R Code for HostSwitch paper (Trivellone, Araujo, Panassiti, 2021). Includes the three examples of application of HostSwitch simulation using real data described in the Usage scenarios section.

Valeria Trivellone, Sabrina B. L. Araujo & Bernd Panassiti

7/17/2021

Scenario 1: Cephaloleia-Zingeriberales (wildlife ecology)

## Code showed for Scenario 1 only.

```
# MODEL SETTINGS
simulations = seq(100, 200, by = 50)  # set the iterations (n_sim) you want to compare
# in the results. Here we compare 100, 150 and 200
# iterations
df = parli$Cephaloleia

# get the column names for Cephaloleia placida
namesCp = colnames(df)[grep("Cp", colnames(df))]
# get the column names for Cephaloleia belti
namesCb = colnames(df)[grep("Cb", colnames(df))]

# creates all possible combinations between namesCp
# and namesCb
Combs = data.frame(expand.grid(a = namesCp, b = namesCb))

# prepares the dataframe to save results of
# comparisons (p-values, one for each estimated
# parameter - j, s, d).
testResult = data.frame(expand.grid(combs = apply(Combs,
    1, function(x) paste(x, collapse = "_")), simulations = simulations))
testResult = data.frame(apply(testResult, 2, as.character))
testResult = transform(testResult, group = do.call(rbind,
    strsplit(combs, "_", fixed = TRUE)), stringsAsFactors = F)
testResult$simulations = as.numeric(testResult$simulations)
testResult$p.value.j = NA
testResult$p.value.s = NA
testResult$p.value.d = NA

# prepare list for simHostSwitch results
simResult <- vector("list", length(simulations))
simResult <- sapply(paste("n_sim_", simulations, sep = ""),
    function(x) NULL)  # assign names

# Prepare 8 HostSwitch objects using the columns of
# Cephaloleia dataset loop for simulations
for (i in 1:length(simulations)) {

    df["n_sim", ] = rep(simulations[i], length(df["n_sim",
        ]))  # overwrite number simulations

    simResult[[i]] = lapply(colnames(df), function(x) simHostSwitch(data = df,
        column = x))
    names(simResult[[i]]) = colnames(df)
}

# Perform 16 pairwise comparisons for 3 parameters
# using t test in testHostSwitch function
for (i in 1:nrow(testResult)) {

    Nsim = which(paste("n_sim_", testResult$simulations[i],
        sep = "") == names(simResult))  # get simulation -> list id
    Ngroup1 = which(testResult$group.1[i] == names(simResult[[Nsim]]))
    Ngroup2 = which(testResult$group.2[i] == names(simResult[[Nsim]]))

    # jump
    simTestResult = testHostSwitch(simResult[[Nsim]][[Ngroup1]],
        simResult[[Nsim]][[Ngroup2]], "j", "t", plot = F)
    testResult$p.value.j[i] = simTestResult$p.value
    # survive
    simTestResult = testHostSwitch(simResult[[Nsim]][[Ngroup1]],
        simResult[[Nsim]][[Ngroup2]], "s", "t", plot = F)
