Mode models and landscape

Gan Yao

2023-04-13

This analysis include more models that are of our interest, as well as generating fitness landscape of these models.

Aaster graph

$root \rightarrow flCt \rightarrow flCtNotConsumed \rightarrow flCtUndamaged \rightarrow capsuleCt \rightarrow isHarvested \rightarrow ovuleCt \rightarrow embryoCt(fitness)$

Load library and data, transfer data

```
library(aster)
library(tidyverse)
library(plotly)
library(hrbrthemes)
library(viridis)
library(gridExtra)
library(metR)
```

```
data <- read.csv("data/output/remLilium2021Data30Nov2022.csv")
names(data)
```

```
##  [1] "id"                  "site"                "year"
##  [4] "Ax"                  "Ly"                  "flCt"
##  [7] "capsuleCt"           "nCapsulesHarvested"  "ovuleCt"
## [10] "embryoCt"            "nn1Dist"             "nn2Dist"
## [13] "nn3Dist"             "nn4Dist"             "nn5Dist"
## [16] "nn6Dist"             "nn7Dist"             "nn8Dist"
## [19] "nn9Dist"             "nn10Dist"            "nn1DistNotConsumed"
## [22] "nn2DistNotConsumed"  "nn3DistNotConsumed"  "nn4DistNotConsumed"
## [25] "nn5DistNotConsumed"  "nn6DistNotConsumed"  "nn7DistNotConsumed"
## [28] "nn8DistNotConsumed"  "nn9DistNotConsumed"  "nn10DistNotConsumed"
## [31] "fecundity"           "flCtNotConsumed"     "flCtUndamaged"
```

```
data <- data[data$site != "lf",]
data <- data[data$site != "wrrx",]
data[is.na(data$nCapsulesHarvested), 'nCapsulesHarvested'] <- 0
data[is.na(data$ovuleCt), 'ovuleCt'] <- 0
data[is.na(data$embryoCt), 'embryoCt'] <- 0
names(data)[names(data) == 'nCapsulesHarvested'] <- 'isHarvested'
```

```
pred <- c(0,1,2,3,4,5,6)
fam <- c(2,1,1,1,1,2,1)
vars <- c("flCt", "flCtNotConsumed","flCtUndamaged", "capsuleCt",
```

```
          "isHarvested", "ovuleCt", "embryoCt")
#test <- data %>% mutate(nn5Dist_s = nn5Dist/1000,
                         #nn5DistNotConsumed =replace_na(nn5DistNotConsumed, 0)) %>%
  #mutate(nn5DistNotConsumed_s = nn5DistNotConsumed/1000)

test <- data %>% mutate(nn5Dist_s = log(nn5Dist)/10,
                         nn5DistNotConsumed =replace_na(nn5DistNotConsumed, 1)) %>%
  mutate(nn5DistNotConsumed_s = log(nn5DistNotConsumed)/10)

redata <- reshape(test, varying = list(vars), direction="long", timevar="varb",
                  times = as.factor(vars), v.names="resp")

redata <- data.frame(redata, root = 1)
redata$fit <- as.numeric(redata$varb == "embryoCt")
redata$Nid <- as.numeric(gsub("[^0-9.-]", "", redata$id))

names(redata)
```

```
##  [1] "id"                   "site"                 "year"
##  [4] "Ax"                   "Ly"                   "nn1Dist"
##  [7] "nn2Dist"              "nn3Dist"              "nn4Dist"
## [10] "nn5Dist"              "nn6Dist"              "nn7Dist"
## [13] "nn8Dist"              "nn9Dist"              "nn10Dist"
## [16] "nn1DistNotConsumed"   "nn2DistNotConsumed"   "nn3DistNotConsumed"
## [19] "nn4DistNotConsumed"   "nn5DistNotConsumed"   "nn6DistNotConsumed"
## [22] "nn7DistNotConsumed"   "nn8DistNotConsumed"   "nn9DistNotConsumed"
## [25] "nn10DistNotConsumed"  "fecundity"            "nn5Dist_s"
## [28] "nn5DistNotConsumed_s" "varb"                 "resp"
## [31] "root"                 "fit"                  "Nid"
```

```
redata$Deer <- as.numeric(redata$varb=='flCtNotConsumed')
redata$Pollination <- as.numeric(is.element(redata$varb,
                                c("capsuleCt", "isHarvested", "ovuleCt", "embryoCt")))

names(redata)
```

```
##  [1] "id"                   "site"                 "year"
##  [4] "Ax"                   "Ly"                   "nn1Dist"
##  [7] "nn2Dist"              "nn3Dist"              "nn4Dist"
## [10] "nn5Dist"              "nn6Dist"              "nn7Dist"
## [13] "nn8Dist"              "nn9Dist"              "nn10Dist"
## [16] "nn1DistNotConsumed"   "nn2DistNotConsumed"   "nn3DistNotConsumed"
## [19] "nn4DistNotConsumed"   "nn5DistNotConsumed"   "nn6DistNotConsumed"
## [22] "nn7DistNotConsumed"   "nn8DistNotConsumed"   "nn9DistNotConsumed"
## [25] "nn10DistNotConsumed"  "fecundity"            "nn5Dist_s"
## [28] "nn5DistNotConsumed_s" "varb"                 "resp"
## [31] "root"                 "fit"                  "Nid"
## [34] "Deer"                 "Pollination"
```

```
# extreme_nnA <- unique(redata[redata$nn5Dist_s < 0,'id'])
# redata <- redata[(!redata$id %in% extreme_nnA), ]
```

