ctrlName
```

```
    Variable varNum
```

```
    String varStr
```

```
    String varName
```

```
    makeRGB(0)
```

```
End
```

```
Function ButtonProc_2(ctrlName) : ButtonControl
```

```
    String ctrlName
```

```
    fScale()
```

```
End
```

```

redimension/N=(numpnts(cell2))/S glu2c

glu2c/=(glu2c[p]!=0)

wavestats/Q glu2c

print "cell:",v_npnts,v_avg,v_sdev

```

```

redimension/N=(numpnts(cell2))/S glu2o

glu2o/=(glu2o[p]>0)

wavestats/Q glu2o

print "outside:",v_npnts,v_avg,v_sdev

```

```

statsttest/T=1/Q glu2c,glu2o

wave w_statsttest

print "p =", w_statsttest[9]

```

```

make/O h_c,h_o

Histogram/B={0,1,255} glu2o,h_o

Histogram/B={0,1,255} glu2c,h_c

```

```

end

```

```

function deletenan(curw)

```

```

wave curw

```

```

variable p1

```

```

for(p1=numpts(curw);p1==0;p1-=1)
    if(curw[p1]<=0)
        deletepoints p1,1,curw
    endif
endfor

end

function fw_projDAPI()

    dowindow/K w_projDAPI

    PauseUpdate; Silent 1          // building window...

    NewPanel /W=(671,133,1443,905)/K=1 /N=w_projDAPI as "DAPI Projection"

    SetDrawLayer UserBack

    SetDrawEnv arrow= 1

    DrawLine 385,370,385,20

    SetDrawEnv arrow= 1

    DrawLine 400,385,750,385

    SetDrawEnv fsize= 18

    DrawText 755,395,"x"

```

```

SetDrawEnv fsize= 18

DrawText 381,770,"z"

SetDrawEnv dash= 1,arrow= 1

DrawLine 370,385,20,385

SetDrawEnv dash= 1,arrow= 1

DrawLine 385,400,385,750

SetDrawEnv fsize= 18

DrawText 10,395,"z"

SetDrawEnv fsize= 18

DrawText 381,17,"y"

GroupBox group0,pos={20,400},size={350,350},labelBack={34816,34816,34816}

//CheckBox check0,pos={36,418},size={43,14},proc=CheckProc_1_setw,title="Crop "

//CheckBox check0,value= 1

Button button0dapi,pos={100,450},size={150,40},proc=ButtonProc_cropDAPI,title="Delete"

Button button1dapi,pos={100,500},size={150,40},proc=ButtonProc_cropDAPI,title="Crop"

Button button4,pos={296,467},size={50,50},proc=ButtonProc_4lbamask,title="Iba mask"

Button button4,fColor=(0,39168,0)

Button button2dapi,pos={100,600},size={150,40},proc=ButtonProc_restore,title="Restore"

Button button3dapi,pos={100,650},size={150,40},proc=ButtonProc_useit,title="Use"

SetVariable setvar0,pos={121,417},size={108,16},bodyWidth=40,proc=SetVarProc_filter,title="Avg Img Filter"

SetVariable setvar0,limits={3,inf,2},value= imgF

```

wave dapiZ, dapiX,dapiY

```

Display/W=(400,20,750,370)/HOST=#

AppendImage/T dapiZ

ModifyImage dapiZ ctab= {*,*,Grays,0}

ModifyGraph margin=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

```


ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,Gxy

SetActiveSubwindow ##

Display/W=(20,20,370,370)/HOST=#

AppendImage/T dapiX

ModifyImage dapiX ctab= {*,*,Grays,0}

ModifyGraph margin=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

SetAxis/A top

RenameWindow #,Gyz

SetActiveSubwindow ##

Display/W=(400,400,750,750)/HOST=#

AppendImage/T dapiY

ModifyImage dapiY ctab= {*,*,Grays,0}

ModifyGraph margin=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,Gxz

```

SetActiveSubwindow ##

setwindow W_projDAPI, hook=hook_crop,hookevents=7

make/O/N=0 xw,yw

GraphWaveDraw /W={"W_projDAPI"}#{("Gyz")} /L/T/O yW, xW

End

```

```

Function ButtonProc_cropDAPI(ctrlName) : ButtonControl

```

```

    String ctrlName

    string rem="\t"

    wave dapiz

    string o1=note(dapiz)

    variable it1=itemsinlist(o1,"\t")

    rem+="Action_"+num2str(it1)+"\r"

    wave temp44dapi

    if(!waveexists(temp44dapi))

    wave all_DAPI2M

        duplicate/o all_DAPI2M, temp44dapi

    endif

    wave xw,yw

    string pr=note(xw)

    rem+=pr+"\r"      //1st row:window

    if(stringmatch(pr,"Gxy"))

    wave curw=root:dapiz

    endif

    if(stringmatch(pr,"Gxz"))

    wave curw=root:dapiY

```

```

endif

if(stringmatch(pr,"Gyz"))
wave curw=root:dapiX

endif

ImageBoundaryToMask
height=dimsize(curw,1),scalingWave=curw,seedy=dimoffset(curw,1),seedx=dimoffset(curw,0), xwave=xw, ywave=yw
width=dimsize(curw,0),

wave M_ROIMask

if(stringmatch (ctrlName,"*1dapi"))//crop

M_Roimask=(M_Roimask[p][q]==0)

rem+="crop_1\r"

else ///delete, as it is

rem+="delete_2\r"

endif

if(stringmatch(pr,"Gxy"))

temp44dapi*=m_roimask[p][q]

endif

if(stringmatch(pr,"Gxz"))

temp44dapi[][]*=m_roimask[p][r]

endif

if(stringmatch(pr,"Gyz"))

imagetransform fliprows m_roimask

temp44dapi[][]*=m_roimask[r][q]

endif

imagetransform zprojection temp44dapi

wave M_zprojection

imagetransform xprojection temp44dapi

wave M_xprojection

```

```

        imagetransform yprojection temp44dapi
wave M_yprojection
        duplicate/O M_zprojection, dapiZ
        duplicate/O M_xprojection, dapiX
        duplicate/O M_yprojection, dapiY
        ImageRotate/A=90/O dapix
        killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh

        rem+="xw:"+w2str(xw)+"\r"
        rem+="yw:"+w2str(yw)+"\r"

        note dapiz, o1+rem

End

```

```

Function CheckProc_2chowAna(ctrlName,checked) : CheckBoxControl

```

```

    String ctrlName

```

```

    Variable checked

```

```

    setdrawlayer/W=panel0#G0/K userfront

```

```

        if(checked)

string exp1=getcells()


variable p1

        for(p1=0;p1<itemsinlist(exp1,";");p1+=1)

SVAR marq=root:cells:$(stringfromlist(p1,exp1,";")):marq

variable l1=str2num(stringfromlist(0,marq,";"))

variable r1=str2num(stringfromlist(1,marq,";"))

variable t1=str2num(stringfromlist(2,marq,";"))

variable b1=str2num(stringfromlist(3,marq,";"))

                setdrawenv/W=panel0#G0 xcoord=top,ycoord=left,save

                SetDrawEnv/W=panel0#G0 fillpat= 0,linethick= 2.00,save

                SetDrawEnv/W=panel0#G0 linefgc= (65535,0,0),save

                drawrect/W=panel0#G0 l1,t1,r1,b1

                SetDrawEnv/W=panel0#G0 textrgb= (65535,0,0)

                drawtext/W=panel0#G0 l1,t1, stringfromlist(p1,exp1,";")

                //      print p1, stringfromlist(p1,exp1,";")

        endfor

endif

End

```

```

function fScale()

        scale_glob()

```

```

makeRGB(1)

wave cm1

dowindow/K W_scale

PauseUpdate; Silent 1          // building window...

Display /W=(340,90,1000,584)/N=W_scale/K=1 as " "

AppendImage/T cm1

ModifyImage cm1 ctab= {*,*,Grays,0}

ModifyGraph margin(left)=14,margin(bottom)=14,margin(top)=14,margin(right)=14

ModifyGraph mirror=2

ModifyGraph nticks=6

ModifyGraph minor=1

ModifyGraph noLabel=2

ModifyGraph fSize=8

ModifyGraph standoff=0

ModifyGraph axThick=0

ModifyGraph tkLblRot(left)=90

ModifyGraph btLen=3

ModifyGraph tlOffset=-2

ModifyGraph axisEnab(top)={0,0.7}

SetAxis/A/R left

NVAR gmin,gmax

GroupBox group2,pos={617,20},size={255,200},frame=0,fColor={13056,0,0}

GroupBox group0,pos={620,22},size={249,120},title="Iba1"

GroupBox group0,labelBack={47872,47872,47872}

SetVariable setvar0,pos={633,40},size={50,16},bodyWidth=40,title="min"

SetVariable setvar0,limits={-inf,inf,1},value=gmin,proc=SetVarProc_scale

Slider slider0,pos={623,60},size={244,19},variable=gmin

Slider slider0,limits={0,255,1},value= 0,vert= 0,ticks= 0,proc=SliderProc_scale

SetVariable setvar1,pos={633,90},size={50,16},bodyWidth=40,title="max"

SetVariable setvar1,limits={-inf,inf,1},value=gmax,proc=SetVarProc_scale

```

Slider slider1,pos={623,110},size={244,19}, variable=gmax

Slider slider1,limits={0,255,1},value= 255,vert= 0,ticks= 0,proc=SliderProc_scale

NVAR rmin,rmax

GroupBox group3,pos={617,230},size={255,200},frame=0,fColor={13056,26112,0}

GroupBox group1,pos={620,232},size={249,120},title="GABBR"

GroupBox group1,labelBack={47872,47872,47872}

SetVariable setvar2,pos={633,250},size={50,16},bodyWidth=40,title="min"

SetVariable setvar2,limits={-inf,inf,1},value=rmin,proc=SetVarProc_scale

Slider slider2,pos={623,270},size={244,19},variable =rmin

Slider slider2,limits={0,255,1},value= 0,vert= 0,ticks= 0,proc=SliderProc_scale

Button button0,pos={768,251},size={50,20},proc=ButtonProc_3,title="Threshold"

SetVariable setvar3,pos={633,300},size={50,16},bodyWidth=40,title="max"

SetVariable setvar3,limits={-inf,inf,1},value=rmax,proc=SetVarProc_scale

Slider slider3,pos={623,320},size={244,19},variable=rmax

Slider slider3,limits={0,255,1},value= 255,vert= 0,ticks= 0,proc=SliderProc_scale

CheckBox check0,pos={769,357},size={95,14},title="Only intracellular",value= 0, proc=check_scale

NVAR bmin,bmax

GroupBox group4,pos={617,440},size={255,200},frame=0,fColor={13056,26112,0}

GroupBox group5,pos={620,442},size={249,120},title="DAPI"

GroupBox group5,labelBack={47872,47872,47872}

SetVariable setvar2d,pos={633,460},size={50,16},bodyWidth=40,title="min"

SetVariable setvar2d,limits={-inf,inf,1},value=bmin,proc=SetVarProc_scale

Slider slider2d,pos={623,490},size={244,19},variable =bmin

Slider slider2d,limits={0,255,1},value= 0,vert= 0,ticks= 0,proc=SliderProc_scale

SetVariable setvar3d,pos={633,510},size={50,16},bodyWidth=40,title="max"

SetVariable setvar3d,limits={-inf,inf,1},value=bmax,proc=SetVarProc_scale

Slider slider3d,pos={623,530},size={244,19},variable=bmax

Slider slider3d,limits={0,255,1},value= 255,vert= 0,ticks= 0,proc=SliderProc_scale

CheckBox check1,pos={769,577},size={95,14},proc=check_scale,title="Only intracellular"

CheckBox check1,value= 0

wave h_gc,h_rc,h_bc

Display/W=(0.7,0.23,0.99,0.33)/HOST=# h_gc

ModifyGraph margin=3,frameInset=1

ModifyGraph mode=5

ModifyGraph rgb(h_gc)=(13056,26112,0)

ModifyGraph hbFill=2

ModifyGraph noLabel=2

ModifyGraph axThick=0

SetAxis left 0,1000

SetAxis bottom 2,255

RenameWindow #,Gg

SetActiveSubwindow ##

Display/W=(0.7,0.55,0.99,0.65)/HOST=# h_rc

ModifyGraph margin=3,frameInset=1

ModifyGraph mode=5

ModifyGraph rgb(h_rc)=(65000,0,0)

ModifyGraph hbFill=2

ModifyGraph noLabel=2

ModifyGraph axThick=0

SetAxis left 0,1000

SetAxis bottom 2,255

RenameWindow #,Gr

SetActiveSubwindow ##

Display/W=(0.7,0.87,0.99,0.97)/HOST=# h_bc

ModifyGraph margin=3,frameInset=1

ModifyGraph mode=5

ModifyGraph rgb=(0,12800,52224)

ModifyGraph hbFill=2

ModifyGraph noLabel=2

ModifyGraph axThick=0

SetAxis left 0,1000

SetAxis bottom 2,255

RenameWindow #,Gb

SetActiveSubwindow ##

End

Function check_scale(ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

makeRGB(0)

End

```
Function ButtonProc_2editresults(ctrlName) : ButtonControl
```

```
String ctrlName
```

```
make/O/T/N=(2,11) resC=""
```

```
wave/T res=root:resC
```

```
res[0][0]="cell"
```

```
res[0][1]="image"
```

```
res[0][2]="pos{x,y}"
```

```
res[0][3]="threshold/filter"
```

```
res[0][4]="cell size (voxels)"
```

```
res[0][5]="background avg signal"
```

```
res[0][6]="background SD"
```

```
res[0][7]="cell avg signal"
```

```
res[0][8]="# voxels above threshold"
```

```
res[0][9]="DAPI size"
```

```
res[0][10]="DAPI excluded"
```

```
setdatafolder root:cells
```

```
string c1=replacestring(" ",stringbykey("FOLDERS",datafolderdir(1),":",";")) //data folders
```

```
variable num=itemsinlist(c1,";")
```

```
setdatafolder root:
```

```
//c1=sortlist(c1)
```

```
variable p1=0
```

```
for(p1=0;p1<num;p1+=1)
```

```
string f1=stringfromlist(p1,c1,";")
```

```
insertpoints (dimsize(res,0)), 1, res
```

```
res[p1+2][0]=f1
```

```
svar n1=root:cells:$(f1):file
```

```
res[p1+2][1]=n1
```

```

svar n2=root:cells:$(f1):marq

        res[p1+2][2]=n2

nvar v1=root:cells:$(f1):ibathr

nvar v2=root:cells:$(f1):ibafilter

        res[p1+2][3]=num2str(v1)+"/"+num2str(v2)


nvar v3=root:cells:$(f1):ibasize

        res[p1+2][4]=num2str(v3)


nvar v4a=root:cells:$(f1):v_extra

        res[p1+2][5]=num2str(v4a)    //signal background

nvar v4=root:cells:$(f1):v_thr

        res[p1+2][6]=num2str(v4)    //SD background

nvar v5=root:cells:$(f1):v_cell

        res[p1+2][7]=num2str(v5)    ///cell signal

nvar v6a=root:cells:$(f1):V_PoT

        res[p1+2][8]=num2str(v6a)    //voxels above thr

nvar v6=root:cells:$(f1):nuc_size

        res[p1+2][9]=num2str(v6)    //vGat attached

nvar v7=root:cells:$(f1):dapi_ex

        res[p1+2][10]=num2str(v7)    ///"vGat incorporated"

    endfor

    dowindow/K w_res

    edit/K=1/N=w_res res as " "

End

```

```

#pragma rtGlobals=3    // Use modern global access method and strict wave access.

```

```
Function ButtonProc_3A(ctrlName) : ButtonControl
```

```
    String ctrlName
```

```
    NVAR rmina
```

```
    NVAR v_thr
```

```
    NVAR v_extra
```

```
        rmina=round(v_extra+v_thr)
```

```
        makeRGBall(0)
```

```
End
```

```
function scaleITall(plane)
```

```
variable plane
```

```
    if(plane==0)//red channel==GABBR
```

```
    wave curw=:pl0all
```

```
    NVAR min1=root:rminA
```

```
    NVAR max1=root:rmaxA
```

```
    elseif(plane==1)//green channel==iba1
```

```
    wave curw=:pl1all
```

```
    NVAR min1=root:gminA
```

```
    NVAR max1=root:gmaxA
```

```
    elseif(plane==2)//blue channel==DAPI
```

```
    wave curw=:pl2all
```

```
    NVAR min1=root:bminA
```

```
    NVAR max1=root:bmaxA
```

```
    endif
```

```
    //print min1,max1,nameofwave(curw),plane
```

```
    redimension/S curw
```

```
    curw-=min1
```

```

    curw*=(curw[p][q]>=0)

    curw/=(max1-min1)

    curw=curw^(curw<1)

    curw*=255

    redimension/B/U curw

end

function sc1all()

wave cm1all

wave pl0all,pl1all,pl2all

    if(waveexists(pl0all))

        imagetransform/P=0/D=pl0all setplane cm1all

    endif

    if(waveexists(pl1all))

        imagetransform/P=1/D=pl1all setplane cm1all

    endif

    if(waveexists(pl2all))

        imagetransform/P=2/D=pl2all setplane cm1all

    endif

end

function makeRGBall(var)

variable var

wave gca=:allcell

wave rca=:all_vglutA

```

```
wave bca=:all_dapiA
```

```
if(var==0)

    duplicate/O rca, pl0all

    imagefilter/N=3 gauss pl0all;imagefilter/N=3 gauss pl0all;imagefilter/N=3 gauss pl0all

    scaleITall(0)

elseif    (var==1)

    duplicate/o gca,pl1all

    imagefilter/N=3 gauss pl1all;imagefilter/N=3 gauss pl1all;;imagefilter/N=3 gauss pl1all

    scaleITall(1)

elseif    (var==2)

    duplicate/O bca, pl2all

    imagefilter/N=3 gauss pl2all;          imagefilter/N=3 gauss pl2all;          imagefilter/N=3 gauss pl2all

    scaleITall(2)

else

    duplicate/O rca, pl0all

    duplicate/o gca,pl1all

    duplicate/O bca, pl2all

    scaleITall(0)

    scaleITall(1)

    scaleITall(2)

endif

sc1all()
```

```
end
```

```
Function SliderProc_scaleA(ctrlName,sliderValue,event) : SliderControl
```

```
String ctrlName
```

Variable sliderValue

Variable event // bit field: bit 0: value set, 1: mouse down, 2: mouse up, 3: mouse moved

if(event==4)

if((stringmatch(ctrlName,"*0"))||(stringmatch(ctrlName,"*1")))//Iba

 makeRGBall(1)

elseif((stringmatch(ctrlName,"*2"))||(stringmatch(ctrlName,"*3")))//GABBR

 makeRGBall(0)

elseif(stringmatch(ctrlName,"*d"))//DAPI

 makeRGBall(2)

endif

endif

return 0

End

Function SetVarProc_scaleA(ctrlName,varNum,varStr,varName) : SetVariableControl

String ctrlName

Variable varNum

String varStr

String varName

if(stringmatch(varname,"g*"))//Iba

 makeRGBall(1)

elseif(stringmatch(varname,"r*"))//GABBR

 makeRGBall(0)

elseif(stringmatch(varname,"b*"))//DAPI

 makeRGBall(2)

endif

End

