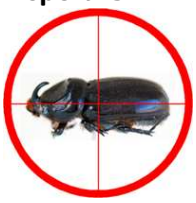
Guam Coconut Rhinoceros Beetle Project

Technical Report CRB-2014-07-29



Funnel Test

Prepared by
Aubrey Moore and Roland Quitugua
University of Guam Cooperative Extension Service

Updated August 20, 2014

In an attempt to minimize the escape rate for pan traps with chicken wire tops, we placed funnels beneath the chicken wire. During a four week trial, pan traps equipped with funnels trapped 35% more beetles than those without funnels. During the experiment, trap catch rate for pan traps equipped with funnels was 16X that of surrounding standard pheromone traps within a 5 km radius.

1 Introduction

We observed that CRB can escape from pans with chicken wire tops. In this experiment we test funnels to see if they increase trap catch.

2 Methods

We designed funnels to fit in pans and had some manufactured out of PVC sheet by a local company, Express Signs. The funnels were tested over 4 one-week trapping periods by placing them on the 31 pan traps in operation at the Yigo Ag. Expt. Stn. Each pan trap was equipped with a chicken wire top, UVLED and orycatalure. Each pan was supported by an empty 55 gal. oil drum. No breeding material substrate was placed inside the barrel.

Funnels were added to approximately half of the pan traps. To decide which traps would be equipped with funnels, we ranked the traps by trap-catch within the month prior to the experiment,

in descending order. On week one and three of the experiment, pans with odd numbered ranks were fitted with funnels. On week two and four, the funnels were moved to pans with even numbered ranks.

Data from the trial were stored in the **YigoBarrelObs** table of the project's **oryctes** database. Analysis was done using a couple of R scripts (Listings 1 and 2).

3 Results

Pan traps equipped with funnels caught 35% more beetles than those without funnels (Table 1, Figure 1). The increase in trap catch rate, from 0.233 to 0.313 beetles per trap-day, is significant ($p = 0.0013$, Fisher's Exact Test for Count Data, two-sided). Trap catch rate for 51 standard pheromone traps with a 5 km radius of the field trial area during the period of the trial was 0.020. During the four week trial period, 3 out of 31 (9%) of pan traps caught no beetles. In contrast, 41 out of 55 (75%) of standard traps caught no beetles.

4 Discussion

There is an operational disadvantage in using funnels in panel traps in that the funnel needs to be removed to check for trapped beetles. This extra time needed for trap checking is offset in a several ways:

- significantly higher trap catch
- trapped beetles are shaded from the sun and remain healthy for several days making them available for experiments
- lures can be protected from direct sunshine by hanging them beneath the cone

Table 1: Beetles caught in pan traps with and without funnels.

barrelID	WithoutFunnel	WithFunnel
1	2	3
2	1	3
3	5	6
4	3	1
5	4	6
6	1	4
7	0	4
8	1	10
9	3	6
10	0	0
11	4	2
12	3	3
13	2	0
14	7	5
15	5	8
16	4	2
17	2	1
18	3	4
19	2	3
20	1	1
21	0	4
22	4	7
23	0	0
24	4	5
25	3	5
26	0	11
27	0	0
28	14	8
29	9	10
30	7	10
31	7	4

