

# T2.D5\_\_LabyInoc\_\_PrelimAnalyse

*Forest Schenck*

*August 17, 2016*

Packages Library

uploading data

GENERAL

ANALYSES

NOTES

MESOCOSM

Question 1: What are the effects of wasting disease on seagrass physiology and morphology

Survival

Shoot Production

Leaf number

Leaf production

Leaf loss

leaf growth

Aboveground biomass

Belowground Biomass

Rhizome Growth

Lesion Percent cover

Whole Shoot Wasting Index (modified) Day 28

Eelgrass Wasting Disease Shoot Severity Day 28

FIELD

Prevalence

Density

Length

leaf number

Environmental Variables

```

#Simulated Annealing
##Function for SA disease model

solve.Disease.SA <- function(t, y, parms) {
  y[y < 0] <- 0
  with(c(parms, y), {
    y[y < 0] <- 0
    dS <- ai - b * S * I # as * S + ai * I (remove per leaf)
    dI <- b * S * I - mi # * I remove per leaf
    dD <- mi
    return(list(c(dS, dI, dD)))
  })
}

#as = healthy plant leaf production rate (currently ommited because no healthy plants)
#ai =diseased plant leaf production rate
#b = transmission rate
#ms = healthy leaf mortality rate (from healthy SA model, no healthy leaf mortality
#in disease treatment)
#mi = infected leaf mortality rate

##Uploading data
mesocosm.transmission <- read.csv("transmission.SimA.model.csv")
mesocosm.transmission.CB <- mesocosm.transmission[c(1:11), c(2:35)]
mesocosm.transmission.LP <- mesocosm.transmission[c(12:22), c(2:35)]
mesocosm.transmission.NB <- mesocosm.transmission[c(23:33), c(2:35)]
mesocosm.transmission.WB <- mesocosm.transmission[c(34:44), c(2:35)]

##Cost function
cost_function <- function(x, times, NO, data) {
  ai <- x[1] #ai <- x[4]
  b <- x[2]
  mi <- x[3]
  if (ai > 0 & b > 0 & mi > 0) { # & ai > 0
    sim <- ode(y = NO, t = times, func = solve.Disease.SA,
              parms = list(ai = x[1], b = x[2], mi = x[3]), #, ai = x[4]
              method = "ode45")[c(1,2,3,4,5,6,8,11,15,22,29),]
    if (NROW(sim) == NROW(data)) {
      rmse <- sqrt(mean((as.numeric(data[,2] - (sim[,2])))^2)) +
               sqrt(mean((as.numeric(data[,3] - (sim[,3])))^2)) +
               sqrt(mean((as.numeric(data[,4] - (sim[,4])))^2))
    } else {
      rmse <- Inf
    }
  } else {
    rmse <- Inf
  }
}

##Acceptance function
acceptance <- function(cost_curr, cost_prop, temp) {
  if (cost_prop < cost_curr) {
    probab_accept <- 1
  }
}

```

```

} else {
  prob_accept <- exp(-(cost_prop - cost_curr)/temp)
}
return(prob_accept)
}

##simulated annealing (main simulation function)
sim_annealing <- function(I, tc, tr, data, times, NO) {
  rmse <- numeric(length = I); pars <- matrix(nrow = I, ncol = 3) #4
  colnames(pars) <- c("ai", "b", "mi") #, "as"
  x_curr <- c(runif(1, min = 0, max = 1), runif(1, min = 0, max = 1),
             runif(1, min = 0, max = 1)) #, runif(1, min = 0, max = 0.1) Change bounds
  rmse_curr <- cost_function(x_curr, times, NO, data)

  for (i in 1:I) {
    temp <- tc * exp(- tr * i / I)
    x_prop <- c(runif(1, min = 0, max = 1), runif(1, min = 0, max = 1),
              runif(1, min = 0, max = 1)) #, runif(1, min = 0, max = 0.1) Change bounds
    rmse_prop <- cost_function(x_prop, times, NO, data)

    if (acceptance(rmse_curr, rmse_prop, temp) >= runif(1)) {
      x_curr <- x_prop
      rmse_curr <- rmse_prop
    }
    pars[i, ] <- x_curr
    rmse[i] <- rmse_curr
  }
  return(list(x = x_curr, pars = pars, rmse = rmse))
}

##Trace plot
trace_plot <- function(x, burnin = 0) {
  names <- colnames(x)
  n <- ncol(x)
  par(mfrow = c(n, 2))
  for (i in 1:n) {
    plot(x[, i], type = "l", xlab = "Iterations", ylab = names[i],
        ylim = range(x[, i]), main = paste0("Trace of ", names[i]))
    d <- density(x[, i])
    plot(d$y, d$x, xlab = "Density", ylab = names[i], type = "l", ylim = range(x[, i]),
        main = paste0("Density of ", names[i]))
    abline(h = x[nrow(x), i], col = "red")
  }
}

##Running Simulations
###single data set
data <- mesocosm.transmission.CB[, c(1, 2, 13, 24)]
times <- 0:28
NO <- c(S = mesocosm.transmission.CB[1, 2], I = 1, D = 0)
sims <- sim_annealing(I = 10^4, tc = 10^8, tr = 20, data, times, NO) #If you get this error message: #E
#REDUCE THE RANGE OF PARAMETER!!!! UGH