Models

Model 0: Null Model

```
model.null <- aster(resp ~ -1 + varb,
                pred, fam,varb,id,root,data=redata)
#summary(model3, info.tol=1e-12)
summary(model.null)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb, pred = pred, fam = fam,
##     varvar = varb, idvar = id, root = root, data = redata)
##
##                         Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt       -1.135e+00  1.703e-01   -6.666 2.64e-11 ***
## varbembryoCt        -5.784e-01  1.189e-02  -48.645  < 2e-16 ***
## varbflCt            -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed -6.847e-02  1.075e-01   -0.637    0.524
## varbflCtUndamaged    1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested     -3.215e+02  1.855e+00 -173.370  < 2e-16 ***
## varbovuleCt          6.027e+00  7.127e-03  845.591  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
eigen(model.null$fisher)$val
```

```
## [1] 1.125517e+07 6.263672e+03 1.179341e+03 1.694817e+02 5.191389e+01
## [6] 3.340106e+01 2.906923e-01
```

Model 1: $Fitness : nnA$

```
model1 <- aster(resp ~ -1 + varb + fit:nn5Dist_s,
                pred, fam,varb,id,root,data=redata)

summary(model1)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + fit:nn5Dist_s, pred = pred,
##     fam = fam, varvar = varb, idvar = id, root = root, data = redata)
##
##                         Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt       -1.135e+00  1.703e-01   -6.666 2.64e-11 ***
## varbembryoCt        -5.788e-01  1.202e-02  -48.164  < 2e-16 ***
## varbflCt            -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed -6.847e-02  1.075e-01   -0.637    0.524
## varbflCtUndamaged    1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested     -3.215e+02  1.855e+00 -173.369  < 2e-16 ***
## varbovuleCt          6.027e+00  7.127e-03  845.591  < 2e-16 ***
## fit:nn5Dist_s        2.087e-03  9.088e-03    0.230    0.818
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 2: $Deer : nnA$

