**Supplementary material 1** A custom python script to conduct a randomization test for within-house sourcing of insects after spraying in El Carrizal. For all bugs collected post-spray from houses infested pre-spray, determines whether nearest pre-spray relative is from same house and keeps a tally (observed), and calculated distance between houses for those not from the same house and records. Compares that to 1000 replicates of house of randomly-picked pre-spray insect. Provides 95% and 99% threshold proportions.

[shebang] /usr/bin/env python

import sys

import random

import math

'''

Sara Helms Cahan

12-22-17

House\_locations.txt:

House\_No Lat Long

D1 14.372582 -89.983922

D2 14.372544 -89.984161

D2A 14.373409 -89.984711

D3 14.373464 -89.985347

D4 14.373420 -89.985364

D5 14.373752 -89.985150

housefile.txt:

ID1 House1 Sample1 ID2 House2 Sample2 k

CHJ356 D31 CAR1 CHJ357 D31 CAR1 0.283980759

CHJ356 D31 CAR1 CHJ358 D36 CAR1 0.013353742

CHJ356 D31 CAR1 CHJ360 D63 CAR1 0.015974383

CHJ356 D31 CAR1 CHJ361 D58 CAR1 0.021099686

CHJ356 D31 CAR1 CHJ362 D58 CAR1 0.043890698

CHJ356 D31 CAR1 CHJ363 D57 CAR1 0.034970817

CHJ356 D31 CAR1 CHJ364 D95 CAR1 0.002408586

CHJ356 D31 CAR1 CHJ365 D93 CAR1 0

CHJ356 D31 CAR1 CHJ366 D93 CAR1 0

CHJ356 D31 CAR1 CHJ367 D93 CAR1 0

Focal post-spray houses:

D17

D4

D58

D80

D113

D39

D108

D7

D39A

D103

D18A

'''

LocFileName = sys.argv[1] #House\_locations.txt

LocFile = open(LocFileName, 'r')

LocDict = {}

LineCounter = 0

for line in LocFile:

if LineCounter == 0:

LineCounter += 1

else:

line=line.strip().split()

LocDict[line[0]]= [float(line[1]),float(line[2])] #makes a location dictionary for each house

LocFile.close()

InfoFileName = sys.argv[2] #housefile.txt

InfoFile = open(InfoFileName, 'r')

FocalHouse = {}

PreHouse = {}

Prek = {}

FocalList = ["D17","D4","D58","D80","D113","D39","D108","D7","D39A","D103","D18A"] #only considers post-spray houses that had 1+ insect

LineCounter = 0

for line in InfoFile: #builds dictionaries for finding nearest-relative relationships for each post-spray individual sample collection house of sample, collection house of each pre-spray sample, k between them)

if LineCounter == 0:

LineCounter += 1

else:

line=line.strip().split("\t")

if line[4] in FocalList and line[5] != "CAR1" and line[2] == "CAR1":

#print "found one"

if line[3] not in FocalHouse.keys():

FocalHouse[line[3]] = [line[4]]

PreHouse[line[3]]= [line[1]]

Prek[line[3]] = [float(line[6])]

else:

PreHouse[line[3]].append(line[1])

Prek[line[3]].append(float(line[6]))

LineCounter += 1

DiffFileName = "Observed\_distances\_CAR.tsv" #this is the name of the output file of observed nearest-relative identifications.

DiffFile = open(DiffFileName, 'w')

SameCount = 0

TotalCount = 0

DiffDist = []

for House in FocalHouse.keys():

HouseOrder = 0

for k in Prek[House]:

if k == max(Prek[House]): #looks through set of potential relationships to find the closest one

FocalHouse[House].append(PreHouse[House][HouseOrder])

else:

HouseOrder += 1

TotalCount +=1

if FocalHouse[House][0] == FocalHouse[House][1]:#counts those from same house and writes them to output file.

SameCount +=1

Post = FocalHouse[House][0]

Pre = FocalHouse[House][1]

DiffFile.write(Post + "\t" + Pre + "\t" + "0\n")

else: #if different houses, calculates and saves inter-house distance in degrees and writes data to output file.