```
#pragma rtGlobals=3           // Use modern global access method and strict wave access.
```

```
Function ButtonProc_2reanalyse(ctrlName) : ButtonControl
```

```
    String ctrlName
```

```
    controlinfo/W=panel0 check0
```

```
    variable en=v_value
```

```
    controlinfo/W=panel0 check2
```

```
    variable os=v_value
```

```
    variable p1
```

```
    string fl=getcells()
```

```
    for(p1=0;p1<itemsinlist(fl,";");p1+=1)//1;p1+=1)//
```

```
    //load cell
```

```
        PopMenuProc_loadcell("",p1,stringfromlist(p1,fl,";"))
```

```
    //analyse
```

```
        checkbox check2, win=panel0, value=(os)
```

```
        checkbox check0, win=panel0, value=(en)
```



```

        ButtonProc_restore("")

        ButtonProc_4lbamask("")

        ButtonProc_useit("dapi")

        CheckProc_1("",en)

        ButtonProc_restore("")

        checkscale("")

    //save

        ButtonProc_saveCell("AUTO_cell"+num2str(p1+1))

    endfor

//Result

    ButtonProc_2editresults("")

End

```

```

Function ButtonProc_saveCell(ctrlName) : ButtonControl

    String ctrlName

    if (!datafolderexists("Cells"))

        newdatafolder/O root:Cells

    endif

    setdatafolder root:cells

string nn

```

variable num

```
if(stringmatch(ctrlName,"auto*"))
```

```
    nn=stringfromlist(1,ctrlName,    "_")
```

```
    newdatafolder/O/S $(nn)
```

```
    num= str2num(ctrlName[strlen(ctrlName)-1])
```

```
else
```

```
string cl=replacestring(", ",stringbykey("FOLDERS",datafolderdir(1),":",";")) //data folders
```

```
    num=itemsinlist(cl,";")+1
```

```
    nn=("cell"+num2str(num))
```

```
    prompt nn,"name of cell"
```

```
    doprompt "Name of stored cell", nn
```

```
    newdatafolder/O/S $(nn)
```

```
endif
```

```
wave w_iba=root:w_iba
```

```
string/G file=note(w_iba)
```

```
wave cell=root:cell
```

```
string/G marq=note(cell)
```

```
wave gz=root:cellz
```

```
string/G crop
```

```
    if(waveexists(gz))
```

```
        crop=note(gz)
```

```
    endif
```

```
NVAR ithr=root:thr
```

```
variable/G ibathr=ithr
```

```
nvar ifil=root:smo
```

```
variable/G ibafilter=ifil
```

```
//results
```

```
nvar isize=root:res_size
```

```
variable/G ibasize=isize
```

```
NVAR mbas=root:v_extra
```

```
variable/G v_extra=mbas
```

```
nvar tbas=root:v_thr
```

```
variable/G v_thr=tbas
```

```
nvar mCell=root:v_cell
```

```
variable/G v_cell=mCell
```

```
nvar pot=root:v_PoT
```

```
variable/G V_PoT=pot
```

```
nvar nucS=root:nuc_size
```

```
variable/G nuc_size=nucs
```

```
nvar nucE=root:dapi_ex
```

```
variable/G dapi_ex=nucE
```

```
wave curw=root:mask44
```

```
wave curwdapi=root:mask44dapi
```

```
duplicate/O curw, mask44
```

```
duplicate/O curwdapi, mask44dapi
```

```
setdatafolder root:
```

```
popupmenu popup0, win=panel0,mode=(num)
```

End

```
function quantify(lasch):ButtonControl
```

```
string lasch
```

```
checkbox check0, win=panel0, value=0
```

```
killwaves/Z mask44dapi,temp44dapi,mask_dapi
```

```
getmarquee /W=panel0#G0 left,top
```

```
if(!V_flag)
```

```
    return 0
```

```
endif
```

```
wave M_zProjection=:allcell
```

```
//display
```

```
duplicate/O/R=[v_left,v_right][v_top,v_bottom] M_zProjection, cell
```

```
Note cell, num2str(v_left)+";"+num2str(v_right)+";"+num2str(v_top)+";"+num2str(v_bottom)
```

```
//calculate
```

```
wave w_iba
```

```
duplicate/O/R=[v_left,v_right][v_top,v_bottom] w_iba, cell1, temp44, mask44
```

```
redimension/B/U cell1, temp44
```

```
mask44=1
```

```
wave w_vglut
```

```
//left IMG
```

```
duplicate/O/R=[v_left,v_right][v_top,v_bottom] w_vglut, glu1
```

```
wave w_dapi
```

```
duplicate/O/R=[v_left,v_right][v_top,v_bottom] w_dapi, celldapi
```

```
redimension/B/U celldapi
```

```
genMask(cell1)
```

```
NVAR res_size
```

```
wave cell2
```

```
wavestats/Q cell2
```

```
res_size=v_sum
```

```
//newimage/K=1/S=0 cell
```

```
end
```

```
function loadIMG(lasch):ButtonControl
```

```
string lasch
```

```
variable fref
```

```
open/R/D/T=".tif" fref
```

```
string fp= S_fileName
```

```
string/G path1=fp
```

```
ImageLoad/T=tiff/S=0/C=100/LR3D/G/N=Img fp
```

```
NVAR ibaChan
```

```
wave img
```

```
controlinfo/W=panel0 popup0
```

```
ibaChan=v_value-1
```

```
variable col
```

```
//      prompt col, "0=red; 1=green;2=blue"
```

```
//      doprompt "select channel color", col
```

```
if(stringmatch(lasch,"*1")) ///iba1
```

```

//g2gray("img", col)

wave output

duplicate/O img, W_iba

note/K w_iba,fp

imagetransform zProjection w_iba

wave M_zprojection

duplicate/O M_zprojection, allcell

make/O/N=(dimsize(allcell,0),dimsize(allcell,1),3)/B/U cm1all

makeRGBall(1)

endif

if(stringmatch(lasch,"*2")) ///vGlut

//g2gray("img", col)

//wave output

duplicate/O img, W_vGlut

note/K W_vGlut,fp

imagetransform zProjection W_vGlut

wave M_zprojection

duplicate/O M_zprojection, all_vglutA

makeRGBall(0)

endif

```

```

        if(stringmatch(lasch,"*3")) ///vGat

//g2gray("img", col)

//wave output

duplicate/O img, W_DAPI

note/K W_DAPI,fp

imagertransform zProjection W_DAPI

wave M_zprojection

duplicate/O          M_zprojection, all_dapiA

makeRGBall(2)

endif

killwaves/Z img

//newimage/K=1 W_iba

end

Function ButtonProc_3(ctrlName) : ButtonControl

String ctrlName

NVAR rmin, rmax

NVAR v_thr

NVAR v_extra

rmin=round(v_extra+v_thr)

rmax=rmin+5

makeRGB(0)

```

End

Function CheckProc_1(ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

checkbox check0, win=panel0, value=checked

if(checked)

wave mask44dapi

if(!waveexists(mask44dapi))

ButtonProc_useit("Button_dapi")

else

endif

endif

ButtonProc_useit("")

End

Function ButtonProc_projDAPI(ctrlName) : ButtonControl

String ctrlName

controlinfo/W=panel0 check44

if(v_value)

buttonproc_restore("laschDAPI")

endif

fw_projDAPI()

End


```
Function PopMenuProc_loadcell(ctrlName,popNum,popStr) : PopupMenuControl
```

```
String ctrlName
```

```
Variable popNum
```

```
String popStr
```

```
popupmenu popup0,win=panel0, mode=popnum
```

```
checkbox check0, win=panel0, value=0
```

```
checkbox check2, win=panel0, value=0
```

```
SVAR marq=root:cells:$(popstr):marq
```

```
if(!SVAR_exists(marq))
```

```
return 0
```

```
endif
```

```
variable/G l1=str2num(stringfromlist(0,marq,";"))
```

```
variable/G r1=str2num(stringfromlist(1,marq,";"))
```

```
variable/G t1=str2num(stringfromlist(2,marq,";"))
```

```
variable/G b1=str2num(stringfromlist(3,marq,";"))
```

```
wave M_zProjection=:allcell //2D
```

```
//display
```

```
duplicate/O/R=[l1,r1][t1,b1] M_zProjection, cell
```

```
Note cell, num2str(l1)+";"+num2str(r1)+";"+num2str(t1)+";"+num2str(b1)
```

```
//calculate
```

```
wave w_iba //3D
```

```
duplicate/O/R=[l1,r1][t1,b1] w_iba, cell1,temp44
```

```
redimension/B/U cell1, temp44
```

```
wave curw=root:cells:$(popstr):mask44
```

```
duplicate/O curw, root:mask44
```

```
temp44*=curw
```

```

        cell1*=curw

        ButtonProc_restore("")

        duplicate/O/R=[l1,r1][t1,b1] w_iba, cell1

wave w_vglut

//left IMG

        duplicate/O/R=[l1,r1][t1,b1] w_vglut, glu1


wave w_dapi

        duplicate/O/R=[l1,r1][t1,b1] w_dapi, celldapi, all_dapi2m, tmp44,tmpOS

        tmp44=1

        tmpOS=1

        redimension/B/U celldapi, all_dapi2m

wave all_dapi2mp

wave curwD=root:cells:$(popstr):mask44dapi

        duplicate/O curwD, root:mask44dapi, root:mask_dapi

wave mask_dapi

        mask_dapi=(mask_dapi==0)


wave mask44dapi

if(waveexists(mask44dapi))

        all_dapi2m*=mask44dapi

endif

NVAR smo

variable p1

if(smo>0)

        for(p1=0;p1<smo;p1+=1)

                Imagefilter/N={3}/O gauss3d all_dapi2m

        endfor

endif

imagetransform zprojection all_dapi2m

```

```
duplicate/O M_zProjection, all_dapi2mp  
  
duplicate/O all_dapi2m,all_DAPI2  
  
buttonproc_restore("laschDAPI")  
  
duplicate/O/R=[l1,r1][t1,b1] w_dapi, all_dapi2  
  
redimension/B/U all_dapi2
```

```
genMask(temp44)
```

```
checkscale("")
```

End

Package 4: IHC-GABA concentration in astocytes

```
#pragma rtGlobals=3
```

```
// Use modern global access method and strict wave access.
```

```
function multipleanalysis(lasch)
```

```
string lasch
```

```
make/O/T/N=(2,12) MAres=""
```

```
wave/T res=root:MAres
```

```
res[0][0]="cell"
```

```
res[0][1]="image"
```

```
res[0][2]="pos[x,y]"
```

```
res[0][3]="threshold/filter"
```

```
res[0][4]="cell size (voxels)"
```

```
res[0][5]="background avg signal"
```

```
res[0][6]="background SD"
```

```
res[0][7]="cell avg signal"
```

```
res[0][8]="# voxels above threshold"
```

```
res[0][9]="DAPI size"
```

```
res[0][10]="DAPI excluded"
```

```
res[0][11]="GFAP signal"
```

```
newpath/O/Q MApath
```

```
if(V_flag)
```

```
return 0
```

```
endif
```

```
pathinfo MApath
```

```
dowindow/K w_MAres
```

```
edit/W=(5.25,43.25,1013.25,237.5)/K=1/N=w_MAres MAres as "All cells from "+stringfromlist(itemsinlist(s_path,":-1,s_path,":-1)
```

```
string fl=IndexedFile(MApath, -1, ".pyp" )// [, creatorStr ])
```

```
string obj="W_iba;w_dapi;w_vglut"
```

```
variable p1
```

```
for(p1=0;p1<itemsinlist(fl,"");p1+=1)
```

```
LoadData/O/Q/P=MApath/J=obj (stringfromlist(p1,fl,""))// [/D/I/J=objectNamesStr /L=loadFlags /O[=overwriteMode ]  
/P=pathName /Q/R/S=subDataFolderStr /T[=topLevelName ] ] fileOrFolderNameStr
```

```
LoadData/O/Q/P=MApath/S="Cells"/R/T (stringfromlist(p1,fl,""))
```

```
ButtonProc_2reanalyse("")
```

```
wave /T resC
```

```
variable rem=(dimsize(MAres,0))
```

```
variable n1=dimsize(resc,0)-2
```

```
insertpoints/M=0 (rem),(n1),MAres
```

```
MAres[rem,rem-1+n1][]=resc[p+2-rem][q]
```

```
endfor
```

```
end
```

```
function checkScale(lasch)
```

```
string lasch
```

```
wave cell2 //threshold 3D
```

```
wave glu2 ///gabbr signal 3D
```

```
wave dapi=:mask_dapi//dapi mask 3D
```

```

duplicate/O glu2, glu2o, glu2c//outside, cell

glu2o*=(cell2==0)

controlinfo/W=panel0 check0

if( (v_value)&&(waveexists(dapi)) )///dapi exclusion
    glu2o*=(dapi==1)//nuclear region is no background
endif

glu2c*=(cell2==1)

Controlinfo/W=panel0 check2

if(v_value)
    onllyomatic()//generates a rectangular mask
wave cell2=:tmpOS
wave tmp44
    glu2o*=(tmp44[p][q]==0)
    glu2c*=(tmp44[p][q]==0)
    corrCS()
else
wave cell2=:cell2
endif

//imagetransform/METH=1 zProjection glu2o
//wave M_zProjection
//    duplicate/O M_zProjection, co,co1
    duplicate/O glu2o,glu2or
wave co=glu2or
    redimension/N=(numpnts(co))/S co
    co/=(co[p]!=0)
    co/=(co[p]>=8)

```

```

//      wavestats/Q co

make/O h_c,h_o

Histogram/B={0,1,255} co,h_o

variable mo=wavemax(h_o)

h_o/=mo

wavestats/Q co

variable/G v_extra=v_avg

variable/G v_thr=2*v_sdev

//print "outside:",v_extra,v_thr


//      imagetransform/METH=1 zProjection glu2c

//      duplicate/O M_zProjection, cc,cc1

      duplicate/O glu2c,glu2cr

wave cc=glu2cr


      redimension/N=(numpnts(cc))/S cc

      cc/=(cc[p]!=0)

//      wavestats/Q cc

//      print "cell:",v_npnts,v_avg,v_sdev

      Histogram/B={0,1,255} cc,h_c

variable mc=wavemax(h_c)

h_c/=mc


k0 = 0;k1 = 1

CurveFit/Q/H="1100"/NTHR=0 gauss h_c /D

wave w_coef

variable/G v_cell=w_coef[2]

wavestats/Q/R=[round(v_extra+v_thr)+1,255] h_c

variable/G v_PoT=V_sum*mc

```

```

        setdrawlayer/K/W=panel0#g5 userfront

        SetDrawEnv/W=panel0#g5 xcoord= bottom,ycoord= prel,linefgc= (0,0,0), linethick=2

        drawline/W=panel0#g5 (V_extra+v_thr),0,(V_extra+v_thr),1

        SetDrawEnv/W=panel0#g5 xcoord= bottom,ycoord= prel,linefgc= (40000,0,0), linethick=2

        drawline/W=panel0#g5 (V_cell),0,(V_cell),1

        removefromgraph/Z/W=panel0#g5 fit_h_o,fit_h_c

```

```

end

```

```

function corrCS()

wave tmp44//rect mask

wave cell2 //cell mask

        duplicate/O cell2, tmp45

        tmp45*=(tmp44==0)

        wavestats/Q tmp45

NVAR res_size

        res_size=v_sum


        killwaves/Z tmp45

```

```

end

```

```

function onlysomatic()

wave mask_dapi

        if(!waveexists(mask_dapi))

                return 0

```



```

        endif

variable xyr=10

variable zr=2

        duplicate/O mask_dapi, tmp44

        if(mean(mask_dapi)==1)

wave cell2

                duplicate/O cell2 tmpOS

                return 0

        endif


wave tmp44

        tmp44*=255

        imageanalyzeparticles/A=500 stats tmp44


wave M_3DParticleInfo

variable x1=M_3DParticleInfo[0][0]-xyr

        if(x1<0)

                x1=0

        endif

variable x2=M_3DParticleInfo[0][1]+xyr

        if(x2>(dimsize(mask_dapi,0)-1))

                x2=dimsize(mask_dapi,0)-1

        endif

variable y1=M_3DParticleInfo[0][2]-xyr

        if(y1<0)

                y1=0

        endif

variable y2=M_3DParticleInfo[0][3]+xyr

        if(y2>dimsize(mask_dapi,1)-1)

                y2=dimsize(mask_dapi,1)-1

        endif

```

```

variable z1=M_3DParticleInfo[0][4]-zr
    if(z1<0)
        z1=0
    endif
variable z2=M_3DParticleInfo[0][5]+zr
    if(z2>dimsize(mask_dapi,2)-1)
        z2=dimsize(mask_dapi,2)-1
    endif

//print x1,x2,y1,y2,z1,z2

tmp44[x1,x2][y1,y2][z1,z2]=0
wave cell2

duplicate/O cell2 tmpOS
tmpOS*=(tmp44==0)
//newimage/S=0/K=1 tmpOS
killwaves/Z M_3DParticleInfo
end

```