```
model2 <- aster(resp ~ -1 + varb + Deer:nn5Dist_s,
                pred, fam,varb,id,root,data=redata)

summary(model2)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + Deer:nn5Dist_s, pred = pred,
##     fam = fam, varvar = varb, idvar = id, root = root, data = redata)
##
##                         Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt       -1.135e+00  1.703e-01   -6.666 2.64e-11 ***
## varbembryoCt        -5.784e-01  1.189e-02  -48.645  < 2e-16 ***
## varbflCt            -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed -3.220e-01  1.397e-01   -2.305  0.02114 *
## varbflCtUndamaged    1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested     -3.215e+02  1.855e+00 -173.370  < 2e-16 ***
## varbovuleCt          6.027e+00  7.127e-03  845.591  < 2e-16 ***
## Deer:nn5Dist_s       1.310e+00  4.465e-01    2.935  0.00333 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 3: $Pollination : nnA$

```
model3 <- aster(resp ~ -1 + varb + Pollination:nn5Dist_s,
                pred, fam,varb,id,root,data=redata)

#summary(model3, info.tol=1e-9)
summary(model3)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + Pollination:nn5Dist_s,
##     pred = pred, fam = fam, varvar = varb, idvar = id, root = root,
##     data = redata)
##
##                           Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt         -1.136e+00  1.703e-01   -6.671 2.55e-11 ***
## varbembryoCt          -5.794e-01  1.190e-02  -48.677  < 2e-16 ***
## varbflCt              -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed   -6.847e-02  1.075e-01   -0.637   0.5241
## varbflCtUndamaged      1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested       -3.215e+02  1.855e+00 -173.360  < 2e-16 ***
## varbovuleCt            6.026e+00  7.146e-03  843.211  < 2e-16 ***
## Pollination:nn5Dist_s  4.522e-03  2.305e-03    1.962   0.0498 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 4: $Fitness : nnA + Deer : nnA$

```
model4 <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s,
                pred, fam,varb,id,root,data=redata)
```

```
summary(model4)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s,
##     pred = pred, fam = fam, varvar = varb, idvar = id, root = root,
##     data = redata)
##
##                         Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt       -1.135e+00  1.703e-01   -6.666 2.64e-11 ***
## varbembryoCt        -5.763e-01  1.203e-02  -47.925  < 2e-16 ***
## varbflCt            -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed -3.690e-01  1.464e-01   -2.521  0.01171 *
## varbflCtUndamaged    1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested     -3.215e+02  1.855e+00 -173.364  < 2e-16 ***
## varbovuleCt          6.027e+00  7.127e-03  845.591  < 2e-16 ***
## fit:nn5Dist_s       -1.109e-02  1.001e-02   -1.108  0.26788
## nn5Dist_s:Deer       1.545e+00  4.918e-01    3.140  0.00169 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 5: $Fitness : nnA + Pollination : nnA$

```
model5 <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Pollination:nn5Dist_s,
                pred, fam,varb,id,root,data=redata)

#summary(model5, info.tol=1e-9)
summary(model5)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + fit:nn5Dist_s + Pollination:nn5Dist_s,
##     pred = pred, fam = fam, varvar = varb, idvar = id, root = root,
##     data = redata)
##
##                           Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt         -1.216e+00  1.704e-01   -7.140 9.30e-13 ***
## varbembryoCt          -3.278e-01  1.424e-02  -23.017  < 2e-16 ***
## varbflCt              -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed   -6.847e-02  1.075e-01   -0.637    0.524
## varbflCtUndamaged      1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested       -3.210e+02  1.853e+00 -173.225  < 2e-16 ***
## varbovuleCt            5.933e+00  8.236e-03  720.346  < 2e-16 ***
## fit:nn5Dist_s         -1.637e+00  5.412e-02  -30.246  < 2e-16 ***
## nn5Dist_s:Pollination  4.281e-01  1.391e-02   30.781  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 6: $Deer : nnA + Pollination : nnA$

```
model6 <- aster(resp ~ -1 + varb + Deer:nn5Dist_s + Pollination:nn5Dist_s,
                pred, fam,varb,id,root,data=redata)
```