```

```

trace_plot(sims$pars)

sims$pars[100,]
sims$x

str(sims)

sim.final <-ode(y = N0, t = times, func = solve.Disease.SA,
               parms = list(as = 0.067, b = 0.245, mi = 0.062),
               method = "ode45")[c(1,2,3,4,5,6,8,11,15,22,29),]

## Plotting data
par(mfrow = c(1,1))
plot(s.M1 ~ Day, data = data, col = "green", ylim = c(0,5)) #Experiment susceptible
points(i.M1 ~ Day, data = data, col = "red") #Experiment infected
points(d.M1 ~ Day, data = data) #Experiment dead
points(S ~ time, data = sim.final, t = "1", col = "green") #simulated susceptible
points(I ~ time, data = sim.final, t = "1", col = "red") #simulated infected
points(D ~ time, data = sim.final, t = "1") #simulated dead

###multiple data sets
set.seed(1)
pars.sim <- matrix(NA, nrow = 11, ncol = 3) #
data.temp <- matrix(NA, nrow = 11, ncol = 34)
pars.sim.master <- array(NA, c(11,3,4))
for(j in 0:3) {
  data.temp <- mesocosm.transmission[c((1+11*j):(11+11*j)), c(2:35)]
  for (i in 1:11) {
    data <- data.temp[, c(1,i + 1, i + 12, i + 23)]
    times <- 0:28
    N0 <- c(S = data.temp[1, i + 1], I = 1, D = 0)
    sims <- sim_annealing(I = 10^4, tc = 10^8, tr = 20, data, times, N0)
    pars.sim[i,] <- sims$x
  }
  pars.sim.master[ , , j+1] <- pars.sim
}

pars.sim.master

##CB
#pars.sim.CB <- matrix(NA, nrow = 11, ncol = 4)
#for (i in 1:11) {
#  data <- mesocosm.transmission.CB[, c(1,i + 1, i + 12, i + 23)]
#  times <- 0:28
#  N0 <- c(S = mesocosm.transmission.CB[1, i + 1], I = 1, D = 0)
#  sims <- sim_annealing(I = 10^4, tc = 10^8, tr = 20, data, times, N0)
#  pars.sim.CB[i,] <- sims$x
#}
#pars.sim.CB

## running model with parameters from sim
sim.final <- array(NA, c(11,4,44), )
for (j in 0:3) {