```

function storedata(vl,vr,vt,vb)

variable vl,vr,vt,vb

string str=replacestring(", ",stringbykey("FOLDERS",DataFolderDir(1),":",";"))

string num=num2str(itemsinlist(str))

```

NVAR res_size

NVAR glut_fl

NVAR glut_sp

NVAR gaba_fl

NVAR gaba_sp

wave w_iba

```
print "file:\t", "cell size:\t", "glu spots:\t", "glu fluorescence:\t", "gab spots:\t", "gab fluorescence:\t"
```

```
print note(W_iba), "\t", vl, vr, vt, vb, "\t", res_size, "\t", glut_sp, "\t", glut_fl, "\t", gaba_sp, "\t", gaba_fl
```

end

function globals()

variable/G thr=30

variable/G smo=10

variable /G ibachan

variable/G res_size

variable /G imgF=3

variable/G xyr// xy range; odd number

variable/G zr//z range; odd number

variable /G rminA,rmaxA=255

variable /G gminA,gmaxA=255

variable /G bminA,bmaxA=255

String/G path1

variable/G somatic=0

make/O h_c=nan

make/O h_o=nan

make/O/N=(2,2)/B/U allcell=0

make/O/N=(2,2)/B/U cell=0

make/O/N=(2,2)/B/U cellsmod=0

make/O/N=(2,2)/B/U cell2d=0

make/O/N=(2,2)/B/U all_vglut=0

make/O/N=(2,2)/B/U s_vglut1=0

make/O/N=(2,2)/B/U s_vglut2=0

make/O/N=(2,2)/B/U all_dapi=0

make/O/N=(2,2)/B/U all_dapiP2=1

make/O/N=(2,2)/B/U all_dapi2MP=1

end

```

function g2gray(wn, var)

string wn

variable var //0=red,1=green, 2=blue

wave curl=${wn}

variable nl=dimsize(curl,3)

    if(!nl)

        duplicate/O curi, output

        return 0

    endif

    make/O/N=(dimsize(curl,0),dimsize(curl,1),dimsize(curl,3)) output
variable p1

    for(p1=0;p1<nl;p1+=1)

        imagetransform/CHIX=(p1) getChunk curl

wave M_Chunk

        imagetransform/P=(var) getplane M_Chunk

wave M_imageplane

        imagetransform/P=(p1)/D=M_imageplane setplane output

    endfor

    killwaves/Z m_chunk,M_imageplane

//newimage/K=1/S=0 output

end

```

Function CheckProc(ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

// print ctrlName

string n1

if(stringmatch (ctrlName,"glut*"))

n1="glut"

elseif(stringmatch (ctrlName,"gab*"))

n1="gab"

endif

string n2

variable m1,ms,c1,c2,c3

if(stringmatch (ctrlName,"*N"))

n2="n"

m1=8

ms=0.5

c1=34816

c2=34816

c3=34816

elseif(stringmatch (ctrlName,"*A"))

n2="o"

m1=19

ms=3

c1=0

c2=26112

c3=26112

elseif(stringmatch (ctrlName,"*I"))

```

n2="i"

m1=16

ms=3

c1=0

c2=52224

c3=52224

endif

if(checked)

AppendToGraph/L/T $(n1+"y"+n2) vs $(n1+"X"+n2)

ModifyGraph marker($(n1+"y"+n2))=m1, mode=3

ModifyGraph msize($(n1+"y"+n2))=ms

ModifyGraph rgb($(n1+"y"+n2))=(c1,c2,c3)

else

removefromGraph/Z $(n1+"Y"+n2)

endif

```

End

Function ButtonProc_proj(ctrlName) : ButtonControl

String ctrlName

controlinfo/W=panel0 check44

if(v_value)

buttonproc_restore("")

endif

fw_proj()

End

function fw_proj()

dowindow/K w_proj

PauseUpdate; Silent 1 // building window...

NewPanel /W=(671,133,1443,905)/K=1 /N=w_proj as "Projection"

SetDrawLayer UserBack

SetDrawEnv arrow= 1

DrawLine 385,370,385,20

SetDrawEnv arrow= 1

DrawLine 400,385,750,385

SetDrawEnv fsize= 18

DrawText 755,395,"x"

SetDrawEnv fsize= 18

DrawText 381,770,"z"

SetDrawEnv dash= 1,arrow= 1

DrawLine 370,385,20,385

SetDrawEnv dash= 1,arrow= 1

DrawLine 385,400,385,750

SetDrawEnv fsize= 18


```

DrawText 10,395,"z"

SetDrawEnv fsize= 18

DrawText 381,17,"y"

GroupBox group0,pos={20,400},size={350,350},labelBack={34816,34816,34816}

//CheckBox check0,pos={36,418},size={43,14},proc=CheckProc_1_setw,title="Crop "

//CheckBox check0,value= 1

Button button0,pos={100,450},size={150,40},proc=ButtonProc_crop,title="Delete"

Button button1,pos={100,500},size={150,40},proc=ButtonProc_crop,title="Crop"

Button button2,pos={100,600},size={150,40},proc=ButtonProc_restore,title="Restore"

Button button3,pos={100,650},size={150,40},proc=ButtonProc_useit,title="Use"

SetVariable setvar0,pos={121,417},size={108,16},bodyWidth=40,proc=SetVarProc_filter,title="Avg Img Filter"

SetVariable setvar0,limits={3,inf,2},value= imgF

```

wave cellZ, cellX, cellY

```

Display/W=(400,20,750,370)/HOST=#

AppendImage/T cellZ

ModifyImage cellZ ctab= {5,255,Grays,0}

ModifyGraph margin=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,Gxy

SetActiveSubwindow ##

```

```

Display/W=(20,20,370,370)/HOST=#

AppendImage/T cellX

ModifyImage cellX ctab= {5,255,Grays,0}

ModifyGraph margin=-1

```

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

SetAxis/A top

RenameWindow #,Gyz

SetActiveSubwindow ##

Display/W=(400,400,750,750)/HOST=#

AppendImage/T cellY

ModifyImage cellY ctab= {10,255,Grays,0}

ModifyGraph margin=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,Gxz

SetActiveSubwindow ##

setwindow W_proj, hook=hook_crop,hookevents=7

make/O/N=0 xw,yw

GraphWaveDraw /W=\$("W_proj")#\$("Gyz") /L/T/O yW, xW

End

```
function gwd(lasch1, lasch2)
```

```
string lasch1 //window
```

```
string lasch2 //subwindow
```

```
removefromgraph/W=${lasch1}#Gxy/Z yw
```

```
removefromgraph/W=${lasch1}#Gxz/Z yw
```

```
removefromgraph/W=${lasch1}#Gyz/Z yw
```

```
GraphWaveDraw /W=${lasch1}#${lasch2} /L/T/O yW, xW
```

```
wave xw
```

```
note/K xw, lasch2
```

```
end
```

```
Function hook_crop (infoStr)
```

```
String infoStr
```

```
String ev= StringByKey("event",infoStr)
```

```
String mo= StringByKey("modifiers",infoStr)
```

```
String Hw= StringByKey("HCSPEC",infoStr)
```

```
variable dec=0
```

```
if((str2num(mo)==5)&&(stringmatch(ev,"mousedown") ))//case Graphmormal
```

```
dec+=1
```

```
endif
```

```
if((str2num(mo)==4)&&(stringmatch(ev,"mouseup") ))//case graphwavedraw
```

```
dec+=1
```

```
endif
```

```
if(dec)
```

```

        if(stringmatch(hw,"**")) //projection win
            gwd(stringfromlist(0,hw,"#"),stringfromlist(1,hw,"#"))
        else
            graphnormal/W=$(hw)#gyz
        endif
    endif
End

```

End

Function CheckProc_1_setw(ctrlName,checked) : CheckBoxControl

```

    String ctrlName
    Variable checked

    if(checked)
        setwindow W_proj, hook=hook_crop,hookevents=7
        make/O/N=0 xw,yw
    else
        setwindow W_proj, hook=""
        killwaves/Z xw,yw
    endif
End

```

End

Function ButtonProc_crop(ctrlName) : ButtonControl

```

    String ctrlName

```

```

string rem="\t"

wave cellz

string o1=note(cellz)

variable it1=itemsinlist(o1,"\t")

    rem+="Action_"+num2str(it1)+"\r"

wave temp44

    if(!waveexists(temp44))

wave cell1

        duplicate/o cell1, temp44

    endif

wave xw,yw

string pr=note(xw)

    rem+=pr+"\r"        //1st row:window

    if(stringmatch(pr,"Gxy"))

wave curw=root:cellz

    endif

    if(stringmatch(pr,"Gxz"))

wave curw=root:cellY

    endif

    if(stringmatch(pr,"Gyz"))

wave curw=root:cellX

    endif

    ImageBoundaryToMask                                width=dimsize(curw,0),
height=dimsize(curw,1),scalingWave=curw,seedy=dimoffset(curw,1),seedx=dimoffset(curw,0), xwave=xw, ywave=yw

wave M_ROIMask

    if(stringmatch (ctrlName,"*1"))//crop

        M_Roimask=(M_Roimask[p][q]==0)

        rem+="crop_1\r"

```

```

else ///delete, as it is

    rem+="delete_2\r"

endif

if(stringmatch(pr,"Gxy"))

    temp44*=m_roimask[p][q]

endif

if(stringmatch(pr,"Gxz"))

    temp44[][]*=m_roimask[p][r]

endif

if(stringmatch(pr,"Gyz"))

    imagetransform fliprows m_roimask

    temp44[][]*=m_roimask[r][q]

endif

endif

imagetransform zprojection temp44

wave M_zprojection

imagetransform xprojection temp44

wave M_xprojection

imagetransform yprojection temp44

wave M_yprojection

duplicate/O M_zprojection, cellZ

duplicate/O M_xprojection, cellX

duplicate/O M_yprojection, cellY

ImageRotate/A=90/O cellx

killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh

rem+="xw:"+w2str(xw)+"\r"

rem+="yw:"+w2str(yw)+"\r"

```

```
note cellz, o1+rem
```

```
End
```

```
function rest_crop(crop)
```

```
string crop
```

```
if(!strlen(crop))
```

```
return 0
```

```
endif
```

```
variable p1
```

```
for(p1=1;p1<itemsinlist(crop,"\t");p1+=1)
```

```
string/G act=stringfromlist(p1,crop,"\t")
```

```
restore_xyw(act)
```

```
wave xw
```

```
note/K xw, stringfromlist(1,act,"\r")
```

```
ButtonProc_crop(stringfromlist(2,act,"\r"))
```

```
endfor
```

```
end
```

```
function/S w2str(w1)
```

```

wave w1

string expo=""

variable p1

    for(p1=0; p1<numpts(w1); p1+=1)

        expo+=num2str(w1[p1])+";"

    endfor

    return expo

end

```

```

Function SetVarProc_filter(ctrlName,varNum,varStr,varName) : SetVariableControl

```

```

    String ctrlName

```

```

    Variable varNum

```

```

    String varStr

```

```

    String varName

```

```

wave temp44

```

```

    ImageFilter/N=(varNum)/O gauss3d temp44

```

```

    imagetransform zprojection temp44

```

```

wave M_zprojection

```

```

    imagetransform xprojection temp44

```



```

wave M_xprojection

    imagetransform yprojection temp44
wave M_yprojection

    duplicate/O M_zprojection, cellZ

    duplicate/O M_xprojection, cellX

    duplicate/O M_yprojection, cellY

    ImageRotate/A=90/O cellx

    killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh

End

```

```

Function PopMenuProc_col(ctrlName,popNum,popStr) : PopupMenuControl

```

```

    String ctrlName

    Variable popNum

    String popStr

    if(str2num(ctrlName[5])==0)

```

```

NVAR g1

```

```

        g1=popnum

    endif

    if(str2num(ctrlName[5])==1)

```

```

NVAR i1

```

```

        i1=popnum

    endif

    if(str2num(ctrlName[5])==2)

```

```

NVAR ga1

```

```

        ga1=popnum

    endif

    if(str2num(ctrlName[5])==3)

```

```

NVAR gl1

```

```

        gl1=popnum
    endif

End

Function ButtonProc_rgbDoit(ctrlName) : ButtonControl

    String ctrlName

    doRGBproj()

End

function ButtonProc_expPICK(lasch):buttoncontrol

string lasch

dowindow/K w_projRGB0

display/K=1 /W=(335,75,1107,847)/N=w_projRGB0 as "RGB projection"

SetDrawLayer userfront

    SetDrawEnv arrow= 1

    SetDrawEnv xcoord=abs, ycoord=abs, save

    DrawLine 385,370,385,20

    SetDrawEnv arrow= 1

    DrawLine 400,385,750,385

    SetDrawEnv fsize= 18

    DrawText 755,395,"x"

    SetDrawEnv fsize= 18

    DrawText 381,770,"z"

    SetDrawEnv dash= 1,arrow= 1

    DrawLine 370,385,20,385

    SetDrawEnv dash= 1,arrow= 1

    DrawLine 385,400,385,750

```

SetDrawEnv fsize= 18

DrawText 10,395,"z"

SetDrawEnv fsize= 18

DrawText 381,17,"y"

wave cellZRGB, cellXRGB,cellYRGB

Display/W=(400,20,750,370)/HOST=#

AppendImage/T cellZRGB

ModifyImage cellZRGB ctab= {5,50,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

RenameWindow #,Gxy

SetActiveSubwindow ##

Display/W=(20,20,370,370)/HOST=#

AppendImage/T cellXRGB

ModifyImage cellXRGB ctab= {5,100,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

RenameWindow #,Gyz

SetActiveSubwindow ##

```

Display/W=(400,400,750,750)/HOST=#

AppendImage/T cellYRGB

ModifyImage cellYRGB ctab= {10,100,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,Gxz

SetActiveSubwindow ##

string rz=calcS()

//print rz

variable relx=str2num(stringfromlist(0,rz,";"))

variable rely=str2num(stringfromlist(1,rz,";"))

variable relz=str2num(stringfromlist(2,rz,";"))

//Gyz

ModifyGraph/W=w_projRGB0#Gyz axisEnab(left)={{(1-relx),1} //x

ModifyGraph/W=w_projRGB0#Gyz axisEnab(top)={{(1-relz),1} //z

//gxz

ModifyGraph/W=w_projRGB0#Gxz axisEnab(left)={{(1-relz),1} //z

ModifyGraph/W=w_projRGB0#Gxz axisEnab(top)={{(1-rely),1} //y

//Gxy

ModifyGraph/W=w_projRGB0#Gxy axisEnab(top)={{(1-rely),1} //y

ModifyGraph/W=w_projRGB0#Gxy axisEnab(left)={{(1-relx),1} //x

//      DolgorMenu "file", "Save Graphics"

```

end

function/S calcS()

//cut segment size

wave cell

string n1=note(cell)

variable vl1=str2num(stringfromlist(0,n1,";"))

variable vr1=str2num(stringfromlist(1,n1,";"))

variable vt1=str2num(stringfromlist(2,n1,";"))

variable vb1=str2num(stringfromlist(3,n1,";"))

variable vsy=abs(vr1-vl1)

variable vsx=abs(vb1-vt1)

//wholeimg

wave allcell

variable vwx=dimsize(allcell,0)

variable vwy=dimsize(allcell,1)

//stack:50 μm

//whole pic: 160X160

make/O/N=3 res

res[0]=(vsx/vwx)*160

res[1]=(vsy/vwy)*160

res[2]=50

variable wm=wavemax(res)

res/=wm

```
return num2str(res[0])+";"+num2str(res[1])+";"+num2str(res[2])+";"
```

```
end
```

```
function f_projRGB()
```

```
Dowindow/K w_projRGB
```

```
    rgbglob()
```

```
PauseUpdate; Silent 1          // building window...
```

```
newpanel/K=1 /W=(335,75,1107,847)/N=w_projRGB as "RGB projection"
```

```
SetDrawLayer UserBack
```

```
SetDrawEnv arrow= 1
```

```
DrawLine 385,370,385,20
```

```
SetDrawEnv arrow= 1
```

```
DrawLine 400,385,750,385
```

```
SetDrawEnv fsize= 18
```

```
DrawText 755,395,"x"
```

```
SetDrawEnv fsize= 18
```

```
DrawText 381,770,"z"
```

```
SetDrawEnv dash= 1,arrow= 1
```

```
DrawLine 370,385,20,385
```

```
SetDrawEnv dash= 1,arrow= 1
```

```
DrawLine 385,400,385,750
```

```
SetDrawEnv fsize= 18
```

```
DrawText 10,395,"z"
```

```
SetDrawEnv fsize= 18
```

```
DrawText 381,17,"y"
```

```
GroupBox group0,pos={20,400},size={350,350},labelBack={34816,34816,34816}
```

NVAR g1,gz1,gz2,gcon

```
GroupBox group1,pos={39,417},size={319,58},title="Background"

CheckBox check2,pos={124,413},size={24,14},title="F",value= 0

PopupMenu popup0,pos={49,442},size={60,21},bodyWidth=60, proc=PopupMenuProc_col

PopupMenu popup0,mode=(g1),popvalue="Gray",value= #""\ "Gray;Red;Green;Blue;Purple\ ""

SetVariable setvar6,pos={114,445},size={45,16},bodyWidth=30,title="z1"

SetVariable setvar6,limits={-inf,inf,0}, value=gz1

SetVariable setvar7,pos={174,445},size={45,16},bodyWidth=30,title="z2"

SetVariable setvar7,limits={-inf,inf,0}, value=gz2

SetVariable setvar0,pos={229,443},size={119,16},bodyWidth=60,title="contribution"

SetVariable setvar0,limits={-inf,inf,10}, value=gcon
```

NVAR i1,iz1,iz2,icon

```
GroupBox group2,pos={39,477},size={319,67},title="lba1"

PopupMenu popup1,pos={49,506},size={60,21},bodyWidth=60, proc=PopupMenuProc_col

PopupMenu popup1,mode=(i1),popvalue="Green",value= #""\ "Gray;Red;Green;Blue;Purple\ ""

