

```

# Change this to your main directory in which all your other directories are
# setwd("")

# libraries
require(fields) # for image.plot

# Change plotting x limits and y limits
xlimits <- c(0.113, 0.14)
ylimits <- c(-.0000009, .0000009)
# set your colors
sinks.col <- "red"
source.col <- "black"
line.col <- "black"

# Change the window in which you want to calculate
# 1. peak sink value
# 2. sink area
xwin <- c(0.12, 0.13)

# Choose the first file in the folder you want to analyze
firstfile <- file.choose()

# Now run the rest of this
directory <- dirname(firstfile)

allfiles <- list.files(directory)
position <- as.numeric(unlist(strsplit(allfiles, ".", fixed=T)))[seq(1, length(allfiles)*2,
2)]
toanalyze <- data.frame("filename" = allfiles[order(position)], "position" =
position[order(position)])
todo <- 3:(length(allfiles)-2)

outfilename <- basename(directory)
outdirname <- dirname(directory)

# for outputting data:
outdata <- toanalyze[todo,]
outdata$peak <- outdata$sum <- NA

pdf(paste(outdirname, "/", outfilename, ".pdf", sep=""), width=4, height=8)

par(mfrow=c(19,1), plt=c(0,1,0.05,0.95))
for(f in todo){
  first <- read.csv(paste(directory, toanalyze$filename[f-2], sep="/"),
header=T, sep="\t", col.names=c("s", "mV"))
  second <- read.csv(paste(directory, toanalyze$filename[f], sep="/"),

```

```

        header=T, sep="\t", col.names=c("s", "mV"))
third <- read.csv(paste(directory, toanalyze$filename[f+2], sep="/"),
        header=T, sep="\t", col.names=c("s", "mV"))
new <- (first$mV + third$mV - 2*second$mV)/(20^2)
plot(second$s, new, type="l", main=toanalyze$position[f],
        xlim=xlimits, ylim=ylimits, bty="n", axes=F, ann=F, lwd=2, col=line.col)
xx <- c(second$s, rev(second$s))
yy <- c(rep(0, nrow(second)), rev(new))
polygon(xx, yy, col=sinks.col)
abline(v=xwin, col="gray")
xrange <- which(second$s > xwin[1] & second$s < xwin[2])
secs <- second$s[xrange]
# adding up only sinks (positive)
pos <- which(new[xrange]>0)
sink.max <- max(new[xrange][pos])
sink.sum <- sum(new[xrange][pos])
text(xwin[2], ylimits[2], adj=c(0, 1), cex=0.5, paste("peak=",
round(sink.max*1000000000, digits=2), sep=""))
text(xwin[2], ylimits[2], adj=c(0, 2.5), cex=0.5, paste("sum=",
round(sink.sum*1000000000, digits=2), sep=""))
text(0.13, 0.0000007, toanalyze$position[f], adj=c(1,1))
# write out calculated data
outdata$peak[outdata$position==toanalyze$position[f]] <-
sink.max*1000000000
outdata$sum[outdata$position==toanalyze$position[f]] <- sink.sum*1000000000
# write out raw source data
if(f==todo[1]){
    csds <- data.frame(s=second$s)
}
csds <- cbind(csds, new)
colnames(csds)[dim(csds)[2]] <- toanalyze$position[f]
}

```

```
dev.off()
```

```

outdatafilename <- paste(outdirname, "/", outfilename, "_", xwin[1], "-", xwin[2],
".csv", sep="")
write.csv(outdata, file=outdatafilename, row.names=F)
csddatafilename <- paste(outdirname, "/", outfilename, "_", "csds.csv", sep="")
write.csv(csds, file=csddatafilename, row.names=F)

```

```

# Now make pretty csd over space plot
# most of this is about getting good color ranges to match up nicely with data
# to make patterns visible.
# sinks (positive) are various shades of red

```

```

# sources (negative) are various shades of blue
#
s <- csds$s # seconds for trace
cplot <- as.matrix(csds[, 2:dim(csds)[2]]) # csd values
pos <- as.numeric(colnames(cplot)) # spatial position

# establish breakpoints for colors
brks <- c(min(cplot), seq(-3e-7, 3e-7, 1e-9), max(cplot))
ncols <- (length(brks)-1)/2
bluesramp <- colorRamp(c("blue", "white"), space="rgb")
lows <- rgb(bluesramp(seq(0,1,length.out = ncols))/255)
redsramp <- colorRamp(c("white", "red"), space="rgb")
highs <- rgb(redsramp(seq(0, 1, length.out=ncols))/255)
colorTable <- c(lows, highs) # 1 less than the breaks
colorTable[c(1, length(colorTable))] <- "#000000"
csdplotfilename <- paste(outdirname, "/", outfilename, "_", "csds.pdf", sep="")

pdf(csdplotfilename)
image.plot(s, pos, cplot, breaks=brks, col=colorTable, bigplot=c(0.15, 0.95, 0.15,
0.95), smallplot = c(1,1,1,1))
dev.off()

```