```

```

data.temp.sim <- mesocosm.transmission[c((1+11*j):(11+11*j)), c(2:35)]
for (i in 1:11) {
  NO <- c(S = data.temp.sim[1, i + 1], I = 1, D = 0)
  sim.final[, , (i+(j*11))] <-ode(y = NO, t = times, func = solve.Disease.SA,
    parms = list(ai = pars.sim.master[i, 1, (j+1)],
      b = pars.sim.master[i, 2, (j+1)],
      mi = pars.sim.master[i, 3, (j+1)]),
    method = "ode45")[c(1,2,3,4,5,6,8,11,15,22,29),] #, ai = pars.sim.master[i,
}
}

sim.final

##PLOTING MODEL
par(mfrow = c(2,2))
for (i in 0:3) {
  data.temp.sim <- mesocosm.transmission[c((1+11*i):(11+11*i)), c(2:35)]
  plot(data.temp.sim[,2] ~ data.temp.sim[, 1], data = data.temp.sim, col = "green", ylim = c(0,6))
  for (j in 2:11) {
    points(data.temp.sim[, 1 + j] ~ data.temp.sim[, 1], data = data.temp.sim, col = "green")
  }
  for (n in 1:11) {
    points(data.temp.sim[, n + 12] ~ data.temp.sim[, 1], data = data.temp.sim, col = "red") #Experiment i
    points(data.temp.sim[, n + 23] ~ data.temp.sim[, 1], data = data.temp.sim) #Experiment dead
    points(sim.final[, 2, (n + i * 11)] ~ sim.final[, 1, (n + i * 11)], data = sim.final[, , (n + i * 11)])
    points(sim.final[, 3, (n + i * 11)] ~ sim.final[, 1, (n + i * 11)], data = sim.final[, , (n + i * 11)])
    points(sim.final[, 4, (n + i * 11)] ~ sim.final[, 1, (n + i * 11)], data = sim.final[, , (n + i * 11)])
  }
}

##MEANS, SE, and SD

means.cb <- cbind(mean(pars.sim.master[, 1, 1]), mean(pars.sim.master[, 2, 1]), mean(pars.sim.master[, 3, 1]))
means.lp <- cbind(mean(pars.sim.master[, 1, 2]), mean(pars.sim.master[, 2, 2]), mean(pars.sim.master[, 3, 2]))
means.nb <- cbind(mean(pars.sim.master[, 1, 3]), mean(pars.sim.master[, 2, 3]), mean(pars.sim.master[, 3, 3]))
means.wb <- cbind(mean(pars.sim.master[, 1, 4]), mean(pars.sim.master[, 2, 4]), mean(pars.sim.master[, 3, 4]))
clabs <- c("ai", "b", "mi")

rbind(clabs, means.cb, means.lp, means.nb, means.wb)

se.cb <- cbind(se(pars.sim.master[, 1, 1]), se(pars.sim.master[, 2, 1]), se(pars.sim.master[, 3, 1]))
se.lp <- cbind(se(pars.sim.master[, 1, 2]), se(pars.sim.master[, 2, 2]), se(pars.sim.master[, 3, 2]))
se.nb <- cbind(se(pars.sim.master[, 1, 3]), se(pars.sim.master[, 2, 3]), se(pars.sim.master[, 3, 3]))
se.wb <- cbind(se(pars.sim.master[, 1, 4]), se(pars.sim.master[, 2, 4]), se(pars.sim.master[, 3, 4]))

rbind(se.cb, se.lp, se.nb, se.wb)

```

## (S)Simulated Annealing

```

##Function for SA healthy model
solve.Healthy.SA <- function(t, y, parms) {
  with(c(parms, y), {

```

```

    y[y < 0] <- 0
    dS <- as - ms
    dD <- ms
    return(list(c(dS, dD)))
  })
}

##Uploading data
mesocosm.transmission.S <- read.csv("transmission.SimA.model.S.csv")
mesocosm.transmission.CB.S <- mesocosm.transmission.S[c(1:11), c(2:10, 14:21)]
mesocosm.transmission.LP.S <- mesocosm.transmission.S[c(12:22), c(2:24)]
mesocosm.transmission.NB.S <- mesocosm.transmission.S[c(23:33), c(2:24)]
mesocosm.transmission.WB.S <- mesocosm.transmission.S[c(34:44), c(2:8, 14:19)]

##Cost function
cost_function.S <- function(x, times, NO, data) {
  as <- x[1]
  ms <- x[2]
  if (as >= 0 & ms >= 0) {
    sim <- ode(y = NO, t = times, func = solve.Healthy.SA,
              parms = list(as = x[1], ms = x[2]),
              method = "ode45")[c(1,2,3,4,5,6,8,11,15,22,29),]
    if (NROW(sim) == NROW(data)) {
      rmse <- sqrt(mean((as.numeric(data[,2] - (sim[,2])))^2)) +
               sqrt(mean((as.numeric(data[,3] - (sim[,3])))^2))
    } else {
      rmse <- Inf
    }
  } else {
    rmse <- Inf
  }
}

##Acceptance function
acceptance.S <- function(cost_curr, cost_prop, temp) {
  if (cost_prop < cost_curr) {
    prob_accept <- 1
  } else {
    prob_accept <- exp(-(cost_prop - cost_curr)/temp)
  }
  return(prob_accept)
}

##simulated annealing (main simulation function)
sim_annealing.S <- function(I, tc, tr, data, times, NO) {
  rmse <- numeric(length = I); pars <- matrix(nrow = I, ncol = 2) #4
  colnames(pars) <- c("as", "ms")
  x_curr <- c(runif(1, min = 0, max = 1), runif(1, min = 0, max = 1)) #Change bounds if error
  rmse_curr <- cost_function.S(x_curr, times, NO, data)

  for (i in 1:I) {
    temp <- tc * exp(- tr * i / I)
    x_prop <- c(runif(1, min = 0, max = 1), runif(1, min = 0, max = 1)) #Change bounds if error