```
#summary(model6, info.tol=1e-9)
summary(model6)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + Deer:nn5Dist_s + Pollination:nn5Dist_s,
##     pred = pred, fam = fam, varvar = varb, idvar = id, root = root,
##     data = redata)
##
##                           Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt         -1.136e+00  1.703e-01   -6.668 2.60e-11 ***
## varbembryoCt          -5.788e-01  1.190e-02  -48.627  < 2e-16 ***
## varbflCt              -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed   -2.900e-01  1.457e-01   -1.990   0.0466 *
## varbflCtUndamaged      1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested       -3.215e+02  1.855e+00 -173.368  < 2e-16 ***
## varbovuleCt            6.026e+00  7.148e-03  843.053  < 2e-16 ***
## Deer:nn5Dist_s         1.150e+00  4.965e-01    2.316   0.0206 *
## nn5Dist_s:Pollination  1.925e-03  2.564e-03    0.751   0.4526
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 7: $Fitness : nnA + Deer : nnA + Pollination : nnA$

```
model7 <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s,
                pred, fam,varb,id,root,data=redata)

#summary(model7, info.tol=1e-9)
summary(model7)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s +
##     Pollination:nn5Dist_s, pred = pred, fam = fam, varvar = varb,
##     idvar = id, root = root, data = redata)
##
##                           Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt         -1.220e+00  1.704e-01   -7.160 8.09e-13 ***
## varbembryoCt          -3.278e-01  1.424e-02  -23.019  < 2e-16 ***
## varbflCt              -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed   -2.542e-01  1.453e-01   -1.749   0.0803 .
## varbflCtUndamaged      1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested       -3.210e+02  1.853e+00 -173.232  < 2e-16 ***
## varbovuleCt            5.934e+00  8.241e-03  720.046  < 2e-16 ***
## fit:nn5Dist_s         -1.634e+00  5.416e-02  -30.168  < 2e-16 ***
## nn5Dist_s:Deer         9.681e-01  4.978e-01    1.945   0.0518 .
## nn5Dist_s:Pollination  4.251e-01  1.400e-02   30.371  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 8: $Pollination : nnB$

```
model8 <- aster(resp ~ -1 + varb + Pollination:nn5DistNotConsumed_s,
                pred, fam,varb,id,root,data=redata)

summary(model8)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + Pollination:nn5DistNotConsumed_s,
##     pred = pred, fam = fam, varvar = varb, idvar = id, root = root,
##     data = redata)
##
##                                    Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt                    -1.137e+00  1.703e-01   -6.675 2.47e-11 ***
## varbembryoCt                     -5.811e-01  1.190e-02  -48.819  < 2e-16 ***
## varbflCt                         -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed              -6.847e-02  1.075e-01   -0.637    0.524
## varbflCtUndamaged                 1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested                  -3.213e+02  1.855e+00 -173.246  < 2e-16 ***
## varbovuleCt                       6.024e+00  7.147e-03  842.916  < 2e-16 ***
## Pollination:nn5DistNotConsumed_s  1.081e-02  1.604e-03    6.742 1.56e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 9: $Fitness : nnA + Pollination : nnB$

```
model9 <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Pollination:nn5DistNotConsumed_s,
                pred, fam,varb,id,root,data=redata)

summary(model9)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + fit:nn5Dist_s + Pollination:nn5DistNotConsumed_s,
##     pred = pred, fam = fam, varvar = varb, idvar = id, root = root,
##     data = redata)
##
##                                    Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt                    -1.138e+00  1.703e-01   -6.682 2.35e-11 ***
## varbembryoCt                     -5.709e-01  1.196e-02  -47.730  < 2e-16 ***
## varbflCt                         -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed              -6.847e-02  1.075e-01   -0.637    0.524
## varbflCtUndamaged                 1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested                  -3.211e+02  1.855e+00 -173.161  < 2e-16 ***
## varbovuleCt                       6.022e+00  7.162e-03  840.861  < 2e-16 ***
## fit:nn5Dist_s                    -6.488e-02  1.161e-02   -5.587 2.31e-08 ***
## Pollination:nn5DistNotConsumed_s  1.954e-02  2.114e-03    9.243  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 10: $Deer : nnA + Pollination : nnB$