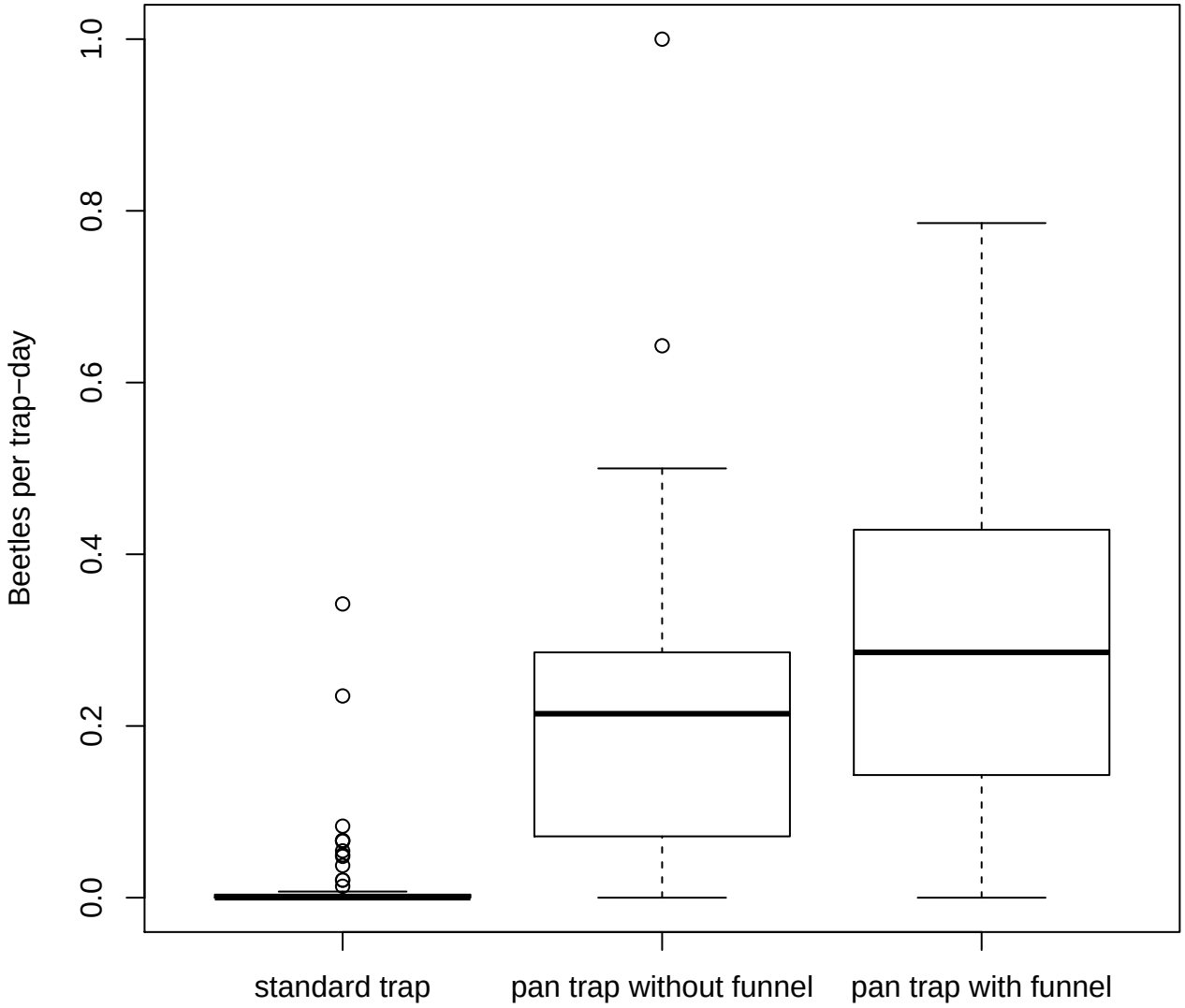


Figure 1: Pan traps equipped with funnels trapped significantly more CRB than those without funnels (increase = 34%; p = 0.0013, Fisher's Exact Test for Count Data, two-sided). Trap catch rate for 52 standard pheromone traps with a 5 km radius of the field trial area during the period of the trial is provided for comparison. Mean trap catch rates were 0.020, 0.233 and 0.313 for standard pheromone traps, pan traps without funnels, and pan traps with funnels, respectively.

