

# Wilcox Power Analysis with Mixed Distributions

wflenoir

## Power Analysis of Wilcox rank sum test with two populations

Code was influenced by <https://www.statmethods.net/stats/power.html> (<https://www.statmethods.net/stats/power.html>) and <https://cran.r-project.org/web/packages/wmwpow/wmwpow.pdf> (<https://cran.r-project.org/web/packages/wmwpow/wmwpow.pdf>)

This power analysis was conducted to help aid an experimental biologist that required power analysis in order to compare measurements of two populations of cells. The first cell population is assumed to be wild-type and normally distributed. The second cell population is expected to be a bimodal distribution made up of  $t$  cells identical to the first cell population and  $1-t$  expected to be a second normal distribution. This second cell population is suppose to be a mixed distribution of wild-type and gene-edited ( $1-t$ ) cells, as the edited cells cannot be filtered from the wild-type cells.

Traditionally, t-tests are used for mean comparisons between two distributions, however given that the second cell population is assumed bimodal and not normal, this violates typical t-test normality assumptions. This suggests that we have to use non-parametric mean comparison methods (Wilcox rank sum) in order to see if there is any difference between the two samples.

With cohen's d, effect size  $d = (u1 - u2)/s$ . For the rest of the analysis we will operate under the assumption that  $sd_1 = sd_2$  for both distributions (wild-type & edited) in the bimodal sample, and that  $u1 = 0$ , and  $u1 = 1$ . Given that magnitude/scale does not influence Wilcox rank sum tests, we will not worry about the scale of numbers, but rather just test different effect sizes (how much does the biomodal distribution mix/separate) in the second population.

