

Supplementary Material for Eglen et al Manuscript Review

Christophe Pouzat

November 6, 2013

Contents

1 Discretionary Revisions	1
1.1 Chunk <code>fourplot</code> leading to Fig. 2	1
1.2 Log ordinate scale for Fig. 4	3

1 Discretionary Revisions

1.1 Chunk `fourplot` leading to Fig. 2

The function `h5.read.spikes` of package `sjemea` is not documented neither is the class `mm.s` of the returned object `s`. Function `fourpot` calls function `plot.corr.index` that calls function `corr.do.fit` that ultimately generates the fitted line on the bottom right panel of Fig. 2. Since the linear fit is done on the log of the correlation indexes, which is the right way to proceed since, on the (ordinate) log scale, as shown by Fig. 3:

- The data fall essentially on a straight line;
- The data dispersion around the straight line is more uniform (*i.e.*, the variance is less heteroscedastic);

I would use an ordinate log scale for this bottom right panel (as done in Fig. 3). But since the authors gave me everything, I can do it! by redefining `fourplot` as follows:

```
fourplot2 <- function (s, names = FALSE, raster.time = NULL) {  
  old.par <- par(no.readonly = TRUE)  
  on.exit(par(old.par))
```

```

par(mfrow = c(2, 2), oma = c(0, 0, 2, 0), las = 1)
plot(s$layout, use.names = names)
plot.meanfiringrate(s, main = "")

  if (is.null(raster.time)) {
plot(s, main = "", label.cells = names,
  use.names = names)}
  else {
plot(s, main = "", label.cells = names,
  use.names = names, beg = raster.time[1],
  end = raster.time[2])}

  if (any(names(s) == "corr")) {
plot.corr.index2(s, main = "", log="y")}

  mtext(basenamepy(s$file)$base, side = 3, outer = T)
}

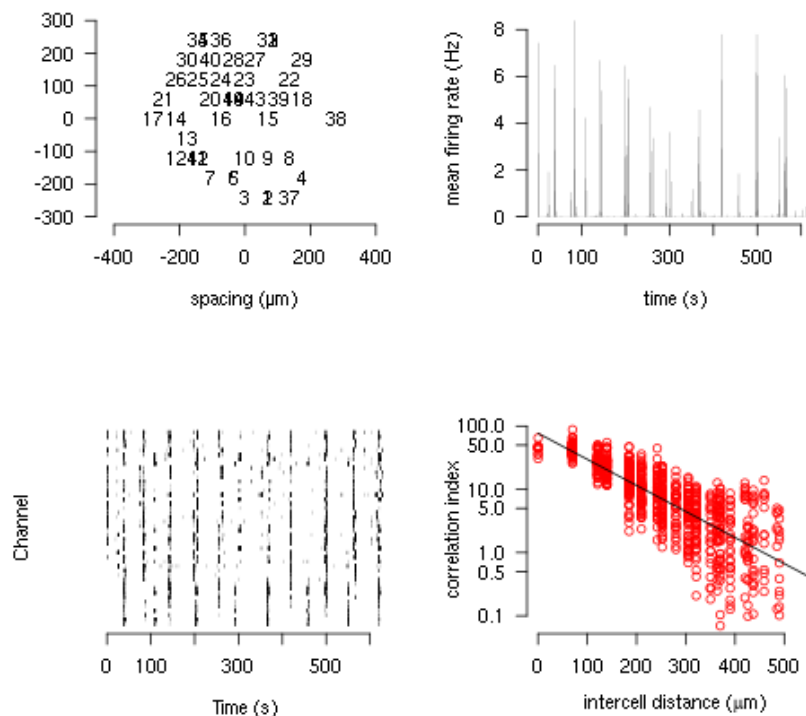
```

Giving an alternative version of the `fourplot` code chunk:

```

file = paste0(hdf5.root, "Wong1993/Wong1993_P1.h5")
s = h5.read.spikes(file)
fourplot2(s)

```



1.2 Log ordinate scale for Fig. 4

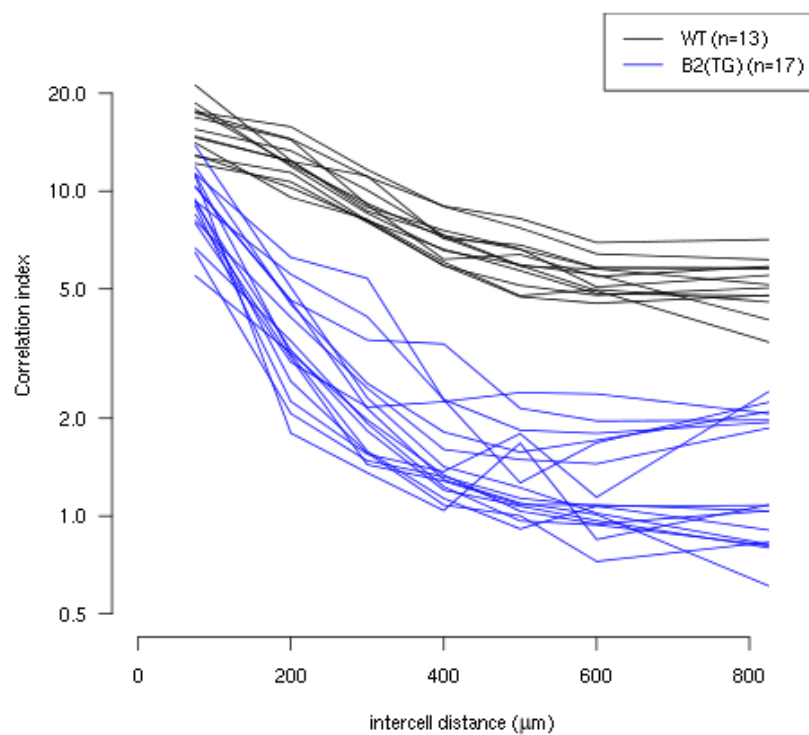
Exploiting further the privilege given to me by the authors, I can modify their Fig. 4 using a log scale on the ordinate:

```
plot(c(0, 850),c(0.5, 30),
     xlim=c(0, 850), ylim=c(0.5, 30),
     bty='n',las=1,
     xlab = expression(paste("intercell distance (", mu, "m)")),
     ylab='Correlation index',
     log="y",type="n")
n <- length(corr.means)
n.wt <- 0
for (i in 1:n) {
  dat <- corr.means[[i]]$means
```

```

col <- ifelse(corr.means[[i]]$cond=='wt', 'black', 'blue')
if (corr.means[[i]]$cond=='wt') n.wt <- n.wt + 1
lines(dat[, "mid"], dat[, "mean"], col=col)
legend('topright', c(sprintf('WT (n=%d)', n.wt),
  sprintf('B2(TG) (n=%d)', n-n.wt)),
  lty=1, col=c('black', 'blue'))

```



Showing that the "rescued" group (B2(TG)) exhibits a shorter space constant as well as a more complex distance dependence than the wild type.