Table of contents

Contents

Description

Reproducing data analysis and figures reported in "Low intensity TMS enhances perception of visual stimuli" by Abrahamyan, Clifford, Arabzadeh and Harris (2015) published in Brain Stimulation. You can accesss single-subject data and reproduce the analysis by downloading files from github.com/armanabraham. If you use RStudio, load the downloaded project and reproduce the analysis by running Data_analysis_and_figures.Rmd file.

Subject data are provided in RData (Data_on_Github.RData) and Excel (Data_On_Github.xslx) formats.

```
library(plyr)
library(ggplot2)
library(ggthemes)
library(gridExtra)
library(effsize) # for cohen.d
library(pastecs) # For stat.desc
library(afex)
library(nlme) # For linear mixed-effects modelling, lme
library(BayesFactor) # Bayes factor analaysis of different experimental designs
## Load Winston Chang's functions to compute within-subjects error bars
source('WithinSbj_StdErrors.R')
# Load Exp 1 and 2 data
load("Data_On_Github.RData")
```

Data

Show how data are formatted for the analysis

```
head(exp1_dataMean, 4)
```

```
##   PPT Condition         Th Sensitivity
## 1 S01       100 0.14608210    7.956563
## 2 S01        60 0.16881954    6.003824
## 3 S01        70 0.09309656   11.156462
## 4 S01        80 0.12383785    8.291708
```

Data analysis

Quadratic trend in data Here we test if visual sensitivity changes as a function of TMS stimulation intensity when subjects discriminated gratings oriented ±45 deg. Due to repeated-measures design, we conducted mixed-modelling analysis wherein subject was a random effect and TMS stimulation intensity was fixed effect. We fitted linear, cubic and quadratic models to the data and found that quadratic fit explains data better than linear or cubic fits ($\chi^2(1)$=5.22, p=0.022).

```
################## --- Data analysis --- ###################
# Order Conditions for plotting convenience
exp1_dataMean$Condition <- factor(exp1_dataMean$Condition,
                                  levels=c("NoTMS", "ip80", "ip90", "60", "70", "80", "90", "100"))
# Find out if there is a quadratic trend in the data as a function of TMS intensity
# Select only experimental conditions (exclude NoTMS and ipsilatoral stimulation)
exp1_quadraticFitData <- droplevels(subset(exp1_dataMean, !(Condition %in% c("ip80", "ip90", "NoTMS")))
exp1_quadraticFitData <- within(exp1_quadraticFitData, {CondAsNum <- as.numeric(as.character(Condition)
baseline <- lme(data=exp1_quadraticFitData, Sensitivity~1, random=~1|PPT/CondAsNum, method="ML")
linear <- update(baseline, .~. + CondAsNum)
quadratic <- update(baseline, .~. + CondAsNum + I(CondAsNum^2))
cubic <- update(baseline, .~. + CondAsNum + I(CondAsNum^2)+ I(CondAsNum^3))
```