SetVariable setvar8,pos={114,509},size={45,16},bodyWidth=30,title="z1"

SetVariable setvar8,limits={-inf,inf,0}, value=iz1

SetVariable setvar9,pos={174,509},size={45,16},bodyWidth=30,title="z2"

SetVariable setvar9,limits={-inf,inf,0}, value=iz2

SetVariable setvar1,pos={229,509},size={119,16},bodyWidth=60,title="contribution"

SetVariable setvar1,limits={-inf,inf,10}, value=icon
```

NVAR ga1,gaz1,gaz2,gacon

```
GroupBox group3,pos={39,549},size={319,68},title="Vgat"

PopupMenu popup2,pos={49,577},size={60,21},bodyWidth=60, proc=PopupMenuProc_col

PopupMenu popup2,mode=(ga1),popvalue="Blue",value= #""\ "Gray;Red;Green;Blue;Purple\ ""

SetVariable setvar4,pos={114,580},size={45,16},bodyWidth=30,title="z1"

SetVariable setvar4,limits={-inf,inf,0}, value=gaz1
```

```

SetVariable setvar5,pos={174,580},size={45,16},bodyWidth=30,title="z2"

SetVariable setvar5,limits={-inf,inf,0}, value=gaz2

SetVariable setvar2,pos={229,573},size={119,16},bodyWidth=60,title="contribution"

SetVariable setvar2,limits={-inf,inf,10},value= gacon

CheckBox check0,pos={289,595},size={43,14},title="ideal ",value= 0,proc=CheckProc_1ideal

```

NVAR gl1,glz1,glz2,glcon

```

GroupBox group4,pos={39,622},size={319,67},title="Vglut"

PopupMenu popup3,pos={49,652},size={60,21},bodyWidth=60, proc=PopMenuProc_col

PopupMenu popup3,mode={gl1},popvalue="Red",value= #"\Gray;Red;Green;Blue;Purple\""

SetVariable setvar05,pos={114,655},size={45,16},bodyWidth=30,title="z1"

SetVariable setvar05,limits={-inf,inf,0}, value=glz1

SetVariable setvar06,pos={174,655},size={45,16},bodyWidth=30,title="z2"

SetVariable setvar06,limits={-inf,inf,0}, value=glz2

SetVariable setvar3,pos={229,645},size={119,16},bodyWidth=60,title="contribution"

SetVariable setvar3,limits={-inf,inf,10},value= glcon

CheckBox check1,pos={289,668},size={43,14},title="ideal ",value= 0,proc=CheckProc_1ideal

```

```

Button button0,pos={295,700},size={60,40},proc=ButtonProc_rgbDoit,title="Doit"

Button button1,pos={40,700},size={60,40},proc=ButtonProc_expPict,title="Export"

```

```
doRGBproj()
```

wave cellZRGB, cellXRGB,cellYRGB

```
Display/W=(400,20,750,370)/HOST=#
```

```
AppendImage/T cellZRGB
```


ModifyImage cellZRGB ctab= {5,50,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

RenameWindow #,Gxy

SetActiveSubwindow ##

Display/W=(20,20,370,370)/HOST=#

AppendImage/T cellxRGB

ModifyImage cellXRGB ctab= {5,100,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

RenameWindow #,Gyz

SetActiveSubwindow ##

Display/W=(400,400,750,750)/HOST=#

AppendImage/T cellyRGB

ModifyImage cellYRGB ctab= {10,100,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

```
ModifyGraph noLabel=2  
ModifyGraph standoff=0  
ModifyGraph axThick=0  
SetAxis/A/R left  
RenameWindow #,Gxz  
SetActiveSubwindow ##  
End
```

```
function rgbglob()
```

```
variable/G g1=1 //gray
```

```
variable/G gz1=0
```

```
variable/G gz2=255
```

```
variable/G gcon=100
```

```
variable/G i1=3//green
```

```
variable/G iz1=0
```

```
variable/G iz2=255
```

```
variable/G icon=100
```

```
variable/G ga1=4//blue
```

```
variable/G gaz1=0
```

```
variable/G gaz2=255
```

```
variable/G gacon=100
```

```
variable/G gl1=2//red
```

```
variable/G glz1=0
```

```
variable/G glz2=255
```

```
variable/G glcon=100
```

```
end
```

```
function tempproj(w1,var1)
```

```
wave w1
```

```
variable var1 //threshold
```

```
    w1*=(w1[p][q][r]>=var1)
```

```
    imagetransform xprojection w1
```

```
wave M_xprojection
```

```
    imagetransform zprojection w1
```

```
wave M_zprojection
```

```
    imagetransform yprojection w1
```

```
wave M_yprojection
```

```
    duplicate/O M_xprojection, tempX
```

```
    duplicate/O M_yprojection, tempY
```

```
    duplicate/O M_zprojection, tempZ
```

```
    ImageRotate/A=90/O tempX
```

```
variable xmax=wavemax(tempX)/255
```

```
    tempX/=(xmax)
```

```
variable ymax=wavemax(tempY)/255
```

```
    tempY/=(ymax)
```

```
variable zmax=wavemax(tempZ)/255
```

```
    tempZ/=(zmax)
```

```
    //print wavemax(tempX),wavemax(tempY),wavemax(tempZ)
```

```
    killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh
```

end

function doRGBproj()

wave cell1 //iba cell stack

wave cellsmo //filtered iba cell stack

Make/O/B/U/N=(dimsize(cell1,0),dimsize(cell1,1),3) cellZRGB=0

Make/O/B/U/N=(dimsize(cell1,2),dimsize(cell1,1),3) cellXRGB=0

Make/O/B/U/N=(dimsize(cell1,0),dimsize(cell1,2),3) cellYRGB=0

wave cellZRGB, cellXRGB,cellYRGB

///background, gray, lba1 channel

NVAR g1

NVAR gz1

NVAR gz2

NVAR gcon

controlinfo/W=w_projRGB check2

if(!v_value)

duplicate/O cell1, temp444///cell1== lba stack

else

duplicate/O cellsmo, temp444///cellsmo==filtered lba stack

```

endif

tempproj(temp444,0)

wave tempx,tempy,tempz

scaletemp(gz1,gz2)

variable s1=gcon

string col1=getcol(g1)

variable p1

for(p1=0;p1<itemsinlist(col1);p1+=1)

    cellxRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempx[p][q]/(255/s1)

    cellyRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempy[p][q]/(255/s1)

    cellzRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempz[p][q]/(255/s1)

endfor

///Iba thresh, green

wave cell2 ///cropped iba image

NVAR i1

NVAR iz1

NVAR iz2

NVAR icon

duplicate/O cell2, temp445

tempproj(temp445,0)

scaletemp(iz1,iz2)

s1=icon

col1=getcol(i1)

for(p1=0;p1<itemsinlist(col1);p1+=1)

    cellxRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempx[p][q]/(255/s1)

    cellyRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempy[p][q]/(255/s1)

    cellzRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempz[p][q]/(255/s1)

```

```
endfor
```

```
killwaves/Z temp444,temp445,temp446,temp447
```

```
//vglut/vgat thresh, red
```

```
NVAR gl1
```

```
NVAR glz1
```

```
NVAR glz2
```

```
NVAR glcon
```

```
NVAR thrGL
```

```
wave gluRGB=root:glu1 //vglut
```

```
duplicate/O gluRGB, temp447
```

```
tempproj(temp447,thrGL)
```

```
scaletemp(glz1,glz2)
```

```
s1=glcon
```

```
col1=getcol(gl1)
```

```
for(p1=0;p1<itemsinlist(col1);p1+=1)
```

```
    cellxRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempz[p][q]/(255/s1)
```

```
    cellyRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempz[p][q]/(255/s1)
```

```
    cellzRGB[][][str2num(stringfromlist(p1,col1,";"))]+=tempz[p][q]/(255/s1)
```

```
endfor
```

```
wave gabRGB=root:gab1 //vgat
```

```
NVAR ga1
```

```
NVAR gaz1
```

```
NVAR gaz2
```

```
NVAR gacon
```

```
NVAR thrGA
```

```
duplicate/O gabRGB, temp446
```

```
tempproj(temp446,thrGA)
```

```
scaletemp(gaz1,gaz2)
```

```
s1=gacon
```

```
col1=getcol(ga1)
```

```
for(p1=0;p1<itemsinlist(col1);p1+=1)
```

```
    cellxRGB[][][str2num(stringfromlist(p1,col1,";"))]+=temp[p][q]/(255/s1)
```

```
    cellyRGB[][][str2num(stringfromlist(p1,col1,";"))]+=temp[p][q]/(255/s1)
```

```
    cellzRGB[][][str2num(stringfromlist(p1,col1,";"))]+=temp[p][q]/(255/s1)
```

```
endfor
```

```
setscale/P x (dimoffset(cell1,0)),1,cellzRGB
```

```
setscale/P y (dimoffset(cell1,1)),1,cellzRGB
```

```
setscale/P y (dimoffset(cell1,1)),1,cellxRGB
```

```
setscale/P x (dimoffset(cell1,0)),1,cellyRGB
```

```
end
```

```
function/S getcol(var)
```

```
variable var
```

```
if(var==1)//gray
```

```
    return "0;1;2"
```

```

endif

if(var==2)//red
    return "0"
endif

if(var==3)
    return "1"
endif

if(var==4)
    return "2"
endif

if(var==5)
    return "0;2"
endif
end

```

end

```
function scaletemp(gz1,gz2)
```

```
variable gz1,gz2
```

```
wave tempx,tempy,tempz
```

```
redimension/S tempx
```

```
redimension/S tempy
```

```
redimension/S tempz
```

```
tempx-=gz1
```

```
tempx*=(tempx[p][q]>=0)
```

```
tempx*=255/(gz2)
```

```
tempx/=(tempx[p][q]<=255)
```

```
tempy-=gz1
```



```

    tempy*=(tempy[p][q]>=0)

    tempy*=255/gz2

    tempy/=(tempy[p][q]<=255)


    tempz-=gz1

    tempz*=(tempz[p][q]>=0)

    tempz*=255/gz2

    tempz/=(tempz[p][q]<=255)

    redimension/B/U tempx

    redimension/B/U tempy

    redimension/B/U tempz

end

```

```

#pragma rtGlobals=3           // Use modern global access method and strict wave access.

```

```

function autoana(lasch)

string lasch

```

```

getmarquee /W=panel0#G0 left,top

if(!V_flag)

    return 0

endif

quantify("")

ButtonProc_4lbamask("")

ButtonProc_useit("werDAPI")

Checkscale("")

setdatafolder root:cells

string cl=replacestring(", ",stringbykey("FOLDERS",datafolderdir(1),":",";")) //data folders

variable  num=itemsinlist(cl,",")+1

string    nn=("cell"+num2str(num))

setdatafolder root:

ButtonProc_saveCell("auto_"+nn)

CheckProc_2chowAna("",1)

end

```

```

menu "analysis"

"_"

"Quantify IHC signals", f_pruning()

end

```

```

function f_Pruning ()

dowindow/K Panel0

    globals()

    PauseUpdate; Silent 1          // building window...

    NewPanel/K=1 /N=Panel0/W=(133,109,1630,894) as "Pruning..."

    modifypanel/W=Panel0 fixedsize=1


    GroupBox group3,pos={21,0},size={747,75},title=" "

    GroupBox group3,labelBack=(39168,39168,39168)


    Button button1,pos={34,22},size={50,20},proc=loadIMG,title="Iba1"

    Button button1,fColor=(6400,13056,0)

    Button button2,pos={100,22},size={50,20},proc=loadIMG,title="GABBR"

    Button button2,fColor=(62976,0,5888)

    Button button03,pos={34,49},size={50,20},proc=loadIMG,title="DAPI"

    Button button03,fColor=(0,12800,52224)


SVAR p_1=:path1

    SetVariable setvar15,pos={160,36},size={570,16},bodyWidth=544,title="Path"

    SetVariable setvar15,value= p_1,noedit= 1


    GroupBox group4,pos={1330,0},size={153,75},title=" "

    GroupBox group4,labelBack=(39168,39168,39168)

    Button button10,pos={1097,29},size={60,30},proc=ButtonProc_2editresults,title="Exp Data"


    GroupBox group5,pos={780,0},size={539,75},title=" "

    GroupBox group5,labelBack=(39168,39168,39168)

```

Button button9,pos={946,26},size={60,30},proc=ButtonProc_saveCell,title="Save Cell"

Button button11,pos={1035,26},size={60,30},proc=ButtonProc_2reanalyse,title="Reanalyse"

PopupMenu popup0,pos={789,31},size={128,21},bodyWidth=80,title="Open cell"

PopupMenu popup0,mode=1,value= getcells()

PopupMenu popup0 proc=PopMenuProc_loadcell

CheckBox check1,pos={1142,38},size={113,14},proc=CheckProc_2chowAna,title="show Analysed cells"

CheckBox check1,value=0

Button button09,pos={1413,26},size={60,30},proc=multipleanalysis,title="Multi Ana"

Button button8,pos={1343,26},size={60,30},proc=ButtonProc_2editresults,title="Data Table"

GroupBox group0,pos={780,85},size={705,240},title="Iba Mask"

GroupBox group0,labelBack={39168,39168,39168}

Button button0,pos={798,293},size={50,20},proc=autoana1,title="AutoANA"

Button button02,pos={868,293},size={50,20},proc=quantify,title="Copy cell"

variable /G smo=5

SetVariable setvar0,pos={943,297},size={78,16},bodyWidth=40,title="smooth"

SetVariable setvar0,value= smo

variable /G thr=30

SetVariable setvar1,pos={1034,297},size={87,16},bodyWidth=40,title="threshold"

SetVariable setvar1,value= thr, limits={0,inf,5}

CheckBox check_iba,pos={1137,299},size={57,14},title="dynamic",value= 0,proc=CheckProc_1_setdyn

SetVariable setvar14,pos={1328,289},size={121,16},bodyWidth=80,title="cell size"

SetVariable setvar14,format="%g voxels"

SetVariable setvar14,limits={-inf,inf,0},value= res_size,noedit= 1

Button button6,pos={1344,155},size={50,50},proc=ButtonProc_proj,title="Proj"

Button button7,pos={1354,220},size={100,50},proc=ButtonProc_useit,title="Mask"

CheckBox check44,pos={1347,131},size={43,14},title="New ",value= 1

GroupBox group1,pos={780,330},size={705,225},title="GABBR"

GroupBox group1,labelBack=(39168,39168,39168)

Button button3,pos={1375,487},size={50,50},proc=ButtonProc_2,title="Overlay"

Button button4,pos={1351,430},size={100,50},proc=checkScale,title="Check"

CheckBox check2,pos={1357,399},size={76,14},title="only somatic",value= 1

Button button5,pos={1410,155},size={50,50},proc=ButtonProc_projDAPI

Button button5 title="Crop\rNucleus"

CheckBox check0,pos={1356,374},size={97,14},proc=CheckProc_1,title="exclude Nucleus"

CheckBox check0,value= 1

NVAR gminA,gmaxA

GroupBox group6,pos={780,560},size={344,63},title="Iba1"

GroupBox group6,labelBack=(47872,47872,47872)

SetVariable setvar2,pos={788,578},size={60,16},bodyWidth=40,proc=SetVarProc_scaleA,title="min"

SetVariable setvar2,value= gminA

SetVariable setvar3,pos={785,599},size={63,16},bodyWidth=40,proc=SetVarProc_scaleA,title="max"

SetVariable setvar3,value= gmaxA

Slider slider0,pos={860,578},size={254,19},proc=SliderProc_scaleA

Slider slider0,limits={0,255,1},variable= gminA,vert= 0,ticks= 0

Slider slider1,pos={860,598},size={254,19},proc=SliderProc_scaleA

Slider slider1,limits={0,255,1},variable= gmaxA,vert= 0,ticks= 0

NVAR rminA,rmaxA

GroupBox group7,pos={780,628},size={344,80},title="GABBR"

GroupBox group7,labelBack=(47872,47872,47872)

SetVariable setvar4,pos={788,644},size={60,16},bodyWidth=40,proc=SetVarProc_scaleA,title="min"

```
SetVariable setvar4,value= rminA
```

```
SetVariable setvar5,pos={785,666},size={63,16},bodyWidth=40,proc=SetVarProc_scaleA,title="max"
```

```
SetVariable setvar5,value= rmaxA
```

```
Slider slider2,pos={860,645},size={254,19},proc=SliderProc_scaleA
```

```
Slider slider2,limits={0,255,1},variable= rminA,vert= 0,ticks= 0
```

```
Slider slider3,pos={860,665},size={254,19},proc=SliderProc_scaleA
```

```
Slider slider3,limits={0,255,1},variable= rmaxA,vert= 0,ticks= 0
```

```
Button button01,pos={909,684},size={100,20},proc=ButtonProc_3A,title="Threshold"
```

```
NVAR bminA,bmaxA
```

```
GroupBox group8,pos={780,710},size={344,63},title="DAPI"
```

```
GroupBox group8,labelBack=(47872,47872,47872)
```

```
SetVariable setvar2d,pos={789,731},size={60,16},bodyWidth=40,proc=SetVarProc_scaleA,title="min"
```

```
SetVariable setvar2d,value= bminA
```

```
SetVariable setvar3d,pos={786,751},size={63,16},bodyWidth=40,proc=SetVarProc_scaleA,title="max"
```

```
SetVariable setvar3d,value= bmaxA
```

```
Slider slider2d,pos={860,730},size={254,19},proc=SliderProc_scaleA
```

```
Slider slider2d,limits={0,255,1},variable= bminA,vert= 0,ticks= 0
```

```
Slider slider3d,pos={860,750},size={254,19},proc=SliderProc_scaleA
```

```
Slider slider3d,limits={0,255,1},variable= bmaxA,vert= 0,ticks= 0
```

```
wave allcell,cm1all
```

```
if(!waveexists(allcell))
```

```
make/O/N=(2,2) allcell
```

```
endif
```

```
if(!waveexists(cm1all))  
    make/O/N=(2,2,3) cm1all  
    redimension/B/U cm1all  
endif  
  