```

```

rmse_prop <- cost_function.S(x_prop, times, NO, data)

if (acceptance.S(rmse_curr, rmse_prop, temp) >= runif(1)) {
  x_curr <- x_prop
  rmse_curr <- rmse_prop
}
pars[i, ] <- x_curr
rmse[i] <- rmse_curr
}
return(list(x = x_curr, pars = pars, rmse = rmse))
}

##CB
pars.sim.CB <- matrix(NA, nrow = 8, ncol = 2)
for (i in 1:8) {
  data <- mesocosm.transmission.CB.S[, c(1,i + 1, i + 9)]
  times <- 0:28
  NO <- c(S = mesocosm.transmission.CB.S[1, i + 1], D = 0)
  sims <- sim_annealing.S(I = 10^4, tc = 10^8, tr = 20, data, times, NO)
  pars.sim.CB[i,] <- sims$x
}
pars.sim.CB

#Simulating data based on parms for Sim.Annealing
sim.final <- array(NA, c(11,3,8), )
for (i in 1:8) {
  NO <- c(S = mesocosm.transmission.CB.S[1, i + 1], D = 0)
  sim.final[, , i] <-ode(y = NO, t = times, func = solve.Healthy.SA,
    parms = list(as = pars.sim.CB[i,1], ms = pars.sim.CB[i,2]),
    method = "ode45")[c(1,2,3,4,5,6,8,11,15,22,29),]
}

sim.final

## Plotting CB data
par(mfrow = c(1,1))
plot(mesocosm.transmission.CB.S[,2] ~ Day, data = mesocosm.transmission.CB.S, col = "green", ylim = c(
for (i in 1:8) {
  points(mesocosm.transmission.CB.S[,i+1] ~ Day, data = mesocosm.transmission.CB.S, col = "green") #Exp
  points(mesocosm.transmission.CB.S[,i+9] ~ Day, data = mesocosm.transmission.CB.S) #Experiment dead
  points(sim.final[,2,i] ~ sim.final[,1,i], data = sim.final[,i], t = "1", col = "green") #simulated s
  points(sim.final[,3,i] ~ sim.final[,1,i], data = sim.final[,i], t = "1") #simulated dead
}

##LP
pars.sim.LP <- matrix(NA, nrow = 11, ncol = 2)
for (i in 1:11) {
  data <- mesocosm.transmission.LP.S[, c(1,i + 1, i + 12)]
  times <- 0:28
  NO <- c(S = mesocosm.transmission.LP.S[1, i + 1], D = 0)
  sims <- sim_annealing.S(I = 10^4, tc = 10^8, tr = 20, data, times, NO)
  pars.sim.LP[i,] <- sims$x
}

```



```

}
pars.sim.LP

#Simulating data based on parms for Sim.Annealing
sim.final <- array(NA, c(11,3,11), )
for (i in 1:11) {
  NO <- c(S = mesocosm.transmission.LP.S[1, i + 1], D = 0)
  sim.final[, , i] <-ode(y = NO, t = times, func = solve.Healthy.SA,
    parms = list(as = pars.sim.LP[i,1], ms = pars.sim.LP[i,2]),
    method = "ode45")[c(1,2,3,4,5,6,8,11,15,22,29),]
}

sim.final

## Plotting LP data
par(mfrow = c(1,1))
plot(mesocosm.transmission.LP.S[,2] ~ Day, data = mesocosm.transmission.LP.S, col = "green", ylim = c
for (i in 1:11) {
  points(mesocosm.transmission.LP.S[,i+1] ~ Day, data = mesocosm.transmission.LP.S, col = "green") #Exp
  points(mesocosm.transmission.LP.S[,i+9] ~ Day, data = mesocosm.transmission.LP.S) #Experiment dead
  points(sim.final[,2,i] ~ sim.final[,1,i], data = sim.final[, ,i], t = "1", col = "green") #simulated s
  points(sim.final[,3,i] ~ sim.final[,1,i], data = sim.final[, ,i], t = "1") #simulated dead
}

##NB
pars.sim.NB <- matrix(NA, nrow = 11, ncol = 2)
for (i in 1:11) {
  data <- mesocosm.transmission.NB.S[, c(1,i + 1, i + 12)]
  times <- 0:28
  NO <- c(S = mesocosm.transmission.NB.S[1, i + 1], D = 0)
  sims <- sim_annealing.S(I = 10^4, tc = 10^8, tr = 20, data, times, NO)
  pars.sim.NB[i,] <- sims$x
}
pars.sim.NB

#Simulating data based on parms for Sim.Annealing
sim.final <- array(NA, c(11,3,11), )
for (i in 1:11) {
  NO <- c(S = mesocosm.transmission.NB.S[1, i + 1], D = 0)
  sim.final[, , i] <-ode(y = NO, t = times, func = solve.Healthy.SA,
    parms = list(as = pars.sim.NB[i,1], ms = pars.sim.NB[i,2]),
    method = "ode45")[c(1,2,3,4,5,6,8,11,15,22,29),]
}

sim.final

## Plotting NB data
par(mfrow = c(1,1))
plot(mesocosm.transmission.NB.S[,2] ~ Day, data = mesocosm.transmission.NB.S, col = "green", ylim = c
for (i in 1:11) {
  points(mesocosm.transmission.NB.S[,i+1] ~ Day, data = mesocosm.transmission.NB.S, col = "green") #Exp
  points(mesocosm.transmission.NB.S[,i+9] ~ Day, data = mesocosm.transmission.NB.S) #Experiment dead
  points(sim.final[,2,i] ~ sim.final[,1,i], data = sim.final[, ,i], t = "1", col = "green") #simulated s