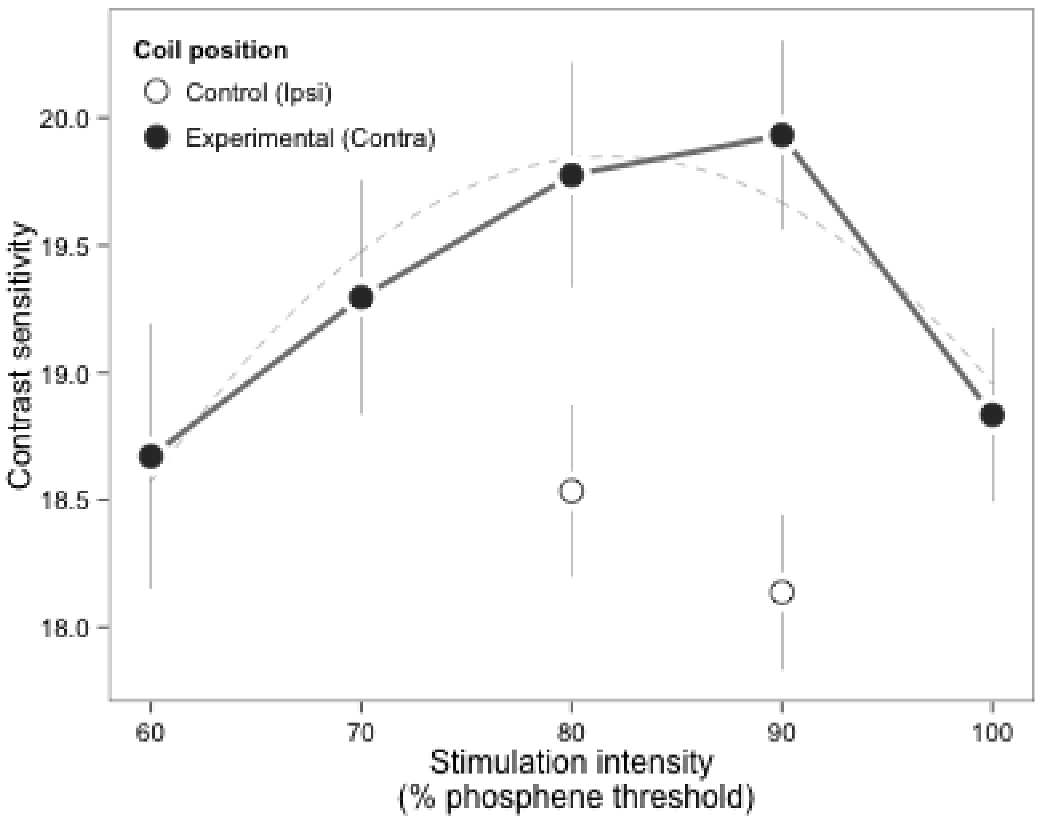
```
anova(baseline, linear, quadratic, cubic)
```

```
##           Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## baseline      1  4 248.5679 256.5973 -120.2840
## linear        2  5 250.1203 260.1570 -120.0602 1 vs 2 0.447629  0.5035
## quadratic     3  6 246.9039 258.9479 -117.4519 2 vs 3 5.216463  0.0224
## cubic         4  7 248.2253 262.2766 -117.1126 3 vs 4 0.678586  0.4101
```

Sensitivity change as a function of TMS stimulation (Figure 1) This corresponds to Figure 1 in the manuscript. The dashed line shows quadratic trend in the data.

```
# Compute high-res quadratic trend for the plot
fitX <- data.frame(CondAsNum=seq(60,100, 0.1))
predVals <- data.frame(x=fitX$CondAsNum, y=predict(quadratic, fitX, level=0))
# Plot sensitivity from Exp 1 together with quadratic trend
# Summarise data for plotting
exp1_dataForPlot <- droplevels(subset(exp1_dataMean, Condition!="NoTMS"))
exp1_dataForPlot$CoilPosition <- "Expt"
exp1_dataForPlot$CoilPosition[exp1_dataForPlot$Condition %in% c("ip80", "ip90")] <- "Cntr"
exp1_dataForPlot$CoilPosition <- as.factor(exp1_dataForPlot$CoilPosition)
## Change the name of the column called "Condition" into "Intensity" cause
## it is more intuitive
colnames(exp1_dataForPlot)[which(names(exp1_dataForPlot) == "Condition")] <- "Intensity"
## Now change ip80 and ip90 into 80 and 90
exp1_dataForPlot$Intensity <- mapvalues(exp1_dataForPlot$Intensity,
                                        from = c("ip80", "ip90"), to = c("80", "90"))
## Change order of columns (not sure if this is needed, but looks clearer)
exp1_dataForPlot <- droplevels(exp1_dataForPlot[,c("PPT", "CoilPosition",
                                                   "Intensity", "Th", "Sensitivity")])
# Compute means and within-subjects std errors
exp1_qFitForPlot <- summarySEwithin(exp1_dataForPlot,
                                    measurevar="Sensitivity",
                                    withinvars=c("Intensity", "CoilPosition"),
                                    idvar="PPT")
gExp1_SensTrend <- ggplot(exp1_qFitForPlot, aes(x=as.numeric(as.character(Intensity)),
                                                y=Sensitivity, color=CoilPosition)) +
  theme_few() +
  theme(legend.position=c(0.19,0.87)) +
  xlab("Stimulation intensity\n (% phosphene threshold)") +
```

```
  ylab("Contrast sensitivity") +
  geom_line(data=predVals, aes(x=x, y=y), linetype="dashed", colour="grey70", size=.3) +
  geom_line(subset=.(CoilPosition=="Expt"), aes(group=1), color="grey50", size=1) +
  geom_errorbar(aes(ymin = Sensitivity - se, ymax = Sensitivity + se),
                width = 0.00,
                size = 0.2,
                color="grey50") +   # Change "se" to "ci" to plot 95% conf intervals
  geom_point(size=7, color="white") +
  geom_point(aes(shape=CoilPosition), size=4, color="grey20") +
  scale_shape_manual(values=c(1,19), name="Coil position",
                     labels=c("Control (Ipsi)","Experimental (Contra)"))
print(gExp1_SensTrend)
```