To put the analysis into better perspective, let's visualize the second cell population (bimodal) with the following: - editing efficiency = 50% -cohen's  $d = 0.5$  (for sample 2),  $d = \text{abs}(u1 - u2/sd) = \text{abs}(0-1/sd)$ .  $sd = 1/d$  -cells in each sample ( $n$ ) = 10,000

```

n <- 10000
eff <- 0.50
d <- 0.5
sd <- 1 / d

y1 <- rnorm(n, 0, sd) ### normal distribution around 0
y2 <- rnorm(n, 1, sd) ### normal distribution around 1
w <-
  rbinom(n, 1, eff) ### number of successes given an efficiency, picks approx eff
dist1 <-
  rnorm(n, 0, sd) ### cell population 1, wild-type, same distribution as y1
dist2 <-
  c(y1[w == 0], y2[w == 1]) ### cell population 2, ko, mixed distribution of y1 & y2

### visualizing knockout cell population
wt_sub <- y1[w == 0]
ko_sub <- y2[w == 1]

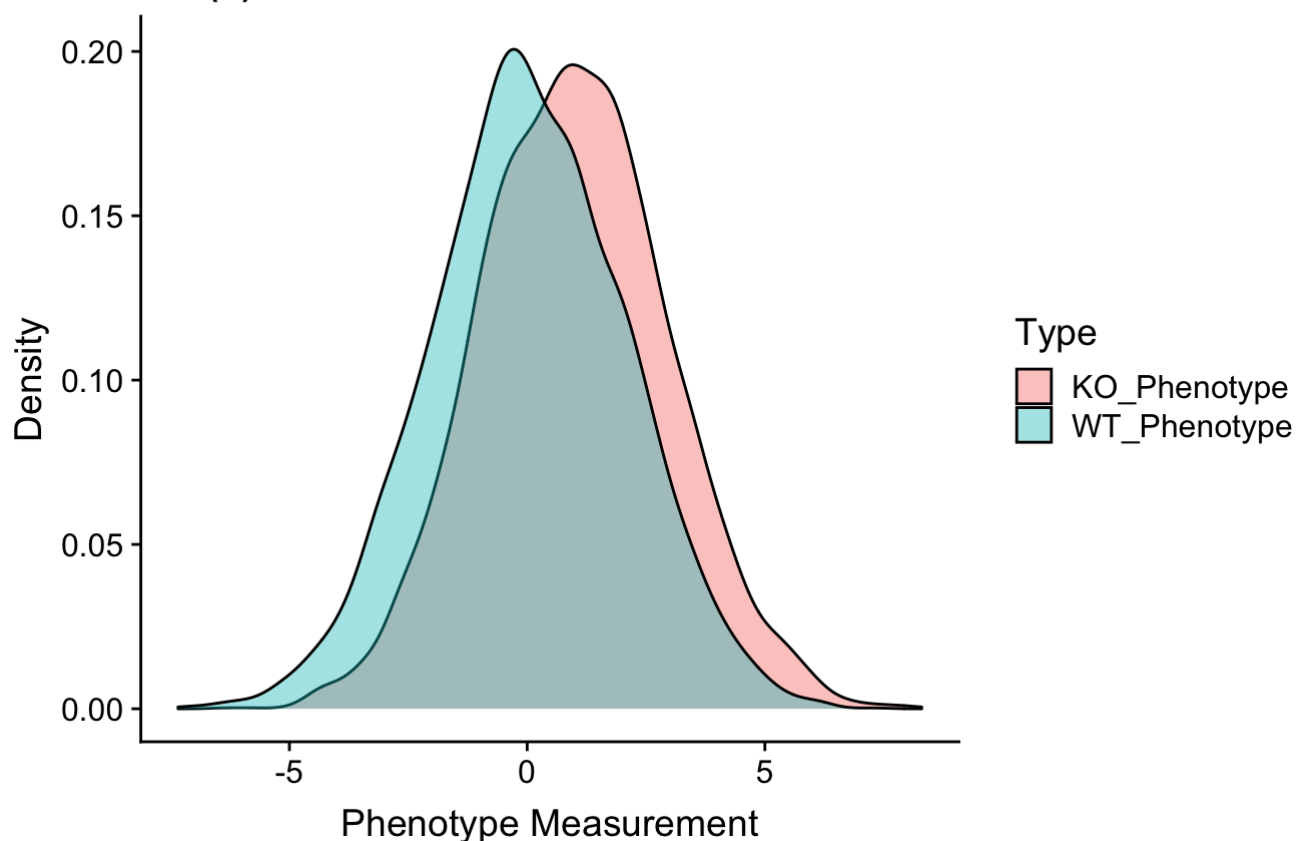
temp <- data.frame(wt_sub)
temp2 <- data.frame(ko_sub)
colnames(temp)[1] <- "Meas"
colnames(temp2)[1] <- "Meas"
temp$Type <- "WT_Phenotype"
temp2$Type <- "KO_Phenotype"
temp <- rbind(temp, temp2)
title <-
  paste(
    "Separated Knock-Out Cell Population Comparison \nCells(n) = ",
    n,
    " Edit Eff = ",
    eff,
    " Effect Size = ",
    d
  )

# pdf("./Density_Knockout_Population.pdf",
#       height = 10,
#       width = 12)
temp %>% ggplot(aes(x = Meas, fill = Type)) +
  geom_density(alpha = 0.4) +
  xlab("Phenotype Measurement") +
  ylab("Density") +
  ggtitle(title)