    testResult$p.value.s[i] = simTestResult$p.value
    # distance
    simTestResult = testHostSwitch(simResult[[Nsim]][[Ngroup1]],
        simResult[[Nsim]][[Ngroup2]], "d", "t", plot = F)
    testResult$p.value.d[i] = simTestResult$p.value
}
# reshape p-value results for each estimated
# parameter
testResult.Cephaloleia = testResult
dcast(testResult.Cephaloleia, combs ~ simulations,
    value.var = "p.value.j")
```

```
##                combs          100          150          200
## 1  Cp.mHxjN_Cb.mHxjN 4.476789e-11 2.313005e-18 9.669690e-24
## 2  Cp.mHxjN_Cb.mHxjY 2.015718e-12 4.282837e-16 4.861736e-20
## 3  Cp.mHxjN_Cb.mLxjN 3.679699e-11 1.032886e-15 5.083653e-22
## 4  Cp.mHxjN_Cb.mLxjY 1.716224e-08 1.354382e-13 2.155270e-17
## 5  Cp.mHxjY_Cb.mHxjN 2.111848e-10 1.053538e-17 1.628592e-22
## 6  Cp.mHxjY_Cb.mHxjY 7.788608e-12 1.440305e-15 4.424558e-19
## 7  Cp.mHxjY_Cb.mLxjN 1.101130e-10 3.036101e-15 4.413821e-21
## 8  Cp.mHxjY_Cb.mLxjY 4.644185e-08 3.890954e-13 1.568426e-16
## 9  Cp.mLxjN_Cb.mHxjN 3.859274e-11 2.275179e-19 2.267250e-25
## 10 Cp.mLxjN_Cb.mHxjY 1.659444e-12 6.120252e-17 2.329420e-21
## 11 Cp.mLxjN_Cb.mLxjN 3.025446e-11 1.747767e-16 2.543285e-23
## 12 Cp.mLxjN_Cb.mLxjY 1.432854e-08 2.394964e-14 1.354331e-18
## 13 Cp.mLxjY_Cb.mHxjN 1.417392e-11 6.457778e-20 2.890690e-26
## 14 Cp.mLxjY_Cb.mHxjY 7.172344e-13 2.237621e-17 4.622140e-22
## 15 Cp.mLxjY_Cb.mLxjN 1.558029e-11 7.177946e-17 5.304007e-24
## 16 Cp.mLxjY_Cb.mLxjY 7.815040e-09 9.933703e-15 3.136143e-19
```

```r
dcast(testResult.Cephaloleia, combs ~ simulations,
    value.var = "p.value.s")
```

```
##                combs          100          150          200
## 1  Cp.mHxjN_Cb.mHxjN 6.217382e-13 3.261113e-20 3.917914e-26
## 2  Cp.mHxjN_Cb.mHxjY 2.571541e-14 4.001581e-19 1.725411e-24
## 3  Cp.mHxjN_Cb.mLxjN 7.325191e-14 6.603974e-19 1.646445e-26
## 4  Cp.mHxjN_Cb.mLxjY 1.521360e-11 1.870196e-17 6.140489e-21
## 5  Cp.mHxjY_Cb.mHxjN 3.210548e-12 1.970846e-19 1.005429e-24
## 6  Cp.mHxjY_Cb.mHxjY 1.184998e-13 2.080658e-18 3.224478e-23
## 7  Cp.mHxjY_Cb.mLxjN 3.408455e-13 3.526209e-18 4.133825e-25
## 8  Cp.mHxjY_Cb.mLxjY 7.658772e-11 1.105853e-16 1.149385e-19
## 9  Cp.mLxjN_Cb.mHxjN 1.351784e-13 1.900966e-21 1.000220e-27
## 10 Cp.mLxjN_Cb.mHxjY 6.224393e-15 2.940224e-20 6.250594e-26
## 11 Cp.mLxjN_Cb.mLxjN 1.755984e-14 4.651880e-20 4.314810e-28
## 12 Cp.mLxjN_Cb.mLxjY 3.368951e-12 1.116502e-18 2.180913e-22
## 13 Cp.mLxjY_Cb.mHxjN 7.434223e-14 8.383300e-22 5.321850e-28
## 14 Cp.mLxjY_Cb.mHxjY 3.590616e-15 1.384218e-20 3.529239e-26
## 15 Cp.mLxjY_Cb.mLxjN 1.007594e-14 2.161946e-20 2.307824e-28
## 16 Cp.mLxjY_Cb.mLxjY 1.858176e-12 4.923404e-19 1.221408e-22
```