Enhancement in visual sensitivity To confirm enhancement in visual sensitivity, here we compare experimental (contralateral) stimulation with control (ipsilateral) stimulation. Because ipsilateral control condition was only tested at two TMS intensities (80 and 90% of PT), we performed a 2x2 repeated measures ANOVA comparing the location of the TMS coil (experimental vs control) and stimulation intensities (80 and 90%).

```
# Compute 2x2 ANOVA comparing 80% and 90% ipsi and contra conditions
exp1_data8090 <- droplevels(subset(exp1_dataForPlot,  Intensity %in% c("80", "90")))
aov8090 <- aov.car(data=exp1_data8090,
                   Sensitivity~CoilPosition*Intensity+Error(PPT/(CoilPosition*Intensity)),
                   return="Anova", args.return=list(es="pes"))
```

```
## Warning in stri_c(..., sep = sep, collapse = collapse, ignore_null = TRUE):
```

```
## longer object length is not a multiple of shorter object length
```

```r
nice.anova(aov8090, es="pes")
```

```
##                   Effect    df  MSE         F  pes     p
## 1           CoilPosition 1, 10 0.76 33.45 ***  .77 .0002
## 2              Intensity 1, 10 1.67      0.10 .009   .76
## 3 CoilPosition:Intensity 1, 10 1.67      0.50  .05   .49
```

This analysis confirmed that there was a significant main effect for the location of the TMS coil (F(1,10)=33.45, p<0.001, partial η^2=0.77). However, the main effect of stimulation intensity was not significant, indicating no difference between 80 and 90% intensities (F(1,10)=0.10, p=0.76, partial η^2=0.009). Further, there was no interaction between TMS location and stimulation intensity (F(1,10)=0.50, p=0.49, partial η^2=0.05).

We followed up with paired t-tests, which showed significant differences between experimental and control conditions at both 80% and 90% stimulation (80%: *Mdiff* =1.24, t(10)=2.65, p=0.02, Cohen's d=0.27; 90%: *Mdiff* =1.80, t(10)=3.80, p=0.003, Cohen's d=0.47).

80% contra vs ipsi

```r
# t-tests
# Compare 80% experimental and control conditions for sensitivity
exp1_data80 <- droplevels(subset(exp1_data8090, Intensity=="80"))
exp1_ttest80 <- t.test(Sensitivity~CoilPosition, data=exp1_data80, paired=TRUE)
# Compute effect size r using formula from DSUR (sqrt(t^2/(t^2+df)))
paste('r =',
      sqrt(exp1_ttest80$statistic[[1]]^2/(exp1_ttest80$statistic[[1]]^2+exp1_ttest80$parameter[[1]])))
```

```
## [1] "r = 0.641549923796786"
```

```r
# Cohen's d is another estimate of effect size.
cohen.d(data=exp1_data80, Sensitivity~CoilPosition)
```

```
##
## Cohen's d
##
## d estimate: -0.268937 (small)
## 95 percent confidence interval:
##        inf        sup
## -1.2064354  0.6685615
```