```

## Separated Knock-Out Cell Population Comparison

Cells(n) = 10000 Edit Eff = 0.5 Effect Size = 0.5



```
#dev.off()

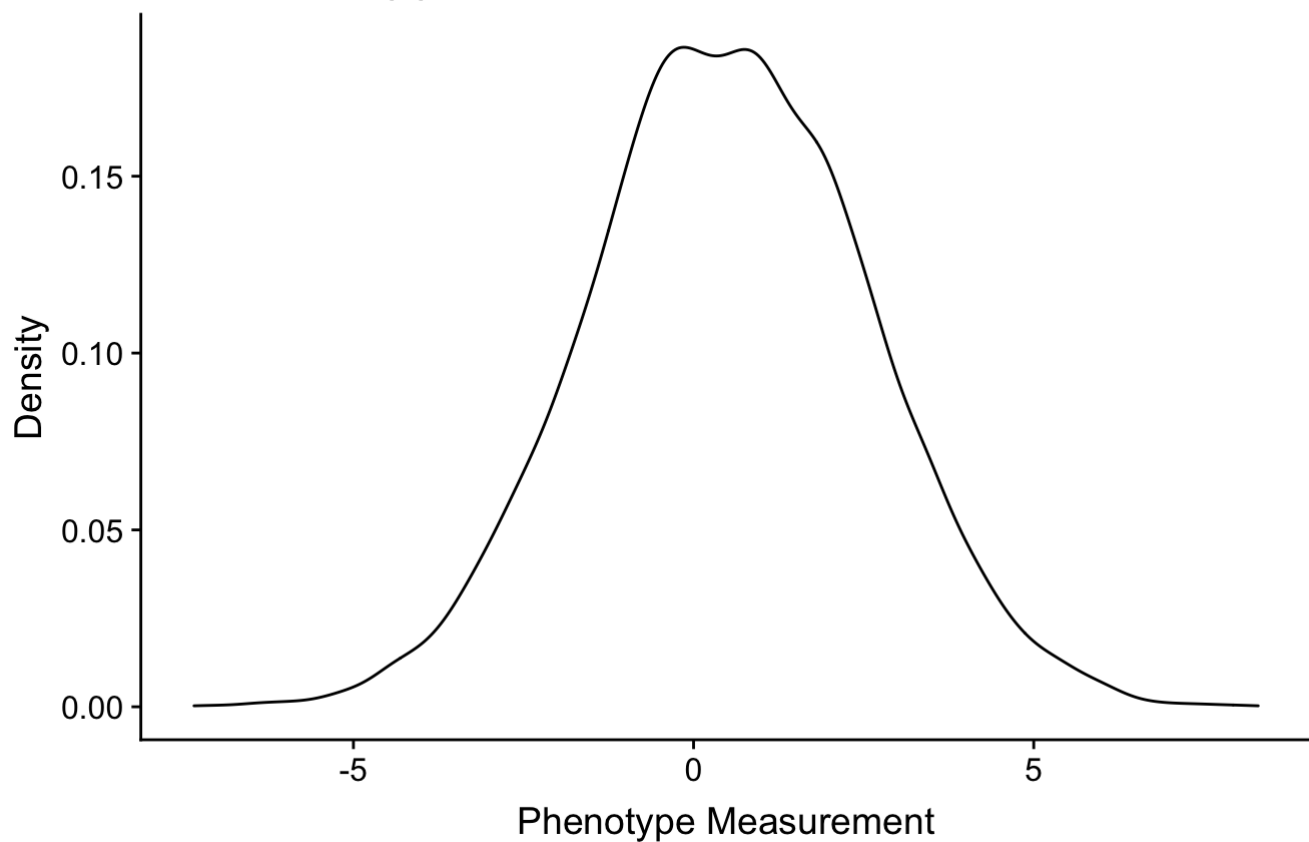
#pdf("./Density_Knockout_Population_Combined.pdf",
#    height = 10,
#    width = 10)

title <-
  paste(
    "Combined Knock-Out Cell Population Comparison \nCells(n) = ",
    n,
    " Edit Eff = ",
    eff,
    " Effect Size = ",
    d
  )

temp %>% ggplot(aes(x = Meas)) +
  geom_density(alpha = 0.4) +
  xlab("Phenotype Measurement") +
  ylab("Density") +
  ggtitle(title)
```

## Combined Knock-Out Cell Population Comparison

Cells(n) = 10000 Edit Eff = 0.5 Effect Size = 0.5



```
#dev.off()
```

Combining the two distributions does not indicate much of a difference, however let's visualize what the actual comparison of the two cell sample populations:

```

### visualizing wilcox comparison

dat_WT <- data.frame(dist1)
dat_KO <- data.frame(dist2)
dat_WT$Type <- "Wild-Type"
dat_KO$Type <- "Knock-Out"
colnames(dat_WT)[1] <- "Meas"
colnames(dat_KO)[1] <- "Meas"
dat <- rbind(dat_WT, dat_KO)

title <-
  paste("Cell Population Comparison \nCells(n) = ",
        n,
        " Edit Eff = ",
        eff,
        " Effect Size = ",
        d)
# pdf("./Violin_Wilcox_Comparison.pdf",
#      height = 10,
#      width = 12)
dat %>% ggplot(aes(x = Type, y = Meas, fill = Type)) +
  geom_violin(alpha = 0.4) +
  stat_compare_means(method = "wilcox") +
  ylab("Phenotype Measurement") +
  xlab("") +
  ggtitle(title)