Display/W=(17,86,768,785)/HOST=#  
  
AppendImage/T cm1all
```

```
  
ModifyGraph margin=-1  
ModifyGraph mirror=0  
ModifyGraph nticks=0  
ModifyGraph noLabel=2  
ModifyGraph standoff=0  
ModifyGraph axThick=0  
  
SetAxis/A/R left  
  
RenameWindow #,G0  
  
SetActiveSubwindow ##
```

wave cell

```
if(!waveexists(cell))  
    make/O/N=(2,2) cell  
endif  
  
Display/W=(795,125,945,275)/HOST=# ///mask, 1st  
  
AppendImage/T cell  
  
ModifyImage cell ctab= {*,*,Green,0}  
  
ModifyGraph margin=-1  
ModifyGraph mirror=0  
ModifyGraph nticks=0  
ModifyGraph noLabel=2  
ModifyGraph standoff=0  
ModifyGraph axThick=0  
  
SetAxis/A/R left  
  
RenameWindow #,G1
```

```
SetActiveSubwindow ##
```

```
wave cellsmod
```

```
if(!waveexists(cellsmod))
```

```
    make/O/N=(2,2) cellsmod
```

```
wave cellsmod
```

```
endif
```

```
Display/W=(975,125,1125,275)/HOST=# //mask,2nd
```

```
AppendImage/T cellsmod
```

```
ModifyImage cellsmod ctab= {10,200,Green,0}
```

```
ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1
```

```
ModifyGraph mirror=0
```

```
ModifyGraph nticks=0
```

```
ModifyGraph noLabel=2
```

```
ModifyGraph standoff=0
```

```
ModifyGraph axThick=0
```

```
SetAxis/A/R left
```

```
RenameWindow #,G3
```

```
SetActiveSubwindow ##
```

```
wave cell2d
```

```
if(!waveexists(cell2d))
```

```
    make/O/N=(2,2) cell2d
```

```
wave cell2d
```

```
endif
```

```
Display/W=(1155,125,1305,275)/HOST=# //mask,3rd
```

```
AppendImage/T cell2d
```

```
ModifyImage cell2d ctab= {*,*,Grays,0}
```

```
ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1
```

```
ModifyGraph mirror=0
```

```
ModifyGraph nticks=0
```



```

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,G2

SetActiveSubwindow ##

///

wave all_vglut

if(!waveexists(all_vglut))

    make/O/N=(2,2) all_vglut

endif

Display/W=(795,360,945,510)/HOST=#

AppendImage/T all_vglut

ModifyImage all_vglut ctab= {0,*,Red,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,G4

SetActiveSubwindow ##

wave s_vglut1,s_vglut2

if(!waveexists(s_vglut1))

    make/O/N=(2,2) s_vglut1

endif

Display/W=(975,360,1125,510)/HOST=#

AppendImage/T S_vglut1

```

ModifyImage S_vglut1 ctab= {10,100,Red,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,G6

SetActiveSubwindow ##

Display/W=(1155,360,1305,510)/HOST=#

AppendImage/T S_vglut2

ModifyImage S_vglut2 ctab= {10,100,Red,0}

ModifyImage S_vglut2 maxRGB=NaN

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,G9

SetActiveSubwindow ##

wave h_o,h_c

Display/W=(1136,567,1483,778)/HOST=# h_o,h_c

ModifyGraph wbRGB=(43520,43520,43520)

ModifyGraph mode(h_o)=5

ModifyGraph lSize(h_c)=2

ModifyGraph rgb(h_o)=(34816,34816,34816),rgb(h_c)=(49152,1280,256)

ModifyGraph hbFill=2

```

ModifyGraph nticks(left)=0

ModifyGraph noLabel(left)=2

SetAxis bottom 0,70

Legend/C/N=text0/J "\\s(h_o) GABBR1 extracell.\r\\s(h_c) GABBR1 intracell."

SetDrawLayer UserFront

SetDrawEnv xcoord= bottom,ycoord= prel,linethick= 2

DrawLine 25.6613517618439,0,25.6613517618439,1

SetDrawEnv xcoord= bottom,ycoord= prel,linethick= 2,linefgc= (40000,0,0)

DrawLine 23.7168012957503,0,23.7168012957503,1

RenameWindow #,G5

SetActiveSubwindow ##

```

End

```
Function ButtonProc(ctrlName) : ButtonControl
```

```
String ctrlName
```

```
f_projRGB()
```

End

//////////

```
function analyse(lasch):ButtonControl
```

```
string lasch
```

```
//      quantify("")
```

```
getmarquee /W=panel0#G0 left,top
```

```
variable vl1
```

```
variable vt1
```

```
variable vr1
```

```
variable vb1
```

```
if(!V_flag)//no marquee
```

```
wave cell
```

```
string n1=note(cell)
```

```
    vl1=str2num(stringfromlist(0,n1,";"))
```

```
    vt1=str2num(stringfromlist(2,n1,";"))
```

```
    vr1=str2num(stringfromlist(1,n1,";"))
```

```
    vb1=str2num(stringfromlist(3,n1,";"))
```

```
else
```

```
    vl1=v_left
```

```
    vt1=v_top
```

```
    vr1=v_right
```

```
    vb1=v_bottom
```

```
endif
```

```
if(stringmatch(lasch,"*4"))//vglut
```

```
        qSpots(vl1,vt1,vr1,vb1,"glu")
    else
        qSpots(vl1,vt1,vr1,vb1,"gab")
    endif
end
```

```
function addspots(s_win,gg)
```

```
string s_win
```

```
string gg
```

```
    if(stringmatch(gg,"glu"))
```

```
        wave Ox=:glutXo
```

```
        wave Oy=:glutYo
```

```
        wave Oz=:glutZo
```

```
        wave Oz1=:glutZo1
```

```
        wave lx=:glutXi
```

```
        wave ly=:glutYi
```

```
        wave lz=:glutZi
```

```
        wave lz1=:glutZi1
```

```
        wave nx=:glutxn
```

```
        wave ny=:glutYn
```

```
        wave nz=:glutZn
```

```
        wave nz1=:glutZn1
```

```
    elseif(stringmatch(gg,"gab"))
```

```
        wave Ox=:gabXo
```

```
        wave Oy=:gabYo
```

```
        wave Oz=:gabZo
```

```
wave Oz1=:gabZo1
```

```
wave lx=:gabXi
```

```
wave ly=:gabYi
```

```
wave lz=:gabZi
```

```
wave lz1=:gabZi1
```

```
wave nx=:gabxn
```

```
wave ny=:gabYn
```

```
wave nz=:gabZn
```

```
wave nz1=:gabZn1
```

```
endif
```

```
variable r1=0
```

```
variable g1=26112
```

```
variable b1=65535
```

```
variable r2=8960
```

```
variable g2=39680
```

```
variable b2=51456
```

```
appendtograph/W=$(s_win)#Gyz/L/T Oy vs Oz1
```

```
appendtograph/W=$(s_win)#Gyz/L/T ly vs lz1
```

```
appendtograph/W=$(s_win)#Gyz/L/T ny vs nz1
```

```
ModifyGraph/W=$(s_win)#Gyz rgb($(nameofwave(iy)))=(r1,g1,b1)
```

```
ModifyGraph/W=$(s_win)#Gyz rgb($(nameofwave(oy)))=(r2,g2,b2)
```

```
ModifyGraph/W=$(s_win)#Gyz mode=3,marker=19
```

```
ModifyGraph/W=$(s_win)#Gyz rgb($(nameofwave(ny)))=(56576,56576,56576)
```

```
ModifyGraph/W=$(s_win)#Gyz marker($(nameofwave(ny)))=8,msize($(nameofwave(ny)))=0.5
```

```
appendtograph/W=$(s_win)#Gxy/L/T Oy vs Ox
```

```

appendtograph/W=$(s_win)#Gxy/L/T ly vs lx

appendtograph/W=$(s_win)#Gxy/L/T ny vs nx

ModifyGraph/W=$(s_win)#Gxy rgb($(nameofwave(iy)))=(r1,g1,b1)

ModifyGraph/W=$(s_win)#Gxy rgb($(nameofwave(oy)))=(r2,g2,b2)

ModifyGraph/W=$(s_win)#Gxy mode=3,marker=19

ModifyGraph/W=$(s_win)#Gxy rgb($(nameofwave(ny)))=(56576,56576,56576)

ModifyGraph/W=$(s_win)#Gxy marker($(nameofwave(ny)))=8,msize($(nameofwave(ny)))=0.5

```

```

appendtograph/W=$(s_win)#Gxz/L/T Oz vs Ox

appendtograph/W=$(s_win)#Gxz/L/T iz vs ix

appendtograph/W=$(s_win)#Gxz/L/T nz vs nx

ModifyGraph/W=$(s_win)#Gxz rgb($(nameofwave(iz)))=(r1,g1,b1)

ModifyGraph/W=$(s_win)#Gxz rgb($(nameofwave(oz)))=(r2,g2,b2)

ModifyGraph/W=$(s_win)#Gxz mode=3,marker=19

ModifyGraph/W=$(s_win)#Gxz rgb($(nameofwave(nz)))=(56576,56576,56576)

ModifyGraph/W=$(s_win)#Gxz marker($(nameofwave(nz)))=8,msize($(nameofwave(nz)))=0.5

```

end

```

//display results

sp=numpts$(spX)

redimension/D w_3dsm

w_3dsm/=(w_3dsm[p][q][r]>0)

wavestats/Q w_3dsm

```

```

    fl=V_avg

end

//NVAR smoga1

    //smooth (smoga1), gab1

wave w_vGat

    duplicate/O/R=[vl1,vr1][vt1,vb1] w_vGat, gab1

    redimension/B/U gab1

    gab1*=cell2

    imagetransform zprojection gab1

wave M_zProjection

    duplicate/O M_zProjection, S_vgat1

//GABA

NVAR gab_min,gab_max

NVAR thrGA

    imagethreshold/Q/M=(m1)/T=(thrGa)/I s_vgat1

wave M_imagethresh

    duplicate/O M_imagethresh, gabThR

    imageanalyzeParticles/A=(gab_min)/MAXA=(gab_max)/Q stats M_imagethresh

wave w_spotx,w_spoty

    w_spotx+=dimoffset(s_vgat1,0)

    w_spoty+=dimoffset(s_vgat1,1)

    duplicate/O w_spotx, gabx

    duplicate/O w_spoty, gaby

NVAR gaba_fl

NVAR gaba_sp

gaba_sp=numpts(gabX)

redimension/D gab1

```



```

gab1/=(gab1[p][q][r]>0)

wavestats/Q gab1

gaba_fl=V_avg

storeDATA(vl1,vr1,vt1,vb1)

killwaves/Z M_imagethresh,M_zProjection

end

```

```

Function ButtonProc_1b(ctrlName) : ButtonControl

    String ctrlName

    wave w_iba=:glu1

    nvar thr

    make/O/N=(dimesize(w_iba,2)) dx1,dy1

    variable p1

    for(p1=0; p1<(dimesize(w_iba,2)); p1+=1)

        imagetransform/P=(p1) getplane w_iba

    wave M_ImagePlane

    // imagethreshold/M=0/T=(thr)/O M_ImagePlane

    wavestats/Q M_ImagePlane

    dx1[p1]=p1

    dy1[p1]=v_sum

    endfor

    dowindow/K w_distr

    display /K=1/N=w_distr dy1 vs dx1

    // edit dx1, dy1

End

```

```

function restore_xyw(str1)

string str1

string x1=stringbykey("xw", str1, ":", "\r")

string y1=stringbykey("yw", str1, ":", "\r")


make/O/N=0 xw,yw

variable p1

for(p1=0;p1<itemsinlist(x1,";");p1+=1)

    insertpoints (numpts(xw)),1, xw

    insertpoints (numpts(yw)),1, yw

    xw[numpts(xw)-1]=str2num(stringfromlist(p1,x1,";"))

    yw[numpts(yw)-1]=str2num(stringfromlist(p1,y1,";"))

endfor

```

```
end
```

```
function/S getcells()
```

```
    if(!datafolderexists("cells"))
```

```
        return ""
```

```
    endif
```

```
    setdatafolder root:cells
```

```
    string cl=replacestring(", ",stringbykey("FOLDERS",datafolderdir(1),":",";")) //data folders
```

```
    setdatafolder root:
```

```
    return cl
```

```
end
```

```
function fGraph0(lasch):buttoncontrol
```

```
    string lasch
```

```
    string n1,n2
```

```
        if(stringmatch(lasch,"glu*"))
```

```
            n1="glut"
```

```
            n2="S_vglut1"
```

```
        elseif(stringmatch(lasch,"gab*"))
```

```
            n1="gab"
```

```
            n2="S_vgat1"
```

```
        endif
```

```

dowindow/K  $\$(lasch)$ 

PauseUpdate; Silent 1           // building window...

Display/K=1 /W=(177,71.75,963.75,761.75)/T/N= $\$(lasch)$   $\$(n1+"yo")$  vs  $\$(n1+"Xo")$  as n1

appendtograph/T  $\$(n1+"yi")$  vs  $\$(n1+"Xi")$ 

appendtograph/T  $\$(n1+"yn")$  vs  $\$(n1+"Xn")$ 

AppendImage/T  $\$(n2)$ 

ModifyImage  $\$(n2)$  ctab= {*,*,Grays,0}

ModifyGraph margin(left)=-1,margin(bottom)=-1,margin(top)=-1,margin(right)=-1


ModifyGraph mode=3

ModifyGraph marker=19

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0


ModifyGraph msize( $\$(n1+"yn")$ )=0.5

ModifyGraph marker( $\$(n1+"yn")$ )=8

ModifyGraph rgb( $\$(n1+"yn")$ )=(34816,34816,34816)

ModifyGraph msize( $\$(n1+"yi")$ )=3

ModifyGraph marker( $\$(n1+"yi")$ )=16

ModifyGraph rgb( $\$(n1+"yi")$ )=(0,52224,52224)

ModifyGraph marker( $\$(n1+"yo")$ )=19

ModifyGraph msize( $\$(n1+"yo")$ )=3

ModifyGraph rgb( $\$(n1+"yo")$ )=(0,26112,26112)


SetAxis/A/R left

```

```
CheckBox $(n1+"A"),pos={21,31},size={68,14},proc=CheckProc,title=n1+ " spot A "
```

```
CheckBox $(n1+"A"),value= 1
```

```
CheckBox $(n1+"N"),pos={21,11},size={69,14},proc=CheckProc,title=n1+ " spot N "
```

```
CheckBox $(n1+"N"),value= 1
```

```
CheckBox $(n1+"I"),pos={21,51},size={64,14},proc=CheckProc,title=n1+ " spot I "
```

```
CheckBox $(n1+"I"),value= 1
```

```
EndMacro
```

```
function fillit1(var,xyr,zr)
```

```
variable var
```

```
variable xyr
```

```
variable zr
```

```
wave M_3DParticleInfo
```

```
variable x1=M_3DParticleInfo[var][0]-xyr
```

```
variable x2=M_3DParticleInfo[var][1]+xyr
```

```
variable y1=M_3DParticleInfo[var][2]-xyr
```

```
variable y2=M_3DParticleInfo[var][3]+xyr
```

```
variable z1=M_3DParticleInfo[var][4]-zr
```

```
variable z2=M_3DParticleInfo[var][5]+zr
```

```
wave cell2//idealised (thresholded) 3d iba img
```

```
duplicate/O/R=[x1,x2][y1,y2][z1,z2] cell2, ts
```

```
end
```

```
function fillit(x1,y1,z1,xyr,zr)
```

```
variable x1//midpoint of a spot x
```

```
variable y1//midpoint of a spot y
```

```
variable z1//midpoint of a spot z
```

```
variable xyr//range xy
```

```
variable zr//range z
```

```
//print x1,y1,z1,xyr,zr
```

```
wave ts
```

```
wave W_iba=root:cellsmo//W_iba//cell2
```

```
variable xoff=dimoffset(w_iba,0)
```

```
variable yoff=dimoffset(w_iba,1)
```

```
variable p1,q1,r1
```

```
for(p1=0; p1<xyr;p1+=1)
```

```

for(q1=0; q1<xyr;q1+=1)

    for(r1=0; r1<zr;r1+=1)

        ts[p1][q1][r1]=w_iba[x1-(xyr-1)/2+p1-xoff][y1-(xyr-1)/2+q1-yoff][z1-(zr-1)/2+r1]

        //print x1-(xyr-1)/2+p1-xoff,y1-(xyr-1)/2+q1-yoff,z1-(zr-1)/2+r1

    endfor

endfor

endfor

end

```

Function CheckProc_1ideal(ctrlName,checked) : CheckBoxControl

String ctrlName//0=gaba,1=glut

Variable checked

if(checked)

if(stringmatch(ctrlName[5],"0"))//GABA

addspots("w_projRGB", "gab")

else

addspots("w_projRGB", "glu")

```

endif

else

if(stringmatch(ctrlName[5],"1"))//glut

removefromgraph/Z/W=w_projRGB#Gyz glutyo,glutyi,glutyn

removefromgraph/Z/W=w_projRGB#Gxy glutyo,glutyi,glutyn

removefromgraph/Z/W=w_projRGB#Gxz glutzo,glutzi,glutzn

else //GABA

removefromgraph/Z/W=w_projRGB#Gyz gabyo,gabyi,gabyn

removefromgraph/Z/W=w_projRGB#Gxy gabyo,gabyi,gabyn

removefromgraph/Z/W=w_projRGB#Gxz gabzo,gabzi,gabzn

endif

endif

End

```

Function CheckProc_1_setdyn(ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

string setvar

if(stringmatch(ctrlName,"*iba"))

NVAR thr=root:thr


```

        setvar="setvar1"

        elseif(stringmatch(ctrlname,"*glu"))

NVAR thr=root:thrGL

        setvar="setvar3"

        elseif(stringmatch(ctrlname,"*gab"))