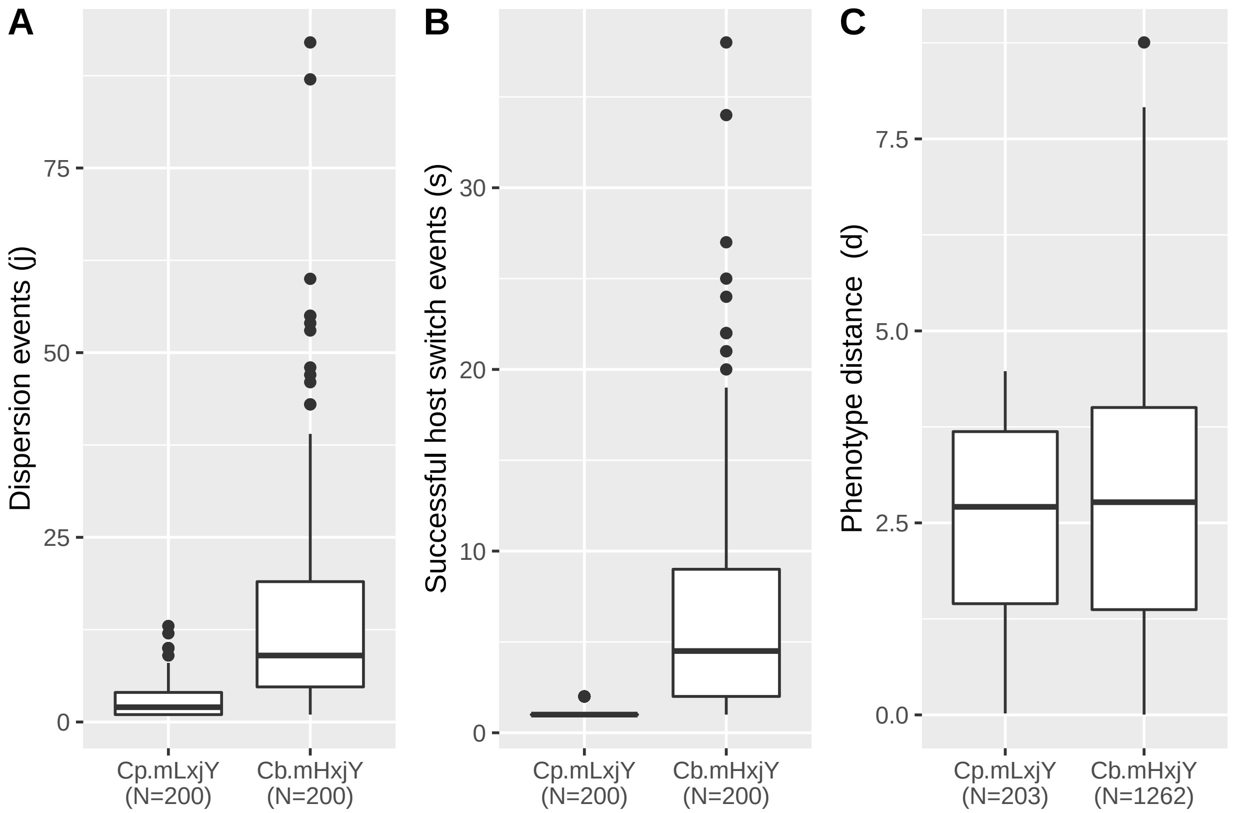
```r
dcast(testResult.Cephaloleia, combs ~ simulations,
    value.var = "p.value.d")
```

```
##                combs          100          150          200
## 1  Cp.mHxjN_Cb.mHxjN 0.4878877377 9.581379e-01 8.882705e-01
## 2  Cp.mHxjN_Cb.mHxjY 0.7533598917 1.993735e-01 2.439203e-01
## 3  Cp.mHxjN_Cb.mLxjN 0.0289345625 2.150971e-03 1.192113e-03
## 4  Cp.mHxjN_Cb.mLxjY 0.0011101184 3.231848e-06 3.699167e-06
## 5  Cp.mHxjY_Cb.mHxjN 0.1344493948 2.344102e-01 3.342520e-01
## 6  Cp.mHxjY_Cb.mHxjY 0.7102271451 8.379813e-01 6.208820e-01
## 7  Cp.mHxjY_Cb.mLxjN 0.0837818535 2.687223e-02 4.905075e-03
## 8  Cp.mHxjY_Cb.mLxjY 0.0031639355 5.453258e-05 1.211737e-05
## 9  Cp.mLxjN_Cb.mHxjN 0.8831183513 9.321297e-01 8.218317e-01
## 10 Cp.mLxjN_Cb.mHxjY 0.3772622534 2.340559e-01 2.646676e-01
## 11 Cp.mLxjN_Cb.mLxjN 0.0053006948 2.364061e-03 1.161424e-03
## 12 Cp.mLxjN_Cb.mLxjY 0.0001142616 2.818006e-06 2.989778e-06
## 13 Cp.mLxjY_Cb.mHxjN 0.5423813300 5.993765e-01 2.957937e-01
## 14 Cp.mLxjY_Cb.mHxjY 0.6686491935 8.609989e-02 2.179861e-02
## 15 Cp.mLxjY_Cb.mLxjN 0.0186691001 5.671991e-04 1.779197e-05
## 16 Cp.mLxjY_Cb.mLxjY 0.0005613414 6.623014e-07 2.226651e-08
```