```



```
#dev.off()
```

Wilcox rank sum test indicates that the population differences are very far apart overall, even though it is not easily observable in the graphic. Let's write a function to simulate through various other editing efficiencies and effect sizes that could be observed in the second cell population:

```
###simulation function, feed in the efficiency, number of samples, and standard deviation calculated from effect size
wilcox_sim <- function(eff, n, sd) {
  y1 <- rnorm(n, 0, sd) ### normal distribution around 0
  y2 <- rnorm(n, 1, sd) ### normal distribution around 1
  w <-
    rbinom(n, 1, eff) ### select randomly from population based on cutting efficiency
  dist1 <- rnorm(n, 0, sd) ### cell population 1 (wild-type)
  dist2 <-
    c(y1[w == 0], y2[w == 1]) ### cell population 2 (KO), mixed distribution
  dist1 <-
    round(dist1, 5) ### rounding the values due to computation processing
  dist2 <- round(dist2, 5)
  ### getting the p-value from the wilcox test
  p <-
    wilcox.test(
      dist1,
      dist2,
      paired = FALSE,
      correct = FALSE,
      alternative = "two.sided",
      exact = FALSE
    )$p.value
  return(p) ### return p-value
}
```

Now let's simulate various editing efficiencies and effect sizes, while fixing the rest of the potential variables. This simulation will run 1000 times for each editing efficiency and effect size, and will compute an empirical power by estimating the probability that the two cell populations are significantly ( $p < 0.05$ ) different.

```

edit_effs <- c(.20, .50, .80, .90) ### tested editing efficiencies
effect_size <-
  seq(0.05, 1.0, 0.05) ### tested effect sizes of bimodal distribution from KO populatio
n

### empty vectors, used to store tested efficiencies (effs),
### effect sizes (ds), empirical power (emp_ps),
### and overlap percentages - overlapping coefficient (overlap)

effs <- c()
ds <- c()
emp_ps <- c()
overlap <- c()
n <- 10000

for (eff in edit_effs) {
  ### for loop through efficiencies
  for (d in effect_size) {
    ### for loop through effect sizes
    sd <- 1 / d

    ### This is an alternative for loop, if replicate is too computationally expensive to
run
    # pvals <- c()
    # for (i in 1:1000){
    #   print(paste(eff,d,i))
    #   p <- wilcox_sim(eff,n,sd)
    #   pvals <- c(pvals,p)
    # }

    print(paste("editing eff",eff," effect size",d))
    ### only running once for example purposes
    pvals <- replicate(1, wilcox_sim(eff, n, sd))

    ### ran on a server 1000 times.
    #pvals <- replicate(1000,wilcox_sim(eff,n,sd))

    emp_power <- round(sum(pvals < 0.05) / length(pvals), 3)
    ### calculating emp_power
    ### power = probability to reject H0 given H1

    ### overlap coefficient, percentage of overlapping populations (alternative way to t
hink of effect size)
    ### identified from: https://rpsychologist.com/calculating-the-overlap-of-two-normal-distributions-using-monte-carlo-integration
    ovl <- 2 * pnorm(-abs(d) / 2)

    ### storing the simulations
    overlap <- c(overlap, ovl)
    effs <- c(effs, eff)
    ds <- c(ds, d)
    emp_ps <- c(emp_ps, emp_power)
  }
}