Listing 1: db.R

```
# Start the clock! Takes about 75 seconds to run.
ptm <- proc.time()

require(RMySQL)
require(xtable)

# Connect to project database.
# DB username and password are hidden in an exernal file, dbsecrets.R,
# which defines two strings: user and password.
# dbsecrets.R should not be made public so that security of the DB is
# protected.

source("dbsecrets.R")

conn = dbConnect(MySQL(),
                 user = user,
                 password = password,
                 host = "mysql.guaminsects.net",
                 dbname = "oryctes")

# The standardTraps.R script gets trap catch data for standard pheromone traps
    within 5 km of the
# centroid of the Yigo Ag Expt Stn over the trial period, 2014-06-15 through
    2014-07-13.

source("standardTraps.R")

# Get trap data for pans not equipped with funnels.

sql = ("
SELECT barrelID, SUM(males+females) AS trapCatch
FROM YigoBarrelObs
WHERE startdate >= '2014-06-15'
  AND endDate <= '2014-07-13'
  AND funnel = 'n'
GROUP BY barrelID
ORDER BY barrelID
")
rs = dbSendQuery(conn, sql)
withoutFunnel = fetch(rs, -1)

# Get trap data for pans equipped with funnels.

sql = ("
SELECT barrelID, SUM(males+females) AS trapCatch
FROM YigoBarrelObs
WHERE startdate >= '2014-06-15'
  AND endDate <= '2014-07-13'
  AND funnel = 'y'
GROUP BY barrelID
ORDER BY barrelID
")
rs = dbSendQuery(conn, sql)
```

```
withFunnel = fetch(rs, -1)

# Create and export trapCatchTable.tex

df = data.frame(withoutFunnel, withFunnel$trapCatch)
names(df) = c('barrelID', 'WithoutFunnel', 'WithFunnel')

xt = xtable(df,
            display=c('d','d','d', 'd'),
            caption='Beetles caught in pan traps with and without funnels.',
            label='trapCatchTableLabel')
print(xt,
      caption.placement='top',
      include.rownames=FALSE,
      table.placement = "p",
      file='trapCatchTable.tex'
      )

# Test significance of difference in trap catch

dm = data.matrix(df[2:3])
fisher.test(dm, workspace=2e8)

# Create and export Boxplot

# Results from 51 standard traps within 5 km radius displayed for comparison

sql = "SELECT meanTrapRate AS tr
  FROM standardTraps
  GROUP BY trapID;"
standardTrap = sqldf(sql)

pdf('boxplot.pdf')
boxplot(standardTrap$tr, df$WithoutFunnel/14.0, df$WithFunnel/14.0,
        names = c('standard trap', 'pan trap without funnel', 'pan trap with funnel
            '),
        ylab = 'Beetles per trap-day')
dev.off()

mstandard = mean(standardTraps$meanTrapRate)
mwith = mean(df$WithFunnel)/14.0
mwithout = mean(df$WithoutFunnel)/14.0

mwith/mwithout

mwithout/mstandard
mwith/mstandard

emptyStandardTraps = sqldf("SELECT COUNT(*) FROM standardTraps WHERE meanTrapRate =
    0.0;")
emptyPans = sqldf("SELECT COUNT(*) FROM df WHERE withoutFunnel + withFunnel = 0.0;")


# Stop the clock
```

```
proc.time() - ptm

print("FINIS")
```