Plot for Cephaloleia

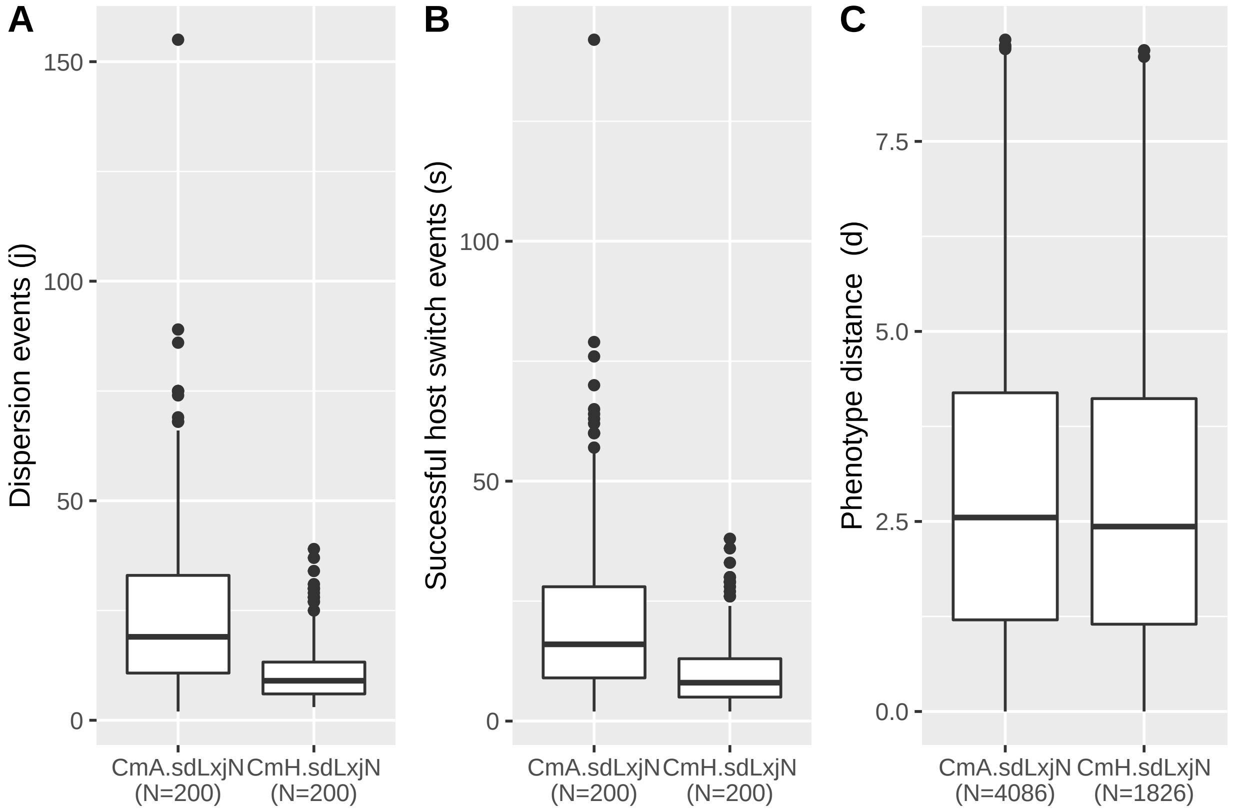
```r
# Choose one combination defining Ngroup1 and
# Ngroup2 this can be customized to get different
# plot
Ngroup1 = "Cp.mLxjY"
Ngroup2 = "Cb.mHxjY"
# Choose the data from a specific saved n_sim list
# of objects this can be customized to get
# different plot
Nsim = "n_sim_200"

# Build the plot for # of of dispersing events (j)
simTestResult = testHostSwitch(simResult[[Nsim]][[Ngroup1]],
    simResult[[Nsim]][[Ngroup2]], "j", "t", plot = F)

# simTestResult@.Data[[11]] -> groups = simulation1
# and simulation2 simTestResult@.Data[[12]] ->
# values
plotInput = data.frame(x = simTestResult@.Data[[11]],
    values = simTestResult@.Data[[12]])
plotInput$x = as.factor(plotInput$x)
xlabs <- paste((as.factor(c(Ngroup1, Ngroup2))), "\n(N=",
    table(plotInput$x), ")", sep = "")

j.ceph = ggplot(data = plotInput, aes(x = x, y = values,
    group = x)) + geom_boxplot() + labs(x = "", y = "Dispersion events (j)") +
    scale_x_discrete(labels = xlabs)

# Build the plot for # of successful host switch s
simTestResult = testHostSwitch(simResult[[Nsim]][[Ngroup1]],
    simResult[[Nsim]][[Ngroup2]], "s", "t", plot = F)

# simTestResult@.Data[[11]] -> groups = simulation1
# and simulation2 simTestResult@.Data[[12]] ->
# values
plotInput = data.frame(x = simTestResult@.Data[[11]],