```

```
}  
}
```



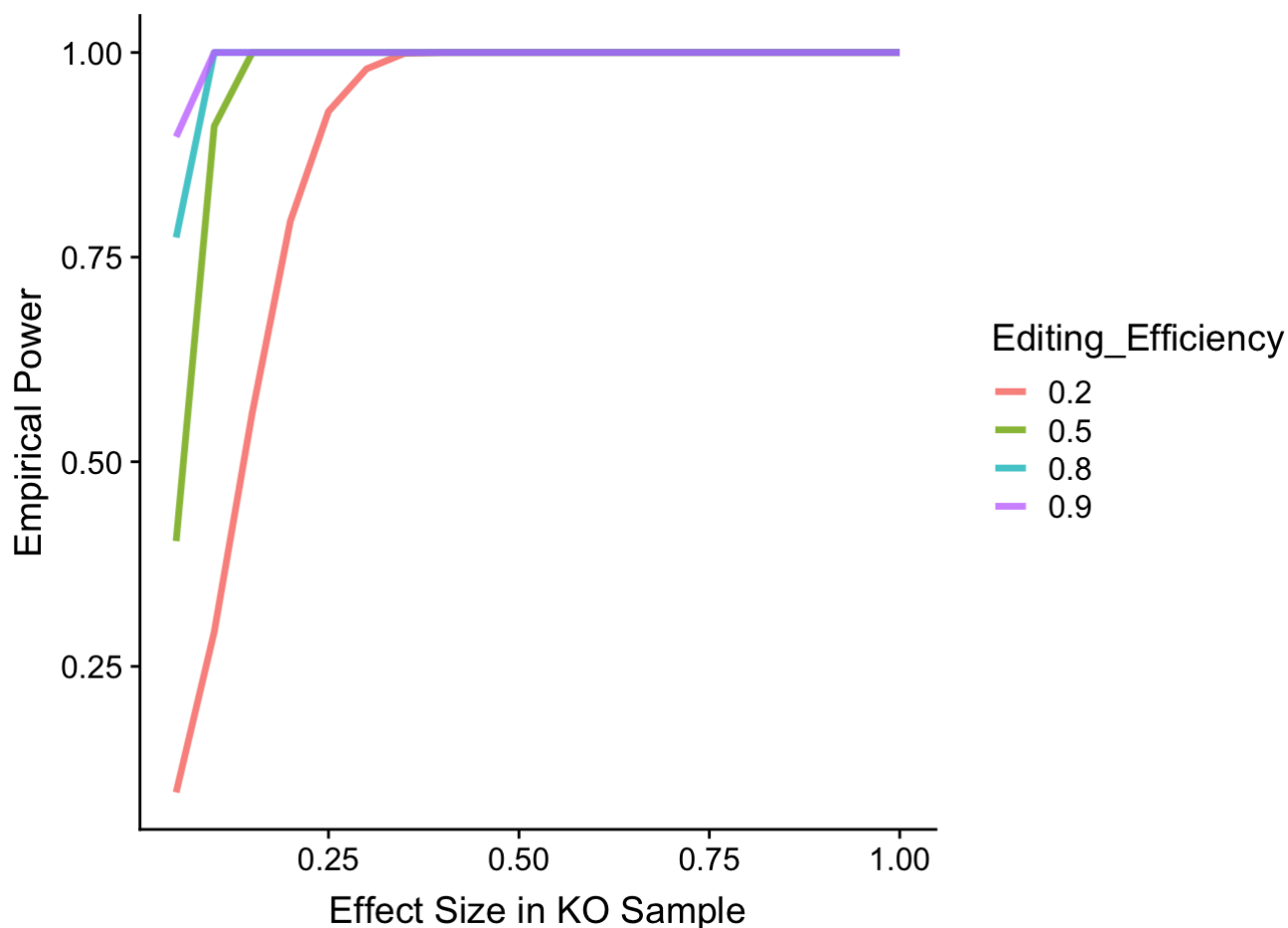
```
## [1] "editing eff 0.2 effect size 0.05"
## [1] "editing eff 0.2 effect size 0.1"
## [1] "editing eff 0.2 effect size 0.15"
## [1] "editing eff 0.2 effect size 0.2"
## [1] "editing eff 0.2 effect size 0.25"
## [1] "editing eff 0.2 effect size 0.3"
## [1] "editing eff 0.2 effect size 0.35"
## [1] "editing eff 0.2 effect size 0.4"
## [1] "editing eff 0.2 effect size 0.45"
## [1] "editing eff 0.2 effect size 0.5"
## [1] "editing eff 0.2 effect size 0.55"
## [1] "editing eff 0.2 effect size 0.6"
## [1] "editing eff 0.2 effect size 0.65"
## [1] "editing eff 0.2 effect size 0.7"
## [1] "editing eff 0.2 effect size 0.75"
## [1] "editing eff 0.2 effect size 0.8"
## [1] "editing eff 0.2 effect size 0.85"
## [1] "editing eff 0.2 effect size 0.9"
## [1] "editing eff 0.2 effect size 0.95"
## [1] "editing eff 0.2 effect size 1"
## [1] "editing eff 0.5 effect size 0.05"
## [1] "editing eff 0.5 effect size 0.1"
## [1] "editing eff 0.5 effect size 0.15"
## [1] "editing eff 0.5 effect size 0.2"
## [1] "editing eff 0.5 effect size 0.25"
## [1] "editing eff 0.5 effect size 0.3"
## [1] "editing eff 0.5 effect size 0.35"
## [1] "editing eff 0.5 effect size 0.4"
## [1] "editing eff 0.5 effect size 0.45"
## [1] "editing eff 0.5 effect size 0.5"
## [1] "editing eff 0.5 effect size 0.55"
## [1] "editing eff 0.5 effect size 0.6"
## [1] "editing eff 0.5 effect size 0.65"
## [1] "editing eff 0.5 effect size 0.7"
## [1] "editing eff 0.5 effect size 0.75"
## [1] "editing eff 0.5 effect size 0.8"
## [1] "editing eff 0.5 effect size 0.85"
## [1] "editing eff 0.5 effect size 0.9"
## [1] "editing eff 0.5 effect size 0.95"
## [1] "editing eff 0.5 effect size 1"
## [1] "editing eff 0.8 effect size 0.05"
## [1] "editing eff 0.8 effect size 0.1"
## [1] "editing eff 0.8 effect size 0.15"
## [1] "editing eff 0.8 effect size 0.2"
## [1] "editing eff 0.8 effect size 0.25"
## [1] "editing eff 0.8 effect size 0.3"
## [1] "editing eff 0.8 effect size 0.35"
## [1] "editing eff 0.8 effect size 0.4"
## [1] "editing eff 0.8 effect size 0.45"
## [1] "editing eff 0.8 effect size 0.5"
## [1] "editing eff 0.8 effect size 0.55"
## [1] "editing eff 0.8 effect size 0.6"
## [1] "editing eff 0.8 effect size 0.65"
```

