Supplemental Material

R Source Code for Simulations

```
#### Required Packages ####
library(plyr)
library(MASS)
library(reshape2)
library(data.table)
library(doMC)
registerDoMC()

#### responderAnalysis() ####
# responderAnalysis - Main Function
#   Arguments:
#     params  - vector of mu and sigma values
#     n       - number of 'subjects' per 'experiment'
#     reps    - number of times to replicate the experiment
#   Returns:
#     sim.v.proof
responderAnalysis <- function(params, n, reps) {

  d = .798                  # point-biserial correction factor
  t.crit <- qt(.95, n-2)    # one-sided t-test critical value

  c.s <- params[5] # variance
  c.g <- params[8] # sessionXreliability cross covariance
  c.r <- params[6] # reliability covarance
  c.t <- params[7] # within session or task covariance

  SIGMA <- matrix(
    c(c.s, c.r, c.t, c.g,
      c.r, c.s, c.g, c.t,
      c.t, c.g, c.s, c.r,
      c.g, c.t, c.r, c.s),
    nrow=4,
    ncol=4,
    byrow=TRUE
  )

  mu <- params[1:4]

  # Run replications
  data <- replicate(reps, mvrnorm(n, mu, SIGMA))

  # Transform array to data.table
  data <- adply(data, 3, )
  data <- as.data.table(data)
  setnames(data, c("rep","x1","x2","y1","y2"))
  setkey(data, rep)
  print(data)

  # Calculate change scores
  data[,dx:=x2-x1]
  data[,dy:=y2-y1]

  # Assign responder group
  data[,med:=median(dx),by=rep]
  data[,is.responder:=dx>median(dx),by=rep]

  # Calculate correlations
  rs <- data.table(rep=1:reps)
  setkey(rs,rep)
  rs[,r.dx.dy:=data[,cor(dx,dy),by=rep][,V1]]
  rs[,r.dxD.dy:=data[,cor(is.responder,dy),by=rep][,V1]]
```

```
  # Calculate t-stats, f-stats, and whether p<.05
  rs[,t.value:=(r.dxD.dy*sqrt(n-2))/sqrt(1-r.dxD.dy^2)]
  rs[,p.value:=t.value>t.crit]
  rs[,f.value:=t.value^2]

  # Calculate median point-biserial correlation
  out <- data.table(r.dxD.dy=rs[,median(r.dxD.dy)])

  # Calculate median Pearson's r correlation
  # b/t 'training' and 'transfer' task change scores
  out[,r.dx.dy:=median(rs[,r.dx.dy])]

  # Calculate median t-test
  # of beta for predicting transfer change score
  # from responder group
  out[,t.val:=median(rs[,t.value])]
  # Calculate P(p<.05) for simulated
  out[,P.p.05:=mean(rs[,p.value])]
  out[,f.val:=median(rs[,f.value])]


 return(out)
}

#### Run responderAnalysis() ####

# 50 subjects per experiment
n <- 50
# Repeat each experiment 1000 times
reps <- 100000

  # Values for covariance matrix
  c.s <- c(1) # s^2
  c.r <- c(.6) # reliability
  c.t <- c(.3) # within session
  c.g <- c(.15) # sessionXreliability

  # Create matrix of all covariance matrix parameter combinations
  all.sigmas <- expand.grid(list(c.s=c.s,c.r=c.r,c.t=c.t,c.g=c.g))

  # Mean vectors
  all.mus <- c(1,5,1,2,
               1,4,1,1.1,
               1,3,1,1.05,
               1,2,1,1.025,
               1,1,2,1.005)

  # Create combination of all mean vectors and covariance matrices
  mu.length <- length(all.mus)/4
  all.mus <- matrix(rep((all.mus), times=dim(all.sigmas)[1]), ncol=4, byrow=T)
  all.sigmas <- apply(all.sigmas, 2, function(x) rep(x, each=mu.length))
  all.params <- cbind(all.mus,all.sigmas)
  all.params <- matrix(as.vector(all.params), ncol=8, byrow=F)

  # Run the simulations
  sim.data <- adply(.data=all.params, .margins=1, .parallel=TRUE, .progress='text',
responderAnalysis ,n, reps)

  params <- all.params[1,]


  #### Create Summary Table ####
  names(sim.data)[1] <- "rep"
  sim.data <- data.table(sim.data)
  setkey(sim.data,rep)

  params <- as.data.table(all.params)
  setnames(params, c("mu.x1","mu.x2","mu.y1","mu.y2","r.v","r.rel","r.val","r.cross"))
  params[,rep:=1:dim(params)[1]]
  setkey(params,rep)
```

```
data <- params[sim.data]

# Output Summary Table
data
```