Post = FocalHouse[House][0]

Pre = FocalHouse[House][1]

X1 = LocDict[Post][0]

X2 = LocDict[Pre][0]

Y1 = LocDict[Post][1]

Y2 = LocDict[Pre][1]

Dist = math.pow(math.pow((X1-X2),2)+ math.pow((Y1-Y2),2),0.5)

DiffDist.append(Dist)

DiffFile.write(Post + "\t" + Pre + "\t" + str(Dist) + "\n")

ObservedProp = float(SameCount)/TotalCount

DiffSum = 0

DiffCount = 0

DiffSet = []

for item in DiffDist: #tallies counts and calculates sum of dstances for calculating average

DiffCount += 1

DiffSum += item

DiffSet.append(item)

ObservedMeanDist = float(DiffSum/DiffCount)

DiffSet.sort()

print "The observed proportion of nearest relative from the same house is %.3f" % ObservedProp

print "The observed mean distance betwen houses for dispersed insects is %.6f" % ObservedMeanDist

print " The observed median distance between houses for dispersed insects is %.6f" % DiffSet[DiffCount/2]

RandFileName = "Randomized\_distances\_CAR.tsv" #saves results from first run of randomized simulation.

RandFile = open(RandFileName, 'w')

RandList = []

DiffList = []

RandCounter = 0

for X in range(2,1002): #1000 randomizations in total.

SameCount = 0

TotalCount = 0

DiffDist = []

for House in FocalHouse.keys(): #uses same post-spray samples as observed - nearest relative is what is randomized, with replacement (from all pre-spray samples).

Random = random.choice(PreHouse[House])

FocalHouse[House].append(Random)

TotalCount +=1

if FocalHouse[House][0] == FocalHouse[House][X]: #same calculations here as with observed.

SameCount +=1

if RandCounter == 0:

Post = FocalHouse[House][0]

Pre = FocalHouse[House][X]

RandFile.write(Post + "\t" + Pre + "\t" + "0\n")

RandCounter += 1

else:

Post = FocalHouse[House][0]

Pre = FocalHouse[House][X]

X1 = LocDict[Post][0]

X2 = LocDict[Pre][0]

Y1 = LocDict[Post][1]

Y2 = LocDict[Pre][1]

Dist = math.pow(math.pow((X1-X2),2)+ math.pow((Y1-Y2),2),0.5)

DiffDist.append(Dist)

if RandCounter == 0:

RandFile.write(Post + "\t" + Pre + "\t" + str(Dist) + "\n")

RandProp = float(SameCount)/TotalCount

RandList.append(RandProp)

DiffSum = 0

DiffCount = 0

for item in DiffDist:

DiffCount += 1

DiffSum += item

RandDist = float(DiffSum/DiffCount)

DiffList.append(RandDist)

RandList.sort() #sorting list of same-house proportions to get significance thresholds.

print "Same house 0.05 cutoff is %.3f" % RandList[949]

print "Same house 0.01 cutoff is %.3f" %RandList[989]

print "Same house 0.001 cutoff is %.3f" %RandList[998]

DiffList.sort() #same here for inter-house distances (looking for shorter than expected, not longer than expected, so uses left-hand thresholds here)

print "Distance 0.05 cutoff is %.6f" % DiffList[50]

print "Distance 0.01 cutoff is %.6f" %DiffList[10]

print "Distance 0.001 cutoff is %.6f" %DiffList[1]

RandFile.close()

DiffFile.close()

'''

Here are the actual results when you run it:

The observed proportion of nearest relative from the same house is 0.472

The observed mean distance betwen houses for dispersed insects is 0.002753

The observed median distance between houses for dispersed insects is 0.002689

Same house 0.05 cutoff is 0.125

Same house 0.01 cutoff is 0.153

Same house 0.001 cutoff is 0.194

Distance 0.05 cutoff is 0.004152

Distance 0.01 cutoff is 0.003973

Distance 0.001 cutoff is 0.003778

'''