```



```
SA.data.master <- matrix(site.vector, nrow = 44, ncol = 1)
SA.data.master <- as.data.frame(SA.data.master)
SA.data.master$ai <- matrix(rbind(pars.sim.master[, , 1], pars.sim.master[, , 2], pars.sim.master[, , 3]),
SA.data.master$beta <- matrix(rbind(pars.sim.master[, , 1], pars.sim.master[, , 2], pars.sim.master[, , 3]),
SA.data.master$mi <- matrix(rbind(pars.sim.master[, , 1], pars.sim.master[, , 2], pars.sim.master[, , 3]),
colnames(SA.data.master) <- c("site", "ai", "beta", "mi")

SA.data.master$sas <-c(pars.sim.CB[,1], NA, NA, NA, pars.sim.LP[,1], pars.sim.NB[,1], pars.sim.WB[,1], NA)
SA.data.master$ms <- c(pars.sim.CB[,2], NA, NA, NA, pars.sim.LP[,2], pars.sim.NB[,2], pars.sim.WB[,2], NA)
SA.data.master$patho <- SA.data.master$mi - SA.data.master$ms

str(SA.data.master)

##Healthy leaf production analyses (as)
SA.as.model <- aov(as ~ site, data = SA.data.master)
Anova(SA.as.model) ##No significant differences

##Healthy leaf loss analyses (ms)
SA.ms.model <-aov(ms ~ site, data = SA.data.master)
Anova(SA.ms.model) #significant effect of site

posthoc.SA.ms.model <- TukeyHSD(SA.ms.model)
multcompLetters(posthoc.SA.ms.model$`site`,["p adj"]) #nb = least, lp&cb = ND, wb = most

##Infected leaf production analyses (ai)
SA.ai.model <- aov(ai ~ site, data = SA.data.master)
Anova(SA.ai.model) #No significant differences

#Infected leaf loss analyses (mi)
SA.mi.model <- aov(mi ~ site, data = SA.data.master)
Anova(SA.mi.model) #No significant differences

#Transmission analyses (beta)
SA.beta.model <- aov(beta ~ site, data = SA.data.master)
Anova(SA.beta.model) #Marginal difference
#CB (0.29), WB (0.54), LP (0.34) and NB (0.43)

#Pathogenicity analyses (patho) ##ANCOVA
SA.patho.model <- aov(patho ~ site, data = SA.data.master)
Anova(SA.patho.model) #significant differences

posthoc.SA.patho.model <- TukeyHSD(SA.patho.model)
multcompLetters(posthoc.SA.patho.model$`site`,["p adj"]) #wb = least, nb&cb = intermediate, lp = most
```

## FIELD MODEL

ABC (Approximate Bayesian Computation)

### Day0 Analyses

Data manipulation

Lesion Spread Figures and Analyses

Lesion Percent Cover Figures and Analyses

Length Figures and Analyses

### Day5 Analyses

Data manipulation

Lesion Spread Figures and Analyses

Lesion Percent Cover Figures and Analyses

Leaf Growth Figures and Analyses

Change in leaf number Figures and Analyses

### Day28 Analyses

Data manipulation

Lesion Spread Figures and Analyses

Lesion Percent Cover Figures and Analyses

Leaf Growth Figures and Analyses

Change in leaf number Figures and Analyses

Biomass Figures and Analysis

Rhizome Figures and Analyses

Home VS. Away

Data management

Biomass Figures and Analysis

## Environment Analysis

Data manipulation

Temperature Figures and Analyses