```
## [1] "editing eff 0.8 effect size 0.7"
## [1] "editing eff 0.8 effect size 0.75"
## [1] "editing eff 0.8 effect size 0.8"
## [1] "editing eff 0.8 effect size 0.85"
## [1] "editing eff 0.8 effect size 0.9"
## [1] "editing eff 0.8 effect size 0.95"
## [1] "editing eff 0.8 effect size 1"
## [1] "editing eff 0.9 effect size 0.05"
## [1] "editing eff 0.9 effect size 0.1"
## [1] "editing eff 0.9 effect size 0.15"
## [1] "editing eff 0.9 effect size 0.2"
## [1] "editing eff 0.9 effect size 0.25"
## [1] "editing eff 0.9 effect size 0.3"
## [1] "editing eff 0.9 effect size 0.35"
## [1] "editing eff 0.9 effect size 0.4"
## [1] "editing eff 0.9 effect size 0.45"
## [1] "editing eff 0.9 effect size 0.5"
## [1] "editing eff 0.9 effect size 0.55"
## [1] "editing eff 0.9 effect size 0.6"
## [1] "editing eff 0.9 effect size 0.65"
## [1] "editing eff 0.9 effect size 0.7"
## [1] "editing eff 0.9 effect size 0.75"
## [1] "editing eff 0.9 effect size 0.8"
## [1] "editing eff 0.9 effect size 0.85"
## [1] "editing eff 0.9 effect size 0.9"
## [1] "editing eff 0.9 effect size 0.95"
## [1] "editing eff 0.9 effect size 1"
```

```
### Saving the simulations into a data frame
#results <- data.frame(effs,ds,emp_ps,overlap)
#write_delim(results,"./wilcox_mix_dist_simulations.txt")
```

For display purposes, the above code is only running one iteration instead of 1000.

```
power_data <- read.csv("./wilcox_mix_dist_simulations.txt", sep = "")
power_data$Editing_Efficiency <- as.character(power_data$effs) ###Setting as a character
for plot
#pdf("./Power_Wilcox_Mix_Dist.pdf",height = 10,width = 12)
power_data %>% ggplot(aes(x = ds, y = emp_ps, color = Editing_Efficiency)) +
  geom_line(alpha = 0.8,size = 1.2) +
  xlab("Effect Size in KO Sample") +
  ylab("Empirical Power")
```



```
#dev.off()
```

As we can see, we get pretty good power when achieving an effect size above 0.3. Similarly, improving the editing efficiency improves power of the provided test.

This analysis suggests that if we are unable to measure a specific phenotype of a knockout by itself, we can still make some inference on whether or not a gene knockout, or at least altered sample, is measurably different from wild-type behavior.