NVAR thr=root:thrGA

        setvar="setvar4"

    endif


    if(checked)

        thr=3

        SetVariable $(setvar), win=Panel0 ,format="%g *avg ",limits={0,20,1}

    else

        thr=50

        SetVariable $(setvar), win=Panel0 ,format="",limits={0,255,5}

    endif

End

```

```

function showsignal(w1,var)

wave w1

variable var

variable p1

make/O/N=(dimsize(w1,2)) signal

    for(p1=0; p1<(dimsize(w1,2)); p1+=1)

        imagertransform/P=(p1) getplane w1

wave M_ImagePlane

        wavestats/Q M_ImagePlane

```

```

        signal[p1]=v_avg
    endfor

    killwaves/Z m_imageplane

    if(var)

        dowindow/K w_signal

        display/K=1/N=W_signal      signal as "z signal of "+nameofwave(w1)

    endif

end

```

```

function thresh3d (w1,GG)

wave w1

string GG// glu or GABA

//print nameofwave(w1),dimsize(w1,0),dimsize(w1,1),gg,"Y"

string spX=""

string spY=""

string spz=""

    if(stringmatch (GG, "glu")) //vglut staining

NVAR thr= :thrGL //manual thr value

NVAR min1=:glu_min

NVAR max1=:glu_max

        spX="glutx"

        spY="glutY"

```

```

        spz="glutz"

wave W_gg=:w_vglut

        elseif(stringmatch (GG, "gab")) //vgat staining

NVAR thr=:thrGA //manual thr value

NVAR min1=:gab_min

NVAR max1=:gab_max

        spX="gabx"

        spY="gabY"

        spz="gabz"

wave W_gg=:w_vgat

        endif


controlinfo/W=panel0 $("check_" +gg)

variable dyn=0

        if(v_value)//checked

                dyn=1


wave signal=:$("signal_" +gg)

        else

                //print gg,"nope"

        endif


//print thr,gg

        duplicate/O w1, test44,test45

        test44=0

        test45=0

variable p1

        for(p1=0; p1<(dimsize(w1,2)); p1+=1)

                imagertransform/P=(p1) getplane w1

wave M_ImagePlane

wave signal

```

```

                                wavestats/Q W_gg

if(dyn)

//print p1, thr,signal[p1]

                                imagethreshold/Q/T=(thr*signal[p1])/M=0/I M_ImagePlane

                                //print p1,thr,signal[p1],thr*signal[p1]

else

                                thr=3*ceil(v_avg)

                                imagethreshold/Q/T=(thr)/M=0/I M_ImagePlane

endif

wave M_imagethresh

M_ImagePlane*=M_imagethresh

imagetransform/P=(p1)/D=M_ImagePlane setplane test44

imagetransform/P=(p1)/D=M_imagethresh setplane test45

endfor

end

```

Function ButtonProc_1(ctrlName) : ButtonControl

String ctrlName

```

wave w_iba=root:w_iba

nvar thr

make/O/N=(dimsize(w_iba,2)) dx1,dy1

variable p1

    for(p1=0; p1<(dimsize(w_iba,2)); p1+=1)

        imagetransform/P=(p1) getplane w_iba

wave M_ImagePlane

        imagethreshold/Q/M=0/T=(thr)/O M_ImagePlane

        wavestats/Q M_ImagePlane

        dx1[p1]=p1

        dy1[p1]=v_sum/255

    endfor

dowindow/K w_distr

display /K=1/N=w_distr dy1 vs dx1

dowindow/K t_distr

edit/K=1/N=t_distr dx1, dy1

End

```

```

function getsignal(cell1)

wave cell1

    duplicate/O cell1,test

wave test

    redimension/S test

    wavestats/Q test

variable t1=ceil(V_avg)

    test/=(test[p][q]>3*t1)

    showsignal(test,0)

wave signal

```

```

CurveFit/Q/L=(numpts(signal)) /NTHR=0 poly 3, signal /D

wave fit_signal

end

#pragma rtGlobals=3           // Use modern global access method and strict wave access.


function qSpots(vl1,vt1,vr1,vb1,GG)

variable vl1

variable vt1

variable vr1

variable vb1

string GG //either glu or gab


//variable m1=0 //method threshold; manual

NVAR xy// xy range; odd number

NVAR zr//z range; odd number


string s_3DspotNN="" // name of cell-filtered 3D spot wave

string s_spotNN="" ///max projection of s_3DspotNN

string spX=""

string spY=""

string spz=""

```

```

        if(stringmatch (GG, "glu")) //vglut staining

wave w_3Dspot=:w_vGlut//3d wave containing all spots

NVAR thr=:thrGL //manual thr value

        s_3DspotNN="glu1" // name of cell-filtered 3D spot wave

        s_spotNN="S_vglut1" ///max projection of s_3DspotNN


NVAR min1=:glu_min

NVAR max1=:glu_max

        spX="glutx"

        spY="gluty"

        spz="glutz"

NVAR smo1=:smogl1

        make/O/N=0 glutXo,glutYo,glutZo,glutZo1

        make/O/N=0 glutXi,glutYi,glutZi,glutZi1

        make/O/N=0 glutXn,glutYn,glutZn,glutZn1

wave Ox=:glutXo

wave Oy=:glutYo

wave Oz=:glutZo

wave Oz1=:glutZo1

wave lx=:glutXi

wave ly=:glutYi

wave lz=:glutZi

wave lz1=:glutZi1

NVAR fl=:glut_fl

NVAR sp=:glut_sp

NVAR spN=:glut_spN

NVAR spA=:glut_spA

NVAR spl=:glut_spl

wave nx=:glutxn

wave ny=:glutyn

wave nz=:glutzn

```

```
wave nz1=:glutzn1
```

```
elseif(stringmatch (GG, "gab")) //vgat staining
```

```
wave w_3Dspot=:w_vGat//3d wave containing all spots
```

```
NVAR thr=:thrGA //manual thr value
```

```
    s_3DspotNN="gab1" // name of cell-filtered 3D spot wave
```

```
    s_spotNN="S_vgat1" ///max projection of s_3DspotNN
```

```
NVAR min1=:gab_min
```

```
NVAR max1=:gab_max
```

```
    spX="gabx"
```

```
    spY="gaby"
```

```
    spz="gabz"
```

```
NVAR smo1=:smoga1
```

```
    make/O/N=0 gabXo,gabYo,gabZo,gabZo1
```

```
    make/O/N=0 gabXi,gabYi,gabZi,gabZi1
```

```
    make/O/N=0 gabXn,gabYn,gabZn,gabZn1
```

```
wave Ox=:gabXo
```

```
wave Oy=:gabYo
```

```
wave Oz=:gabZo
```

```
wave Oz1=:gabZo1
```

```
wave lx=:gabXi
```

```
wave ly=:gabYi
```

```
wave lz=:gabZi
```

```
wave lz1=:gabZi1
```

```
NVAR fl=:gaba_fl
```



```

NVAR sp=:gaba_sp

NVAR spN=:gaba_spN

NVAR spA=:gaba_spA

NVAR spl=:gaba_spl

wave nx=:gabxn

wave ny=:gabyn

wave nz=:gabzn

wave nz1=:gabzn1

endif


//      print      gg,dimoffset(w_3dsm,0), dimoffset(w_3dsm,1),nameofwave (w_3dsm)//seedx

wave cell2 ///mask; IBA1-based; 3D, single cell

      duplicate/O/R=[v1,vr1][vt1,vb1] w_3Dspot, $(s_3DspotNN)      //W_vglut into glu1(w_3dsm)

wave w_3dsm=:(s_3DspotNN) ///3D wave spots; mask; glu1

      //dynamic

      controlinfo/W=panel0 $("check_" +gg)

variable dyn=0

      if(v_value)//checked

            dyn=1

            showsignal(w_3dsm,0)


wave signal

      duplicate/O signal, $("signal_" +gg)

endif


variable scale=wavemax(w_3dsm)

//      w_3dsm/=(scale/255)

      redimension/B/U w_3dsm //glu1

```

```

        //smooth

    if(smo1)

        //print smo1,"F"

        imagefilter/N=(smo1)/O gauss3d w_3dsm

    endif


    thresh3d (w_3dsm,GG)

wave test45

    imageanalyzeParticles/A=(min1)/MAXA=(max1) stats test45

wave M_3DParticleInfo


    w_3dsm*=cell2 //masking

    imagetransform zprojection w_3dsm

wave M_zProjection

    duplicate/O M_zProjection, $(s_spotNN) //s_vglut1

wave w_sm1=$(s_spotNN)      //max proj of 3D masked spot wave

//      setscale/P x 0,1, w_sm1

//      setscale/P y 0,1, w_sm1


    make/O/N=(dimsize(M_3DParticleInfo,0)) $(spX),$(spY),$(spZ)

wave spX1=$(spX)//midpoint of spot

wave spY1=$(spY)//midpoint of spot

wave spZ1=$(spZ)//midpoint of spot


    spX1=M_3DParticleInfo[p][0]+(M_3DParticleInfo[p][1]-M_3DParticleInfo[p][0])/2

    spY1=M_3DParticleInfo[p][2]+(M_3DParticleInfo[p][3]-M_3DParticleInfo[p][2])/2

    spZ1=M_3DParticleInfo[p][4]+(M_3DParticleInfo[p][5]-M_3DParticleInfo[p][4])/2


    spX1+=dimoffset(w_3dsm,0)

    spY1+=dimoffset(w_3dsm,1)


    //check spots

```

variable p1

```
for(p1=0; p1<numpnts(spz1); p1+=1)
```

```
  fillIT1(p1,x,yr,zr)
```

wave ts

```
  wavestats/q ts
```

```
  if(!v_avg) ///no touch
```

```
    insertpoints (numpnts(nx)),1, nx,ny,nz
```

```
    nx[numpnts(nx)-1]=spX1[p1]
```

```
    ny[numpnts(ny)-1]=spy1[p1]
```

```
    nz[numpnts(nz)-1]=spz1[p1]
```

```
  elseif(v_avg<0.8) ///out
```

```
    insertpoints (numpnts(Ox)),1, ox,oy,oz
```

```
    ox[numpnts(ox)-1]=spx1[p1]
```

```
    oy[numpnts(oy)-1]=spy1[p1]
```

```
    oz[numpnts(oz)-1]=spz1[p1]
```

```
  elseif(v_avg>=0.8) //in
```

```
    insertpoints (numpnts(ix)),1, ix,iy,iz
```

```
    ix[numpnts(ix)-1]=spx1[p1]
```

```
    iy[numpnts(iy)-1]=spy1[p1]
```

```
    iz[numpnts(iz)-1]=spz1[p1]
```

```
  endif
```

```
endfor
```

```
///scaling stuff
```

```
  duplicate/o oz, $(nameofwave(oz1))
```

```
  duplicate/o iz, $(nameofwave(iz1))
```

```
  duplicate/o nz, $(nameofwave(nz1))
```

```
  if(numpnts(oz1))
```

```

        oz1=dimsize(w_3dsm,2)-oz
    endif

    if(numpts(iz1))

        iz1=dimsize(w_3dsm,2)-iz
    endif

    if(numpts(nz1))

        nz1=dimsize(w_3dsm,2)-nz
    endif

variable xoff=dimoffset(cell2,0)

variable yoff=dimoffset(cell2,1)

```

```
//results
```

```
sp=numpts($(spX))
```

```
spN=numpts(nX)
```

```
spA=numpts(oX)
```

```
spl=numpts(iX)
```

```
duplicate/O w_3dsm, test
```

```
redimension/D test
```

```
test/=(test[p][q][r]>0)
```

```
wavestats/Q test
```

```
fl=V_avg
```

```
killwaves/Z test
```

end

Function ButtonProc_4Ibmask(ctrlName) : ButtonControl

String ctrlName

wave mask=:cell2//iba mask

wave curw=:all_DAPI2//without mask

duplicate/O curw,temp44dapi

wave cell1=:temp44dapi

cell1*=(mask>0)

imagetransform/METH=1 xprojection cell1

wave M_xprojection

imagetransform/METH=1 zprojection cell1

wave M_zprojection

imagetransform/METH=1 yprojection cell1

wave M_yprojection

duplicate/O M_xprojection, dapiX

duplicate/O M_yprojection, dapiY

duplicate/O M_zprojection, dapiZ

ImageRotate/A=90/O dapix

killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh

End

Function ButtonProc_restore(ctrlName) : ButtonControl

String ctrlName

if(stringmatch(ctrlName,"*dapi"))

wave cell1=:all_DAPI2

else

wave cell1=:cell1

endif

imagetransform/METH=1 xprojection cell1

wave M_xprojection

imagetransform/METH=1 zprojection cell1

wave M_zprojection

imagetransform/METH=1 yprojection cell1

wave M_yprojection

if(stringmatch(ctrlName,"*dapi"))

duplicate/O M_xprojection, dapiX

duplicate/O M_yprojection, dapiY

duplicate/O M_zprojection, dapiZ

```

        ImageRotate/A=90/O dapix
    else

        duplicate/O M_xprojection, cellX

        duplicate/O M_yprojection, cellY

        duplicate/O M_zprojection, cellZ

        ImageRotate/A=90/O cellx

    endif

    killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh


    if(stringmatch(ctrlname,"*dapi"))

        duplicate/O cell1, temp44dapi

    else

        duplicate/O cell1, temp44//temp44=cell1*mask

    endif

End

#pragma rtGlobals=3          // Use modern global access method and strict wave access.


function genMask(cell1)

wave cell1 ///n layers


variable nl=dimsize(cell1,2)//number of layers

NVAR thr

```

NVAR smo

variable p1

duplicate/O cell1, cell2,cellsmo

cell2=0

if(smo>0)

for(p1=0;p1<smo;p1+=1)

Imagefilter/N=(3)/O gauss3d cellsmo

endfor

endif

controlinfo/W=panel0 check_iba

variable dyn=0

if(v_value)//checked

dyn=1

wave W_iba

showsignal(w_iba,0)

wave signal

endif

for(p1=0;p1<nl;p1+=1)

imagetransform/P=(p1) getplane cellsmo

wave M_ImagePlane

if(dyn)

imagethreshold/Q/T=(thr*signal[p1])/M=0 M_ImagePlane

else


```

                                imagethreshold/Q/T=(thr)/M=0 M_ImagePlane
                                endif

wave m_imagethresh

                                imagetransform/P=(p1)/D=m_imagethresh setplane cell2

                                endfor

                                killwaves/Z signal

                                //ImageSeedFill/B=0 seedX=526,seedY=327,seedZ=25,target=255,min=30,max=255,fillnumber=1,srcWave=cell1

                                cell2/=255

                                controlinfo/W=Panel0 check0

                                if(V_value)//nucleus excluded

wave mask_dapi

                                if(waveexists(mask_dapi))

                                    cell2*=(mask_dapi)

                                    wavestats/Q          mask_dapi

variable/G dapi_ex=1

variable/G nuc_size= v_npnts-v_sum

                                else

variable/G dapi_ex=0

variable/G nuc_size=nan

                                endif

                                else

variable/G dapi_ex=0

variable/G nuc_size=nan

                                endif

```

```

        imagetransform zprojection cell2
wave M_zProjection

        duplicate/O M_zProjection, cell2d

        imagetransform zprojection cellsmo
wave M_zProjection

        duplicate/O M_zProjection, cellsmod

```

```

wave glu1

        imagetransform zprojection glu1
wave M_zProjection

        duplicate/O M_zProjection, all_vglut

        redimension/B/U all_vglut

//middle img
NVAR smo

        duplicate/O glu1, glu2

```

```

wave temp44

        if(waveexists(temp44))

                glu2*=(temp44>0)

        endif

        if(smo>0)

                for(p1=0;p1<smo;p1+=1)

                        Imagefilter/N={3}/O gauss3d glu2

                endfor

        endif

        imagetransform zprojection glu2

```

```
duplicate/O M_zProjection, S_vglut1
```

```
//right img
```

```
duplicate/O glu2, glu2M
```

```
wave cell2
```

```
glu2m*=cell2
```

```
imagetransform zprojection glu2m
```

```
duplicate/O M_zProjection, S_vglut2
```

```
/////////DAPI
```

```
wave celldapi
```

```
imagetransform zprojection celldapi
```

```
duplicate/O M_zProjection, all_dapi
```

```
redimension/B/U all_dapi
```

```
//middle img
```

```
NVAR smo
```

```
duplicate/O celldapi, all_dapi2
```

```
wave mask44
```

```
if(waveexists(mask44))
```

```
all_dapi2*=mask44
```

```
endif
```

```
if(smo>0)
```

```
for(p1=0;p1<smo;p1+=1)
```

```
Imagefilter/N=(3)/O gauss3d all_dapi2
```

```
endfor
```

```
endif
```

```

        imagetransform zprojection all_dapi2

        duplicate/O M_zProjection, all_dapiP2

//Right

//wave mask44dapi

        duplicate/O all_dapi2, all_dapi2m

        if(waveexists(mask_dapi))

                all_dapi2m*=(mask_dapi==0)

        endif

        if(smo>0)

                for(p1=0;p1<smo;p1+=1)