Listing 2: standardTraps.R

```
##### Calculate trap rate for standard pheromone traps within a
##### 5 km radius of the centroid of the Yigo Ag Expt Stn.

# This code is imported using source().

# Running this code takes several minutes.

library(RMySQL)
require(geosphere)
require(sqldf)
options(sqldf.driver = "SQLite") # as per FAQ #7 force SQLite

getNearestTraps = function(longitude, latitude, delta=0.05, n=5) {
  # Gets the trap_id for n traps nearest to the lat-long point.
  # Returns a dataframe containing trap_id and distance to point in meters.

  # Start by finding all traps within a bounding rectangle which is plus
  # and minus delta degrees from the point. This step reduces processing time.
  # We don't want to calculated distance to point for all traps, just those
  # nearby.

  sql = paste(
    "SELECT trap_id, longitude, latitude FROM trap",
    "WHERE ISNULL(date_removed)",
    " AND longitude >", longitude - delta,
    " AND longitude < ", longitude + delta,
    " AND latitude > ", latitude - delta,
    " AND latitude < ", latitude + delta)
  #  conn = odbcConnect("oryctes")
  conn = dbConnect(MySQL(),
                   user = "readonlyguest",
                   password = "readonlypassword",
                   host = "mysql.guaminsects.net",
                   dbname = "oryctes")
  #  result = sqlQuery(conn, sql)
  #  odbcClose(conn)
  rs = dbSendQuery(conn, sql)
  result = fetch(rs, -1)
  print(result)
  if (nrow(result) < n){
    print(
      paste(
        "WARNING: There are less than n traps within the bounding box.",
        "Reduce n or increase delta."
      )
    )
  }

  # Now that we have a list of nearby traps, calculate distance to the point
```

```
  # using the distm function from th geosphere library and select the n
  # nearest points.

  result$dist = distm(result[,2:3], c(longitude, latitude))
  sql = paste(
    "SELECT * FROM result",
    "ORDER BY dist",
    "LIMIT 0,", n
  )
  sql
  result = sqldf(sql)
}

# TEST
# Get nearest traps to Tai Cliff barrel trap
#(nearestTraps = getNearestTraps(144.785008, 13.427339))

##### End of getNearestTraps function definition.

(nearestTraps = getNearestTraps(144.872720, 13.531372, delta=0.5, n=58))

################################################################################

# Create tempTrapCatch table

# tempTrapCatch table will containing the fields:
# trapID, latitude, longitude, date2, date1, trapCatch, trapRate

# Note that this query uses all records in the trap_visit table.
# The intent is to add a new field, date1, which is the date on
# which the trap was previously visited. Thus the trap period is
# date1 through date2, or DATEDIFF(date1, date2) days.

dbSendQuery(conn, "DROP TABLE IF EXISTS tempTrapCatch;")

sql = ("
       CREATE TABLE tempTrapCatch
       SELECT trap_visit.trap_id AS trapID, latitude, longitude, visit_date as date2
           ,
       (
       SELECT visit_date
       FROM trap_visit
       WHERE trap_id = trapID
       AND visit_date < date2
       ORDER BY visit_date DESC
       LIMIT 1
       ) AS date1,
       male_count + female_count + unsexed_count AS trapCatch
       FROM trap_visit, trap
       WHERE trap_visit.trap_id = trap.trap_id;
       ")
dbSendQuery(conn, sql)

# If date1 is null, this indicates a first trap visit.
```

```
# Replace date1 with trap.date_deployed.

sql = ("
       UPDATE tempTrapCatch, trap
       SET date1 = date_deployed
       WHERE trapID = trap_id
       AND date1 IS NULL;
       ")
dbSendQuery(conn, sql)

# Delete any records where either date is NULL or date1 >= date2.

sql = ("
       DELETE FROM tempTrapCatch
       WHERE date1 IS NULL
       OR date2 IS NULL
       OR date1 >= date2;
       ")
dbSendQuery(conn, sql)

# Add trapRate field

dbSendQuery(conn, "ALTER TABLE `tempTrapCatch` ADD `trapRate` DOUBLE NOT NULL
    COMMENT 'beetles per trap day';")

# Calculate trapRate

dbSendQuery(conn, "UPDATE tempTrapCatch SET trapRate = trapCatch / DATEDIFF(date2,
    date1);")

##############################################################################

# Create tempTrap table.

# This table contains a single field, trapID, which is simply a list of traps
# of interest. In this case, all traps within a 5 km radius of the centroid of the
# Yigo Expt Stn.

dbSendQuery(conn, "DROP TABLE IF EXISTS tempTrap");
dbSendQuery(conn, "CREATE TABLE tempTrap (trapID VARCHAR(255), PRIMARY KEY(trapID))
    ;")

sql = "INSERT INTO tempTrap (trapID) VALUES"
for (i in 1:nrow(nearestTraps)){
  sql = paste(sql, "('", nearestTraps$trap_id[i], "')", sep='')
  if (i < nrow(nearestTraps)){
    sql = paste(sql, ",", sep='')
  }else{
    sql = paste(sql, ";", sep='')
  }
}
dbSendQuery(conn, sql)
```

```
##############################################################################################


# Create tempDaily table.
# Fields: trapID, tdate, trapRate

dbSendQuery(conn, "DROP TABLE IF EXISTS tempDaily;")

dbSendQuery(conn, "CREATE TABLE tempDaily (trapID VARCHAR( 255 ), tdate DATE,
    trapRate DOUBLE);")

mydate = seq(as.Date("2014-06-15"), as.Date("2014-07-13"), "days")
for (i in 1:length(mydate)){
  curdate = mydate[i]
  sql = paste("
    INSERT INTO tempDaily
    SELECT tempTrap.trapID, '", curdate, "' AS tdate, trapRate
    FROM tempTrapCatch, tempTrap
    WHERE
      tempTrapCatch.trapID = tempTrap.trapID
      AND '", curdate, "' >= date1
      AND '", curdate, "' <= date2;",
              sep="")
  dbSendQuery(conn, sql)
}

##############################################################################################


# Calculate trap rate for each trap over the trapping period
sql = ("SELECT trapID, MIN(tdate) AS minDate, MAX(tdate) AS maxDate, AVG(trapRate)
    AS meanTrapRate
        FROM tempDaily
        GROUP BY trapID
        ORDER BY trapID;")
rs = dbSendQuery(conn, sql)
standardTraps = fetch(rs, -1)
```