#pragma rtGlobals=1 // Use modern global access method.

function initIPSC() //creates table for storing data generated from function storeIPSC

make /o/n=0 IPSCpeak

make/o/t/n=0 wvname

Edit wvname, IPSCpeak

end

function storeIPSC(w, t) //measures IPSC amplitude

wave w //name of wave

variable t //time of stimulus

wave/t wvname

wave IPSCpeak

variable IPSC

wavestats/q/r = (t+2, t+10) w //edit values in parentheses to make sure peak min is captured

wavestats/q/r = (V\_minloc-0.2, V\_minLoc+.2) w

IPSC = V\_avg

addtoS(wvname, nameofwave(w))

addto(IPSCpeak, IPSC)

end

function initHoldingCurrent() //creates table for storing data generated from function holdingCurrent

make /o/n=0 HoldCurr

make/o/t/n=0 wvname

edit wvname, HoldCurr

end

function holdingCurrent(w) //measures holding current of cell

wave w //name of wave

variable current

variable holdCurr

wavestats/q/r = (20, 98) w

current = V\_avg

addtoS(wvname, nameofwave(w))

addto(HoldCurr, current)

end

function initParameters\_ALH() //creates table for storing data generated by function storeParameters\_ALH

make /o/n=0 RinWave, RsWave, CmWave

make/o/t/n=0 wName

make /o tempExp

make /o fit\_tempExp

display tempExp, fit\_tempExp

Edit wName, CmWave, RinWave, RsWave

end

function storeParameters\_ALH(w) //measures Rs, Rin, and Cm based on RC check

wave w //name of wave

NVAR Rin

NVAR Rs

NVAR Cm

wave/t wName

wave RinWave

wave RsWave

wave CmWave

parameters\_ALH(w)

addtoS(wName, nameofwave(w))

addto(RinWave, Rin)

addto(RsWave, Rs)

addto(CmWave, Cm)

end

function parameters\_ALH(w) //called by function storeParameters\_ALH

wave w

variable bl1 = 0

variable bl2 =100

variable amp=-5

wavestats/r=(700,1000) w

variable p1 = floor(V\_minloc/5)\*5

variable p2 = floor(V\_maxloc/5)\*5

variable bl = mean(w, bl1, bl2)

variable n = x2pnt(w, p2-p1)

make /o/n = (n) tempExp

SetScale/P x 0, deltax(w), "", tempExp

tempExp=w(x+p1)-bl;

wavestats /q tempExp

variable start

if (amp<0)

start=v\_minloc

else

start=v\_maxloc

endif

curveFit/q exp\_Xoffset tempExp(start, inf) /D

//print k0, k1, k2

variable /g Rs1 = 1000\*amp/(k0 + k1)

variable /g Rin1=1000\*amp/k0-Rs1

variable /g Cm=(1000\*k2)/Rs1

CurveFit/q dblexp\_XOffset tempExp(start, inf) /D

variable /g Rs = 1000\*amp/(k0 + k1 + k3)

variable /g Rin=1000\*amp/k0-Rs

end

function APfreq(w) //measures action potential frequency

wave w //name of wave

FindLevels/q w, -10

print (V\_LevelsFound/2)/.2

end

function sagRatio\_ALH(w, t1, t2) //measures sag ratio

wave w //name of wave

variable t1 //when pulse starts

variable t2 //when pulse stops

wavestats/q/r = (t1, t1+150) w

wavestats/q/r = (V\_minloc-1, V\_minLoc+4) w

make/o/n=1 steadyState

steadyState = V\_avg

wavestats/q/r = (t2, t2-100) w

wavestats/q/r = (V\_maxloc-6, V\_maxLoc) w

make/o/n=1 endOfPulse

endOfPulse = V\_avg

make/o/n=1 sagRatio

sagRatio=endOfPulse/steadyState

edit sagRatio

end

function initRiseTime() //creates table fo rstoring data generated by function riseTime

make/t/o/n=0 traceName

make/o/n=0 IPSCpeak

make/o/n=0 riseTimeStart

make /o/n=0 riseTimeAmp1

make/o/n=0 riseTimeStop

make /o/n=0 riseTimeAmp2

make/o/n=0 percRiseTime

make/o/n=0 Slope

edit traceName, IPSCpeak, riseTimeStart, riseTimeAmp1, riseTimeStop, riseTimeAmp2, percRiseTime, slope

end

function riseTime(w, t, p1, p2) //measures rise of wave, reports start time and amplitude, stop time and amplitude, percent rise time, and slope of rise

wave w //name of wave

variable t //time of pulse

variable p1, p2 //low end of percent range, high end of percent range i.e. 20, 80

wave/t traceName

wave IPSCpeak

wave riseTimeStart

wave riseTimeAmp1

wave riseTimeStop

wave riseTimeAmp2

wave percRiseTime

wave slope

addtos(traceName, nameofwave(w))

variable currentPeak

wavestats/q/r = (t, t+20) w

wavestats/q/r = (V\_minloc-0.5, V\_minloc+1.5) w

currentPeak = V\_avg

addto(IPSCpeak, currentPeak)

variable crossingValue1= currentPeak\*p1/100

variable time1

variable time2

addto(riseTimeAmp1, crossingValue1)

findlevel /q/r=(t,t+20) w, crossingValue1 //make sure peak amp is reached within 20 ms or this should be edited

addto(riseTimeStart, V\_levelx)

time1 = V\_levelx

variable crossingValue2= currentPeak\*p2/100

addto(riseTimeAmp2, crossingValue2)

findlevel /q/r=(t,t+20) w, crossingValue2 //make sure peak amp is reached within 20 ms or this should be edited

addto(riseTimeStop, V\_levelx)

time2 = V\_levelx

variable riseTimeTotal

riseTimeTotal = time2-time1

addto(percRiseTime, riseTimeTotal)

variable m

m = (crossingValue2-crossingValue1)/riseTimeTotal

addto(slope, m)

end

function initPPR() //generates table for storing data generated by function PPR\_ah

make/t/o/n=0 traceName

make/o/n=0 tpeak1

make/o/n=0 wpeak1

make/o/n=0 peak2

make/o/n=0 ppr

make/o/n=0 isi

edit traceName, tpeak1, wpeak1, peak2, ppr, isi

end

function PPR\_ah(w, t, p1, p2) //measures paired pulse ratios for two IPSCs

wave w //name of wave

wave t //name of template wave (single IPSC)

variable p1 //time of pulse 1

variable p2 //time of pulse 2

variable tpeak1amp

variable wpeak1amp

variable peak2amp

variable ratio

variable interstim

addtos(traceName, nameofwave(w))

interstim=p2-p1

addto(isi, interstim)

wavestats/q/r = (p1, p1+15) t

wavestats/q/r = (V\_minloc-0.25, V\_minLoc+.25) t

tpeak1amp = V\_avg

addto(tpeak1, tpeak1amp)

wavestats/q/r = (p1, p1+15) w

wavestats/q/r = (V\_minloc-0.25, V\_minLoc+.25) w

wpeak1amp = V\_avg

addto(wpeak1, wpeak1amp)

t=t\*(wpeak1amp/tpeak1amp)

w=w-t

wavestats/q/r = (p2, p2+15) w

wavestats/q/r = (V\_minloc-0.25, V\_minLoc+.25) w

peak2amp = V\_avg

addto(peak2, peak2amp)

ratio=peak2amp/wpeak1amp

addto(ppr, ratio)

end