                        Imagefilter/N={3}/O gauss3d all_dapi2m

                endfor

        endif

        imagetransform zprojection all_dapi2m

        duplicate/O M_zProjection, all_dapi2mp

end

function sc1()

wave cm1

wave cm1

wave pl0,pl1,pl2

```

```

controlinfo/W=panel0 check2

if(V_value)

wave tmp44

    if(waveexists(tmp44))

        imagetransform/METH=3 zprojection tmp44

wave m_zProjection

        pl0*=(m_zProjection==0)

        pl1*=(m_zProjection==0)

        pl2*=(m_zProjection==0)

        killwaves/Z M_zprojection

    endif

endif

imagetransform/P=0/D=pl0 setplane cm1

imagetransform/P=1/D=pl1 setplane cm1

imagetransform/P=2/D=pl2 setplane cm1

end

```

```

function makeRGB(doit)

variable doit

wave gc=cellsmod///IBA

```

```

controlinfo/W=W_scale check0///GABBR

```

```
        if(v_value)
```

```
    wave rc=s_vglut2
```

```
        else
```

```
    wave rc=s_vglut1
```

```
        endif
```

```
controlinfo/W=W_scale check1//DAPI
```

```
        if(v_value)
```

```
    wave bc=all_dapi2mp
```

```
        else
```

```
    wave bc=all_dapip2
```

```
        endif
```

```
duplicate/O rc,pl0
```

```
duplicate/o gc,pl1
```

```
duplicate/o bc,pl2
```

```
if(doit>0)
```

```
make/O/N=(dimsize(pl0,0),dimsize(pl0,1),3)/b/U cm1=0
```

```
imagehistogram rc
```

```
wave W_imageHist
```

```
duplicate/O W_imageHist,h_rc
```

```
imagehistogram gc
```

```
duplicate/O W_imageHist,h_gc
```

```
imagehistogram bc
```

```
duplicate/O W_imageHist,h_bc
```

```
endif
```

```
        scaleT(0)

        scaleT(1)

        scaleT(2)

        sc1()

end
```

```
function scale_glob()

variable/G gmin,gmax=255

variable/G rmin,rmax=255

variable/G bmin,bmax=255

end
```

```
function scaleT(plane)

variable plane


        if(plane==0)//red channel==GABBR

wave curw=:pl0

NVAR min1=root:rmin

NVAR max1=root:rmax


        elseif(plane==1)//green channel==iba1

wave curw=:pl1
```

```

NVAR min1=root:gmin

NVAR max1=root:gmax

        elseif(plane==2)//blue channel==DAPI

wave curw=:pl2

NVAR min1=root:bmin

NVAR max1=root:bmax

        endif

        //print min1,max1,nameofwave(curw),plane

        redimension/S curw

        curw-=min1

        curw*=(curw[p][q]>=0)


        curw/=(max1-min1)

        curw=curw^(curw<1)

        curw*=255


        redimension/B/U curw

end

```

Function SliderProc_scale(ctrlName,sliderValue,event) : SliderControl


```

String ctrlName

Variable sliderValue

Variable event      // bit field: bit 0: value set, 1: mouse down, 2: mouse up, 3: mouse moved

makeRGB(0)

return 0

End

```

```

Function SetVarProc_scale(ctrlName,varNum,varStr,varName) : SetVariableControl

String ctrlName

Variable varNum

String varStr

String varName

makeRGB(0)

End

```

```

Function ButtonProc_2(ctrlName) : ButtonControl

String ctrlName

fScale()

End

```

redimension/N=(numpnts(cell2))/S glu2c

```
glu2c/=(glu2c[p]!=0)
```

```
wavestats/Q glu2c
```

```
print "cell:",v_npnts,v_avg,v_sdev
```

```
redimension/N=(numpnts(cell2))/S glu2o
```

```
glu2o/=(glu2o[p]>0)
```

```
wavestats/Q glu2o
```

```
print "outside:",v_npnts,v_avg,v_sdev
```

```
statsttest/T=1/Q glu2c,glu2o
```

```
wave w_statsttest
```

```
print "p =", w_statsttest[9]
```

```
make/O h_c,h_o
```

```
Histogram/B={0,1,255} glu2o,h_o
```

```
Histogram/B={0,1,255} glu2c,h_c
```

```
end
```

```
function deletenan(curw)
```

```
wave curw
```

```
variable p1
```

```
for(p1=numpnts(curw);p1==0;p1-=1)
```

```
if(curw[p1]<=0)
```

```
deletepoints p1,1,curw
```

```
endif
```

```
endfor
```

```
end
```

```
function fw_projDAPI()
```

```
    dowindow/K w_projDAPI
```

```
    PauseUpdate; Silent 1           // building window...
```

```
    NewPanel /W=(671,133,1443,905)/K=1 /N=w_projDAPI as "DAPI Projection"
```

```
    SetDrawLayer UserBack
```

```
    SetDrawEnv arrow= 1
```

```
    DrawLine 385,370,385,20
```

```
    SetDrawEnv arrow= 1
```

```
    DrawLine 400,385,750,385
```

```
    SetDrawEnv fsize= 18
```

```
    DrawText 755,395,"x"
```

```
    SetDrawEnv fsize= 18
```

```
    DrawText 381,770,"z"
```

```
    SetDrawEnv dash= 1,arrow= 1
```

```
    DrawLine 370,385,20,385
```

```

SetDrawEnv dash= 1,arrow= 1

DrawLine 385,400,385,750

SetDrawEnv fsize= 18

DrawText 10,395,"z"

SetDrawEnv fsize= 18

DrawText 381,17,"y"

GroupBox group0,pos={20,400},size={350,350},labelBack=(34816,34816,34816)

//CheckBox check0,pos={36,418},size={43,14},proc=CheckProc_1_setw,title="Crop "

//CheckBox check0,value= 1

Button button0dapi,pos={100,450},size={150,40},proc=ButtonProc_cropDAPI,title="Delete"

Button button1dapi,pos={100,500},size={150,40},proc=ButtonProc_cropDAPI,title="Crop"

Button button4,pos={296,467},size={50,50},proc=ButtonProc_4lbamask,title="lba mask"

Button button4,fColor=(0,39168,0)

Button button2dapi,pos={100,600},size={150,40},proc=ButtonProc_restore,title="Restore"

Button button3dapi,pos={100,650},size={150,40},proc=ButtonProc_useit,title="Use"

SetVariable setvar0,pos={121,417},size={108,16},bodyWidth=40,proc=SetVarProc_filter,title="Avg Img Filter"

SetVariable setvar0,limits={3,inf,2},value= imgF

```

wave dapiZ, dapiX,dapiY

```

Display/W=(400,20,750,370)/HOST=#

AppendImage/T dapiZ

ModifyImage dapiZ ctab= {*,*,Grays,0}

ModifyGraph margin=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,Gxy

SetActiveSubwindow ##

```

Display/W=(20,20,370,370)/HOST=#

AppendImage/T dapiX

ModifyImage dapiX ctab= {*,*,Grays,0}

ModifyGraph margin=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

SetAxis/A top

RenameWindow #,Gyz

SetActiveSubwindow ##

Display/W=(400,400,750,750)/HOST=#

AppendImage/T dapiY

ModifyImage dapiY ctab= {*,*,Grays,0}

ModifyGraph margin=-1

ModifyGraph mirror=0

ModifyGraph nticks=0

ModifyGraph noLabel=2

ModifyGraph standoff=0

ModifyGraph axThick=0

SetAxis/A/R left

RenameWindow #,Gxz

SetActiveSubwindow ##

setwindow W_projDAPI, hook=hook_crop,hookevents=7

make/O/N=0 xw,yw

```
GraphWaveDraw /W=$( "W_projDAPI" )#$( "Gyz" ) /L/T/O yW, xW
```

```
End
```

```
Function ButtonProc_cropDAPI(ctrlName) : ButtonControl
```

```
String ctrlName
```

```
string rem=""
```

```
wave dapiz
```

```
string o1=note(dapiz)
```

```
variable it1=itemsinlist(o1,"t")
```

```
rem+="Action_"+num2str(it1)+"\r"
```

```
wave temp44dapi
```

```
if(!waveexists(temp44dapi))
```

```
wave all_DAPI2M
```

```
duplicate/o all_DAPI2M, temp44dapi
```

```
endif
```

```
wave xw,yw
```

```
string pr=note(xw)
```

```
rem+=pr+"\r" //1st row:window
```

```
if(stringmatch(pr,"Gxy"))
```

```
wave curw=root:dapiz
```

```
endif
```

```
if(stringmatch(pr,"Gxz"))
```

```
wave curw=root:dapiY
```

```
endif
```

```
if(stringmatch(pr,"Gyz"))
```

```
wave curw=root:dapiX
```

```
endif
```

```

ImageBoundaryToMask
height=dimsize(curw,1),scalingWave=curw,seedy=dimoffset(curw,1),seedx=dimoffset(curw,0), xwave=xw, ywave=yw
width=dimsize(curw,0),

wave M_ROIMask

```

```

    if(stringmatch (ctrlName,"*1dapi"))//crop

        M_Roimask=(M_Roimask[p][q]==0)

        rem+="crop_1\r"

    else ///delete, as it is

        rem+="delete_2\r"

    endif

    if(stringmatch(pr,"Gxy"))

        temp44dapi*=m_roimask[p][q]

    endif

    if(stringmatch(pr,"Gxz"))

        temp44dapi[][]*=m_roimask[p][r]

    endif

    if(stringmatch(pr,"Gyz"))

        imagetransform fliprows m_roimask

        temp44dapi[][]*=m_roimask[r][q]

    endif

    endif

    imagetransform zprojection temp44dapi

wave M_zprojection

    imagetransform xprojection temp44dapi

wave M_xprojection

    imagetransform yprojection temp44dapi

wave M_yprojection

    duplicate/O M_zprojection, dapiZ

    duplicate/O M_xprojection, dapiX

```

duplicate/O M_yprojection, dapiY

ImageRotate/A=90/O dapix

killwaves/Z M_xprojection,M_yprojection,M_zprojection, M_imagethresh

rem+="xw:"+w2str(xw)+"\r"

rem+="yw:"+w2str(yw)+"\r"

note dapiz, o1+rem

End

Function CheckProc_2chowAna(ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

setdrawlayer/W=panel0#G0/K userfront

if(checked)

string exp1=getcells()

variable p1


```

        for(p1=0;p1<itemsinlist(exp1,";");p1+=1)

SVAR marq=root:cells:$(stringfromlist(p1,exp1,";")):marq

variable l1=str2num(stringfromlist(0,marq,";"))

variable r1=str2num(stringfromlist(1,marq,";"))

variable t1=str2num(stringfromlist(2,marq,";"))

variable b1=str2num(stringfromlist(3,marq,";"))

                setdrawenv/W=panel0#G0 xcoord=top,ycoord=left,save

                SetDrawEnv/W=panel0#G0 fillpat= 0,linethick= 2.00,save

                SetDrawEnv/W=panel0#G0 linefgc= (65535,0,0),save

                drawrect/W=panel0#G0 l1,t1,r1,b1

                SetDrawEnv/W=panel0#G0 textrgb= (65535,0,0)

                drawtext/W=panel0#G0 l1,t1, stringfromlist(p1,exp1,";")

        //      print p1, stringfromlist(p1,exp1,";")

        endfor

    endif

End

```

```

function fScale()

    scale_glob()

    makeRGB(1)

wave cm1

dowindow/K W_scale

PauseUpdate; Silent 1      // building window...

```

Display /W=(340,90,1000,584)/N=W_scale/K=1 as " "

AppendImage/T cm1

ModifyImage cm1 ctab= {*,*,Grays,0}

ModifyGraph margin(left)=14,margin(bottom)=14,margin(top)=14,margin(right)=14

ModifyGraph mirror=2

ModifyGraph nticks=6

ModifyGraph minor=1

ModifyGraph noLabel=2

ModifyGraph fSize=8

ModifyGraph standoff=0

ModifyGraph axThick=0

ModifyGraph tkLblRot(left)=90

ModifyGraph btLen=3

ModifyGraph tlOffset=-2

ModifyGraph axisEnab(top)={0,0.7}

SetAxis/A/R left

NVAR gmin,gmax

GroupBox group2,pos={617,20},size={255,200},frame=0,fColor={13056,0,0}

GroupBox group0,pos={620,22},size={249,120},title="Iba1"

GroupBox group0,labelBack=(47872,47872,47872)

SetVariable setvar0,pos={633,40},size={50,16},bodyWidth=40,title="min"

SetVariable setvar0,limits={-inf,inf,1},value=gmin,proc=SetVarProc_scale

Slider slider0,pos={623,60},size={244,19},variable=gmin

Slider slider0,limits={0,255,1},value= 0,vert= 0,ticks= 0,proc=SliderProc_scale

SetVariable setvar1,pos={633,90},size={50,16},bodyWidth=40,title="max"

SetVariable setvar1,limits={-inf,inf,1},value=gmax,proc=SetVarProc_scale

Slider slider1,pos={623,110},size={244,19}, variable=gmax

Slider slider1,limits={0,255,1},value= 255,vert= 0,ticks= 0,proc=SliderProc_scale

NVAR rmin,rmax

GroupBox group3,pos={617,230},size={255,200},frame=0,fColor={13056,26112,0}

GroupBox group1,pos={620,232},size={249,120},title="GABBR"

GroupBox group1,labelBack=(47872,47872,47872)

SetVariable setvar2,pos={633,250},size={50,16},bodyWidth=40,title="min"

SetVariable setvar2,limits={-inf,inf,1},value=rmin,proc=SetVarProc_scale

Slider slider2,pos={623,270},size={244,19},variable =rmin

Slider slider2,limits={0,255,1},value= 0,vert= 0,ticks= 0,proc=SliderProc_scale

Button button0,pos={768,251},size={50,20},proc=ButtonProc_3,title="Threshold"

SetVariable setvar3,pos={633,300},size={50,16},bodyWidth=40,title="max"

SetVariable setvar3,limits={-inf,inf,1},value=rmax,proc=SetVarProc_scale

Slider slider3,pos={623,320},size={244,19},variable=rmax

Slider slider3,limits={0,255,1},value= 255,vert= 0,ticks= 0,proc=SliderProc_scale

CheckBox check0,pos={769,357},size={95,14},title="Only intracellular",value= 0, proc=check_scale

NVAR bmin,bmax

GroupBox group4,pos={617,440},size={255,200},frame=0,fColor={13056,26112,0}

GroupBox group5,pos={620,442},size={249,120},title="DAPI"

GroupBox group5,labelBack=(47872,47872,47872)

SetVariable setvar2d,pos={633,460},size={50,16},bodyWidth=40,title="min"

SetVariable setvar2d,limits={-inf,inf,1},value=bmin,proc=SetVarProc_scale

Slider slider2d,pos={623,490},size={244,19},variable =bmin

Slider slider2d,limits={0,255,1},value= 0,vert= 0,ticks= 0,proc=SliderProc_scale

SetVariable setvar3d,pos={633,510},size={50,16},bodyWidth=40,title="max"

SetVariable setvar3d,limits={-inf,inf,1},value=bmax,proc=SetVarProc_scale

Slider slider3d,pos={623,530},size={244,19},variable=bmax

Slider slider3d,limits={0,255,1},value= 255,vert= 0,ticks= 0,proc=SliderProc_scale

```
CheckBox check1,pos={769,577},size={95,14},proc=check_scale,title="Only intracellular"
```

```
CheckBox check1,value= 0
```

wave h_gc,h_rc,h_bc

```
Display/W=(0.7,0.23,0.99,0.33)/HOST=# h_gc
```

```
ModifyGraph margin=3,frameInset=1
```

```
ModifyGraph mode=5
```

```
ModifyGraph rgb(h_gc)=(13056,26112,0)
```

```
ModifyGraph hbFill=2
```

```
ModifyGraph noLabel=2
```

```
ModifyGraph axThick=0
```

```
SetAxis left 0,1000
```

```
SetAxis bottom 2,255
```

```
RenameWindow #,Gg
```

```
SetActiveSubwindow ##
```

```
Display/W=(0.7,0.55,0.99,0.65)/HOST=# h_rc
```

```
ModifyGraph margin=3,frameInset=1
```

```
ModifyGraph mode=5
```

```
ModifyGraph rgb(h_rc)=(65000,0,0)
```

```
ModifyGraph hbFill=2
```

```
ModifyGraph noLabel=2
```

```
ModifyGraph axThick=0
```

```
SetAxis left 0,1000
```

```
SetAxis bottom 2,255
```

```
RenameWindow #,Gr
```

```
SetActiveSubwindow ##
```

```

Display/W=(0.7,0.87,0.99,0.97)/HOST=# h_bc

ModifyGraph margin=3,frameInset=1

ModifyGraph mode=5

ModifyGraph rgb=(0,12800,52224)

ModifyGraph hbFill=2

ModifyGraph noLabel=2

ModifyGraph axThick=0

SetAxis left 0,1000

SetAxis bottom 2,255

RenameWindow #,Gb

SetActiveSubwindow ##

End

```

```

Function check_scale(ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

makeRGB(0)

End

```

```

#pragma rtGlobals=3           // Use modern global access method and strict wave access.

```

```
Function ButtonProc_3A(ctrlName) : ButtonControl
```

```
    String ctrlName
```

```
    NVAR rmina
```

```
    NVAR v_thr
```

```
    NVAR v_extra
```

```
        rmina=round(v_extra+v_thr)
```

```
        makeRGBall(0)
```

```
End
```

```
function scaleITall(plane)
```

```
variable plane
```

```
    if(plane==0)//red channel==GABBR
```

```
    wave curw=:pl0all
```

```
    NVAR min1=root:rminA
```

```
    NVAR max1=root:rmaxA
```

```
    elseif(plane==1)//green channel==iba1
```

```
    wave curw=:pl1all
```

```
    NVAR min1=root:gminA
```

```
    NVAR max1=root:gmaxA
```

```
    elseif(plane==2)//blue channel==DAPI
```

```
    wave curw=:pl2all
```

```
    NVAR min1=root:bminA
```

```
    NVAR max1=root:bmaxA
```

```
    endif
```

```
    //print min1,max1,nameofwave(curw),plane
```

```

redimension/S curw

curw-=min1

curw*=(curw[p][q]>=0)


curw/=(max1-min1)

curw=curw^(curw<1)

curw*=255


redimension/B/U curw

end


function sc1all()


wave cm1all

wave pl0all,pl1all,pl2all

    if(waveexists(pl0all))

        imagetransform/P=0/D=pl0all setplane cm1all

    endif

    if(waveexists(pl1all))

        imagetransform/P=1/D=pl1all setplane cm1all

    endif

    if(waveexists(pl2all))

        imagetransform/P=2/D=pl2all setplane cm1all

    endif

end


function makeRGBall(var)

variable var

```

```
wave gca=:allcell  
wave rca=:all_vglutA  
wave bca=:all_dapiA
```

```
    if(var==0)  
        duplicate/O rca, pl0all  
        scaleITall(0)  
    elseif    (var==1)  
        duplicate/o gca,pl1all  
        scaleITall(1)  
    elseif    (var==2)  
        duplicate/O bca, pl2all  
        scaleITall(2)  
  
    else  
        duplicate/O rca, pl0all  
        duplicate/o gca,pl1all  
        duplicate/O bca, pl2all  
        scaleITall(0)  
        scaleITall(1)  
        scaleITall(2)  
    endif  
    sc1all()  
  
end
```

Function SliderProc_scaleA(ctrlName,sliderValue,event) : SliderControl

```
String ctrlName  
Variable sliderValue
```



```
Variable event      // bit field: bit 0: value set, 1: mouse down, 2: mouse up, 3: mouse moved
```