90% contra vs ipsi

```r
# 90% experimental vs control
exp1_data90 <- droplevels(subset(exp1_data8090, Intensity=="90"))
exp1_ttest90 <- t.test(Sensitivity~CoilPosition, data=exp1_data90, paired=TRUE)
print(exp1_ttest90)
```

```
##
##  Paired t-test
##
## data:  Sensitivity by CoilPosition
```

```
## t = -3.8084, df = 10, p-value = 0.003438
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.8447441 -0.7446994
## sample estimates:
## mean of the differences
##               -1.794722
```

```
paste('r =',
      sqrt(exp1_ttest90$statistic[[1]]^2/(exp1_ttest90$statistic[[1]]^2+exp1_ttest90$parameter[[1]])))
```

```
## [1] "r = 0.769350361865236"
```

```
cohen.d(data=exp1_data90, Sensitivity~CoilPosition)
```

```
##
## Cohen's d
##
## d estimate: -0.472718 (small)
## 95 percent confidence interval:
##        inf        sup
## -1.4198125  0.4743765
```

Control (ipsi) vs no-TMS

```
## Compare ipsi and no-TMS for
exp1_ipsiNoTMS <- droplevels(subset(exp1_dataMean, Condition %in% c("ip80", "ip90", "NoTMS")))
exp1_ipsiNoTMSSens <- cast(exp1_ipsiNoTMS, PPT~Condition, value=.(Sensitivity))
exp1_ipsiNoTMSSens <- ddply(exp1_ipsiNoTMSSens, .(PPT, NoTMS, ip80, ip90), mutate, ipMean=(ip80+ip90)/2)
exp1_ttestNoTMS <- t.test(exp1_ipsiNoTMSSens$NoTMS, exp1_ipsiNoTMSSens$ipMean, paired=TRUE)
print(exp1_ttestNoTMS)
```

```
##
##  Paired t-test
##
## data:  exp1_ipsiNoTMSSens$NoTMS and exp1_ipsiNoTMSSens$ipMean
## t = 2.6245, df = 10, p-value = 0.0254
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1650503 2.0208311
## sample estimates:
## mean of the differences
##                1.092941
```

```
# Compute effect size r
paste('r =',
      sqrt(exp1_ttestNoTMS$statistic[[1]]^2/(exp1_ttestNoTMS$statistic[[1]]^2+exp1_ttestNoTMS$parameter
```

```
## [1] "r = 0.638638128620297"
```

```r
# Cohen's d
cohen.d(exp1_ipsiNoTMSSens$NoTMS, exp1_ipsiNoTMSSens$ipMean)
```

```
##
## Cohen's d
##
## d estimate: 0.2624841 (small)
## 95 percent confidence interval:
##        inf        sup
## -0.6747955  1.1997637
```

Best TMS condition (90% of PT) vs no-TMS

```r
exp1_NoTMSAnd90 <- droplevels(subset(exp1_dataMean, Condition %in% c("90", "NoTMS")))
exp1_ttestNoTMSvs90 <- t.test(data=exp1_NoTMSAnd90, Sensitivity~Condition, paired=TRUE)
print(exp1_ttestNoTMSvs90)
```

```
##
##  Paired t-test
##
## data:  Sensitivity by Condition
## t = -0.82135, df = 10, p-value = 0.4306
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.8679120  0.8617083
## sample estimates:
## mean of the differences
##              -0.5031019
```

```r
# Effect size as r
paste('r =',
      sqrt(exp1_ttestNoTMSvs90$statistic[[1]]^2/(exp1_ttestNoTMSvs90$statistic[[1]]^2+exp1_ttestNoTMSvs
```

```
## [1] "r = 0.251391125531333"
```

```r
# Cohen's d
exp1_NoTMSAnd90Wide <- cast(exp1_NoTMSAnd90, PPT~Condition, value='Sensitivity')
cohen.d(exp1_NoTMSAnd90Wide$`90`, exp1_NoTMSAnd90Wide$NoTMS)
```

```
##
## Cohen's d
##
## d estimate: 0.1276009 (negligible)
## 95 percent confidence interval:
##       inf       sup
## -0.806314  1.061516
```