    values = simTestResult@.Data[[12]])
plotInput$x = as.factor(plotInput$x)
xlabs <- paste((as.factor(c(Ngroup1, Ngroup2))), "\n(N=",
    table(plotInput$x), ")", sep = "")

s.ceph = ggplot(data = plotInput, aes(x = x, y = values,
    group = x)) + geom_boxplot() + labs(x = "", y = "Successful host switch events (s)") +
    scale_x_discrete(labels = xlabs)

# Build the plot for distance between the pRes_sim
# and pRes_new_sim when successful switch occurs d
simTestResult = testHostSwitch(simResult[[Nsim]][[Ngroup1]],
    simResult[[Nsim]][[Ngroup2]], "d", "t", plot = F)

# simTestResult@.Data[[11]] -> groups = simulation1
# and simulation2 simTestResult@.Data[[12]] ->
# values
plotInput = data.frame(x = simTestResult@.Data[[11]],
    values = simTestResult@.Data[[12]])
plotInput$x = as.factor(plotInput$x)
```

```
xlabs <- paste((as.factor(c(Ngroup1, Ngroup2))), "\n(N=",
    table(plotInput$x), ")", sep = "")

d.ceph = ggplot(data = plotInput, aes(x = x, y = values,
    group = x)) + geom_boxplot() + labs(x = "", y = "Phenotype distance  (d)") +
    scale_x_discrete(labels = xlabs)

figure1 <- ggarrange(j.ceph, s.ceph, d.ceph, labels = c("A",
    "B", "C"), ncol = 3, nrow = 1)
figure1
```