```
model10 <- aster(resp ~ -1 + varb + Deer:nn5Dist_s + Pollination:nn5DistNotConsumed_s,
                pred, fam,varb,id,root,data=redata)
```

```
summary(model10)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + Deer:nn5Dist_s + Pollination:nn5DistNotConsumed_s,
##     pred = pred, fam = fam, varvar = varb, idvar = id, root = root,
##     data = redata)
##
##                                    Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt                    -1.137e+00  1.703e-01   -6.675 2.48e-11 ***
## varbembryoCt                     -5.809e-01  1.190e-02  -48.805  < 2e-16 ***
## varbflCt                         -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed              -1.526e-01  1.404e-01   -1.086    0.277
## varbflCtUndamaged                 1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested                  -3.213e+02  1.855e+00 -173.245  < 2e-16 ***
## varbovuleCt                       6.024e+00  7.148e-03  842.835  < 2e-16 ***
## Deer:nn5Dist_s                    4.441e-01  4.721e-01    0.941    0.347
## Pollination:nn5DistNotConsumed_s  1.026e-02  1.699e-03    6.043 1.51e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 11: $Fitness : nnA + Deer : nnA + Pollination : nnB$

```
model11 <- aster(resp ~ -1 + varb + fit:nn5Dist_s +Deer:nn5Dist_s + Pollination:nn5DistNotConsumed_s,
                 pred, fam,varb,id,root,data=redata)

summary(model11)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s +
##     Pollination:nn5DistNotConsumed_s, pred = pred, fam = fam,
##     varvar = varb, idvar = id, root = root, data = redata)
##
##                                    Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt                    -1.139e+00  1.703e-01   -6.684 2.33e-11 ***
## varbembryoCt                     -5.684e-01  1.197e-02  -47.489  < 2e-16 ***
## varbflCt                         -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed              -3.673e-01  1.464e-01   -2.509  0.01209 *
## varbflCtUndamaged                 1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested                  -3.211e+02  1.855e+00 -173.156  < 2e-16 ***
## varbovuleCt                       6.022e+00  7.162e-03  840.860  < 2e-16 ***
## fit:nn5Dist_s                    -7.797e-02  1.235e-02   -6.311 2.77e-10 ***
## nn5Dist_s:Deer                    1.536e+00  4.919e-01    3.123  0.00179 **
## Pollination:nn5DistNotConsumed_s  1.952e-02  2.114e-03    9.234  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model Selection

```
aster_AIC <- function(mod) {
  return(mod$deviance + 2*length(mod$coefficients))
```