```
if(event==4)
```

```
if((stringmatch(ctrlName,"*0"))||(stringmatch(ctrlName,"*1")))//Iba
```

```
    makeRGBall(1)
```

```
elseif((stringmatch(ctrlName,"*2"))||(stringmatch(ctrlName,"*3")))//GABBR
```

```
    makeRGBall(0)
```

```
elseif(stringmatch(ctrlName,"*d"))//DAPI
```

```
    makeRGBall(2)
```

```
endif
```

```
endif
```

```
return 0
```

```
End
```

```
Function SetVarProc_scaleA(ctrlName,varNum,varStr,varName) : SetVariableControl
```

```
String ctrlName
```

```
Variable varNum
```

```
String varStr
```

```
String varName
```

```
if(stringmatch(varname,"g*"))//Iba
```

```
    makeRGBall(1)
```

```
elseif(stringmatch(varname,"r*"))//GABBR
```

```
    makeRGBall(0)
```

```
elseif(stringmatch(varname,"b*"))//DAPI
```

```
    makeRGBall(2)
```

```
endif
```

```
End
```

```
Function ButtonProc_useit(ctrlName) : ButtonControl
```

```
    String ctrlName
```

```
    wave temp44      , temp44dapi
```

```
    if(stringmatch(ctrlName,"*dapi"))
```

```
        wave dapi=:celldapi
```

```
            duplicate/O dapi,      temp45dapi
```

```
        NVAR smo
```

```
            if(smo)
```

```
                variable p1
```

```
                    for(p1=0;p1<smo;p1+=1)
```

```
                        imagefilter/N=3/O gauss temp45dapi
```

```
                    endfor
```

```
            endif
```

```
            imagethreshold/Q/M=1/O/I temp45dapi
```

```
            temp45dapi/=255
```

```
            imagethreshold/Q/O/M=1/I temp44dapi
```

```
            temp44dapi/=255
```

```
            temp44dapi*=temp45dapi
```

```
            killwaves/Z temp45dapi
```

```
            duplicate/O temp44dapi,mask44dapi
```

```
        //      mask44dapi=(mask44dapi>0)
```

```
            duplicate/O mask44dapi,mask_dapi
```

```
            checkbox check0,win=panel0, value=1
```

```

else

    duplicate/O temp44,mask44

    mask44=(mask44>0)

endif

genMask(temp44)

NVAR res_size

wave cell2

    wavestats/Q cell2

    res_size=v_sum

End

#pragma rtGlobals=3          // Use modern global access method and strict wave access.

```

```

Function ButtonProc_2reanalyse(ctrlName) : ButtonControl

    String ctrlName

    controlinfo/W=panel0 check0

variable en=v_value

    controlinfo/W=panel0 check2

variable os=v_value

variable p1

string fl=getcells()

    for(p1=0;p1<itemsinlist(fl,";");p1+=1)//1;p1+=1)//

    //load cell

```

```

        PopMenuProc_loadcell("",p1,stringfromlist(p1,fl,";"))

//analyse

checkbox check2, win=panel0, value=(os)

checkbox check0, win=panel0, value=(en)


ButtonProc_useit("")

CheckProc_1("",en)

checkscale1("")

//save

ButtonProc_saveCell("AUTO_cell"+num2str(p1+1))

endfor

//Result

ButtonProc_2editresults("")

End

```

```

function autoana1(lasch):buttoncontrol

string lasch

getmarquee /W=panel0#G0 left,top

if(!V_flag)

    return 0

endif

quantify("")

//ButtonProc_4lbamask("")

//ButtonProc_useit("werDAPI")

Checkscale1("")

```

```

        setdatafolder root:cells

string cl=replacestring(",","stringbykey("FOLDERS",datafolderdir(1),":",";") //data folders

variable    num=itemsinlist(cl,")+1

string      nn=("cell"+num2str(num))

        setdatafolder root:

        ButtonProc_saveCell("auto_"+nn)

        CheckProc_2chowAna("",1)

end

```

```

function checkScale1(lasch):buttoncontrol

string lasch

```

```

wave cell2 //threshold 3D

wave glu2 ///gabbr signal 3D

//wave dapi=:mask_dapi//dapi mask 3D

```

```

duplicate/O glu2, glu2o, glu2c//outside, cell

```

```

glu2o*=(cell2==0)

glu2c*=(cell2==1)

```

```

        //imagertransform/METH=1 zProjection glu2o

//wave M_zProjection

//      duplicate/O M_zProjection, co,co1

        duplicate/O glu2o,glu2or

wave co=glu2or

```

```

redimension/N=(numpts(co))/S co

co/=(co[p]!=0)

co/=(co[p]>=5)

make/O h_c,h_o

Histogram/B={0,1,255} co,h_o

variable mo=wavemax(h_o)

h_o/=mo

wavestats/Q co

variable/G v_extra=v_avg

variable/G v_thr=2*v_sdev

//print "outside:",v_extra,v_thr

//      imagetransform/METH=1 zProjection glu2c

//      duplicate/O M_zProjection, cc,cc1

      duplicate/O glu2c,glu2cr

wave cc=glu2cr

redimension/N=(numpts(cc))/S cc

cc/=(cc[p]!=0)

//      wavestats/Q cc

//      print "cell:",v_npnts,v_avg,v_sdev

      Histogram/B={0,1,255} cc,h_c

variable mc=wavemax(h_c)

h_c/=mc

wavestats/Q cc

variable/G v_cell=v_avg

```

```
variable/G v_PoT=V_sum*mc
```

```
//print v_cell,v_extra
```

```
setdrawlayer/K/W=panel0#g5 userfront
```

```
SetDrawEnv/W=panel0#g5 xcoord= bottom,ycoord= prel,linefgc= (0,0,0), linethick=2
```

```
drawline/W=panel0#g5 (V_extra+v_thr),0,(V_extra+v_thr),1
```

```
SetDrawEnv/W=panel0#g5 xcoord= bottom,ycoord= prel,linefgc= (40000,0,0), linethick=2
```

```
drawline/W=panel0#g5 (V_cell),0,(V_cell),1
```

```
removefromgraph/Z/W=panel0#g5 fit_h_o,fit_h_c
```

```
wave cellsmo
```

```
duplicate/O cellsmo,Cgfap
```

```
wave cg=Cgfap
```

```
redimension/N=(numpts(co))/5 cg
```

```
co/=(co[p]!=0)
```

```
wavestats/Q cg
```

```
variable/G v_gfap=v_avg
```

```
killwaves/Z cg
```

```
end
```

```
Function ButtonProc_saveCell(ctrlName) : ButtonControl
```

```
String ctrlName
```

```
if (!datafolderexists("Cells"))
```

```
newdatafolder/O root:Cells
```

```
endif
```

```
setdatafolder root:cells
```

```
string nn
```

```
variable num
```

```

        if(stringmatch(ctrlName,"auto*"))

            nn=stringfromlist(1,ctrlName,    "_")

            newdatafolder/O/S $(nn)

            num= str2num(ctrlName[strlen(ctrlName)-1])

        else

string cl=replacestring(", ",stringbykey("FOLDERS",datafolderdir(1),":",";"))    //data folders

            num=itemsinlist(cl, ";")+1

            nn=("cell"+num2str(num))

            prompt nn,"name of cell"

            doprompt "Name of stored cell", nn

            newdatafolder/O/S $(nn)


        endif


wave w_iba=root:w_iba

string/G file=note(w_iba)

wave cell=root:cell

string/G marq=note(cell)

wave gz=root:cellz

string/G crop

        if(waveexists(gz))

            crop=note(gz)

        endif

NVAR ithr=root:thr

variable/G ibathr=ithr

nvar ifil=root:smo

variable/G ibafilter=ifil

```



```
//results
```

```
nvar isize=root:res_size
```

```
variable/G ibasize=isize
```

```
NVAR mbas=root:v_extra
```

```
variable/G v_extra=mbas
```

```
nvar tbas=root:v_thr
```

```
variable/G v_thr=tbas
```

```
nvar mCell=root:v_cell
```

```
variable/G v_cell=mCell
```

```
nvar cg=root:v_gfap
```

```
variable/G v_gfap=cg
```

```
nvar pot=root:v_PoT
```

```
variable/G V_PoT=pot
```

```
nvar nucS=root:nuc_size
```

```
variable/G nuc_size=nucs
```

```
nvar nucE=root:dapi_ex
```

```
variable/G dapi_ex=nucE
```

```
wave curw=root:mask44
```

```
wave curwdapi=root:mask44dapi
```

```
duplicate/O curw, mask44
```

```
    if(waveexists(curwdapi))
```

```
duplicate/O curwdapi, mask44dapi
```

```
endif
```

```
setdatafolder root:
```

```
popupmenu popup0, win=panel0,mode=(num)
```

```
End
```

```
function quantify(lasch):ButtonControl
```

```
string lasch
```

```
checkbox check0, win=panel0, value=0
```

```
killwaves/Z mask44dapi,temp44dapi
```

```
getmarquee /W=panel0#G0 left,top
```

```
if(!V_flag)
```

```
return 0
```

```
endif
```

```
wave M_zProjection=:allcell
```

```
//display
```

```
duplicate/O/R=[v_left,v_right][v_top,v_bottom] M_zProjection, cell
```

```
Note cell, num2str(v_left)+";"+num2str(v_right)+";"+num2str(v_top)+";"+num2str(v_bottom)
```

```
//calculate
```

```
wave w_iba
```

```
duplicate/O/R=[v_left,v_right][v_top,v_bottom] w_iba, cell1, temp44, mask44
```

```
redimension/B/U cell1, temp44
```

```
mask44=1
```

```
wave w_vglut
```

```
//left IMG
```

```
duplicate/O/R=[v_left,v_right][v_top,v_bottom] w_vglut, glu1
```

```
wave w_dapi
```

```
duplicate/O/R=[v_left,v_right][v_top,v_bottom] w_dapi, celldapi
```

```
redimension/B/U celldapi
```

```
genMask(cell1)
```

```
NVAR res_size
```

```
wave cell2
```

```
wavestats/Q cell2
```

```
res_size=v_sum
```

```
//newimage/K=1/S=0 cell
```

```
end
```

```
function loadIMG(lasch):ButtonControl
```

```
string lasch
```

```
variable fref
```

```
open/R/D/T=".tif" fref
```

```
string fp= S_fileName
```

```
string/G path1=fp
```

```
ImageLoad/T=tiff/S=0/C=100/LR3D/G/N=Img fp
```

```
NVAR ibaChan
```

```
wave img
```

```
controlinfo/W=panel0 popup0
```

```
ibaChan=v_value-1
```

```
variable col
```

```

//      prompt col, "0=red; 1=green;2=blue"

//      doprompt "select channel color", col


if(stringmatch(lasch,"*1")) ///iba1

    //g2gray("img", col)

wave output

    duplicate/O img, W_iba

    note/K w_iba,fp


    imagetransform zProjection w_iba

wave M_zprojection

    duplicate/O          M_zprojection, allcell


    make/O/N=(dimsize(allcell,0),dimsize(allcell,1),3)/B/U cm1all


    makeRGBall(1)


endif


if(stringmatch(lasch,"*2")) ///vGlut


    //g2gray("img", col)

//wave output

    duplicate/O img, W_vGlut

    note/K W_vGlut,fp


    imagetransform zProjection W_vGlut

wave M_zprojection

```

```

        duplicate/O          M_zprojection, all_vglutA

        makeRGBall(0)

    endif

    if(stringmatch(lasch,"*3")) ///vGat

    //g2gray("img", col)

//wave output

    duplicate/O img, W_DAPI

    note/K W_DAPI,fp

    imagetransform zProjection W_DAPI

wave M_zprojection

    duplicate/O          M_zprojection, all_dapiA

    makeRGBall(2)

    endif

    killwaves/Z img

    //newimage/K=1 W_iba

end

Function ButtonProc_3(ctrlName) : ButtonControl

    String ctrlName

    NVAR rmin, rmax

    NVAR v_thr

```

NVAR v_extra

 rmin=round(v_extra+v_thr)

 rmax=rmin+5

 makeRGB(0)

End

Function CheckProc_1(ctrlName,checked) : CheckBoxControl

 String ctrlName

 Variable checked

 checkbox check0, win=panel0, value=checked

 if(checked)

wave mask44dapi

 if(!waveexists(mask44dapi))

 ButtonProc_useit("Button_dapi")

 else

 endif

endif

ButtonProc_useit("")

End

Function ButtonProc_projDAPI(ctrlName) : ButtonControl

 String ctrlName

 controlinfo/W=panel0 check44

 if(v_value)

 buttonproc_restore("laschDAPI")

 endif

```
fw_projDAPI()
```

```
End
```

```
Function PopMenuProc_loadcell(ctrlName,popNum,popStr) : PopupMenuControl
```

```
String ctrlName
```

```
Variable popNum
```

```
String popStr
```

```
popupmenu popup0,win=panel0, mode=popnum
```

```
checkbox check0, win=panel0, value=0
```

```
checkbox check2, win=panel0, value=0
```

```
SVAR marq=root:cells:$(popstr):marq
```

```
if(!SVAR_exists(marq))
```

```
return 0
```

```
endif
```

```
variable/G l1=str2num(stringfromlist(0,marq,";"))
```

```
variable/G r1=str2num(stringfromlist(1,marq,";"))
```

```
variable/G t1=str2num(stringfromlist(2,marq,";"))
```

```
variable/G b1=str2num(stringfromlist(3,marq,";"))
```

```
wave M_zProjection=:allcell //2D
```

```
//display
```

```
duplicate/O/R=[l1,r1][t1,b1] M_zProjection, cell
```

```
Note cell, num2str(l1)+";"+num2str(r1)+";"+num2str(t1)+";"+num2str(b1)
```

```
//calculate
```

```
wave w_iba //3D
```

```
duplicate/O/R=[l1,r1][t1,b1] w_iba, cell1,temp44
```

```

        redimension/B/U cell1, temp44

wave curw=root:cells:$(popstr):mask44

duplicate/O curw, root:mask44

        temp44*=curw

        cell1*=curw

        ButtonProc_restore("")

        duplicate/O/R=[l1,r1][t1,b1] w_iba, cell1

wave w_vglut

//left IMG

        duplicate/O/R=[l1,r1][t1,b1] w_vglut, glu1


wave w_dapi

        duplicate/O/R=[l1,r1][t1,b1] w_dapi, celldapi, all_dapi2m, tmp44,tmpOS

        tmp44=1

        tmpOS=1

        redimension/B/U celldapi, all_dapi2m

wave all_dapi2mp

wave curwD=root:cells:$(popstr):mask44dapi

        duplicate/O curwD, root:mask44dapi, root:mask_dapi

wave mask_dapi

        mask_dapi=(mask_dapi==0)


        wave mask44dapi

        if(waveexists(mask44dapi))

                all_dapi2m*=mask44dapi

        endif

NVAR smo

variable p1

        if(smo>0)

```



```

        for(p1=0;p1<smo;p1+=1)

            Imagefilter/N={3}/O gauss3d all_dapi2m

        endfor

    endif

    imagetransform zprojection all_dapi2m

    duplicate/O M_zProjection, all_dapi2mp

    duplicate/O all_dapi2m,all_DAPI2

    buttonproc_restore("laschDAPI")

    duplicate/O/R=[l1,r1][t1,b1] w_dapi, all_dapi2

    redimension/B/U all_dapi2


    genMask(temp44)

    checkscale("")

End

```

Function ButtonProc_2editresults(ctrlName) : ButtonControl

```

    String ctrlName


    make/O/T/N=(2,12) resC=""

wave/T res=root:resC

    res[0][0]="cell"

    res[0][1]="image"

    res[0][2]="pos[x,y]"

    res[0][3]="threshold/filter"

    res[0][4]="cell size (voxels)"

    res[0][5]="background avg signal"

    res[0][6]="background SD"

    res[0][7]="cell avg signal"

    res[0][8]="# voxels above threshold"

    res[0][9]="DAPI size"

    res[0][10]="DAPI excluded"

```

```

res[0][11]="GFAP signal"

setdatafolder root:cells

string c1=replacestring(" ",stringbykey("FOLDERS",datafolderdir(1),":",";")) //data folders

variable num=itemsinlist(c1, ";")

setdatafolder root:

//c1=sortlist(c1)

variable p1=0

for(p1=0;p1<num;p1+=1)

string f1=stringfromlist(p1,c1, ";")


insertpoints (dimsize(res,0)), 1, res

res[p1+2][0]=f1

svar n1=root:cells:$(f1):file

res[p1+2][1]=n1

svar n2=root:cells:$(f1):marq

res[p1+2][2]=n2

nvar v1=root:cells:$(f1):ibathr

nvar v2=root:cells:$(f1):ibafilter

res[p1+2][3]=num2str(v1)+"/"+num2str(v2)


nvar v3=root:cells:$(f1):ibasize

res[p1+2][4]=num2str(v3)


nvar v4a=root:cells:$(f1):v_extra

res[p1+2][5]=num2str(v4a) //signal background

nvar v4=root:cells:$(f1):v_thr

res[p1+2][6]=num2str(v4) //SD background

nvar v5=root:cells:$(f1):v_cell

res[p1+2][7]=num2str(v5) ///cell signal

```

```

nvar v6a=root:cells:$(f1):V_PoT

                res[p1+2][8]=num2str(v6a)    //voxels above thr

nvar v6=root:cells:$(f1):nuc_size

                res[p1+2][9]=num2str(v6)      //vGat attached

nvar v7=root:cells:$(f1):dapi_ex

                res[p1+2][10]=num2str(v7)     ///"vGat incorporated"

nvar v8=root:cells:$(f1):V_gfap

                res[p1+2][11]=num2str(v8)     ///"cGFAP signal

            endfor

        dowindow/K w_res

        edit/K=1/N=w_res res as " "

End

```