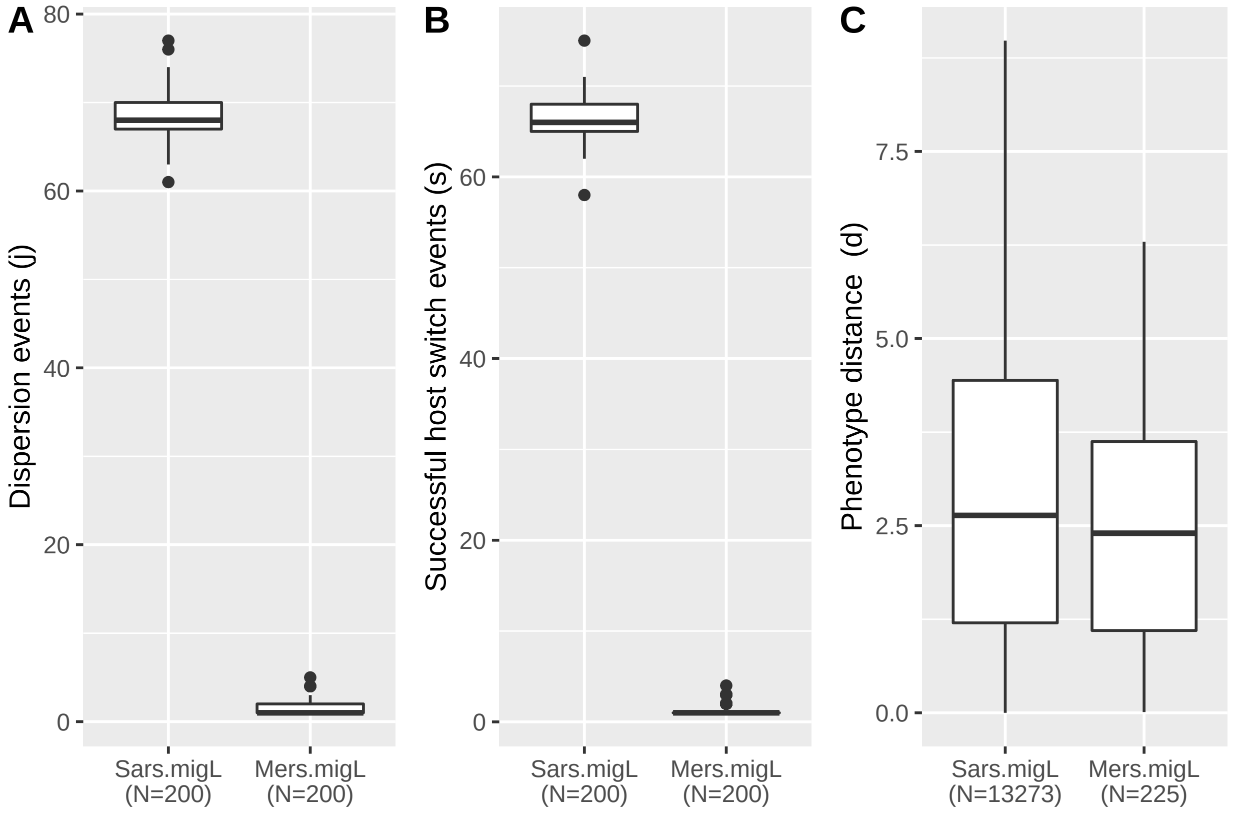
Scenario 2: Cacopsylla melanoneura-Rosaceae (agricultural pests)

Plot for Cacopsylla



Scenario 3: Sarbecovirus sp. and Merbacovirus sp. (SARS-MERS)-Mammals (zoonotic pathogens)

Plot for SarsMers



System information

```
R.Version()
```

```
## $platform
## [1] "x86_64-w64-mingw32"
##
## $arch
## [1] "x86_64"
##
## $os
## [1] "mingw32"
##
## $system
## [1] "x86_64, mingw32"
##
## $status
## [1] ""
##
## $major
## [1] "4"
##
```

```
## $minor
## [1] "0.5"
##
## $year
## [1] "2021"
##
## $month
## [1] "03"
##
## $day
## [1] "31"
##
## $‘svn rev‘
## [1] "80133"
##
## $language
## [1] "R"
##
## $version.string
## [1] "R version 4.0.5 (2021-03-31)"
##
## $nickname
## [1] "Shake and Throw"
```