```
}
```

```
## resp ~ -1 + varb
## [1] -249884.7
## resp ~ -1 + varb + fit:nn5Dist_s
## [1] -249882.8
## resp ~ -1 + varb + Deer:nn5Dist_s
## [1] -249891.2
## resp ~ -1 + varb + Pollination:nn5Dist_s
## [1] -249886.4
## resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s
## [1] -249890.4
## resp ~ -1 + varb + fit:nn5Dist_s + Pollination:nn5Dist_s
## [1] -250315.4
## resp ~ -1 + varb + Deer:nn5Dist_s + Pollination:nn5Dist_s
## [1] -249889.7
## resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s
## [1] -250317.1
## resp ~ -1 + varb + Pollination:nn5DistNotConsumed_s
## [1] -249928
## resp ~ -1 + varb + fit:nn5Dist_s + Pollination:nn5DistNotConsumed_s
## [1] -249957.7
## resp ~ -1 + varb + Deer:nn5Dist_s + Pollination:nn5DistNotConsumed_s
## [1] -249926.9
## resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5DistNotConsumed_s
## [1] -249965.3
```

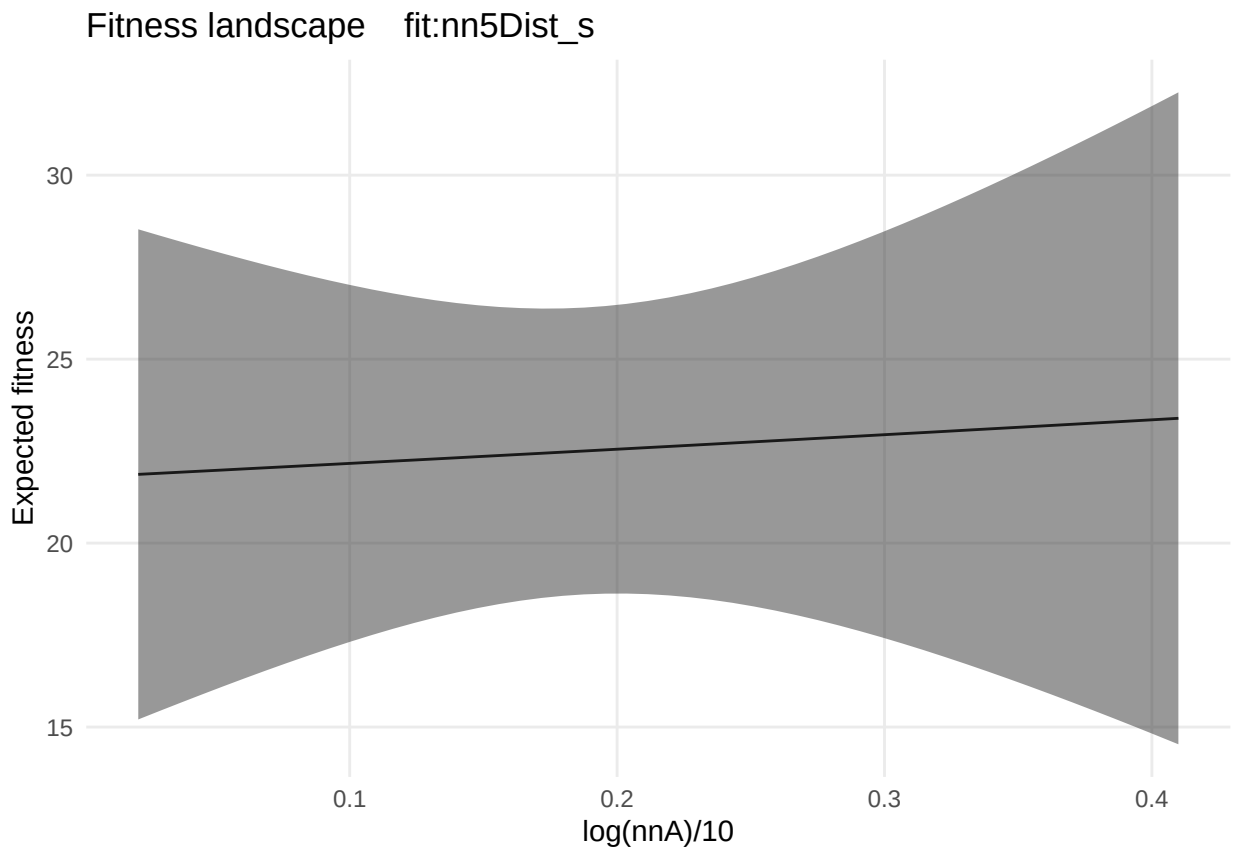
1-D Fitness landscape

```
fitness_landscape <- function(model, covariate = 'nn5Dist_s', lower = NULL, upper = NULL,
                              observation=FALSE, scale_back = FALSE) {
  # Make fake individuals
  nInd <- 50
  lwr <- if (is.null(lower)) min(unique(model$data[,covariate])) else lower
  upr <- if (is.null(upper)) max(unique(model$data[,covariate])) else upper
  cand.nnA <- seq(from = lwr, to = upr, length = nInd)
  cand <- as.data.frame(cand.nnA)
  colnames(cand) <- covariate
  cand$root <- 1
  blah <- data[1:nInd, colnames(data) %in% vars]
  cand <- cbind(cand, blah)
```

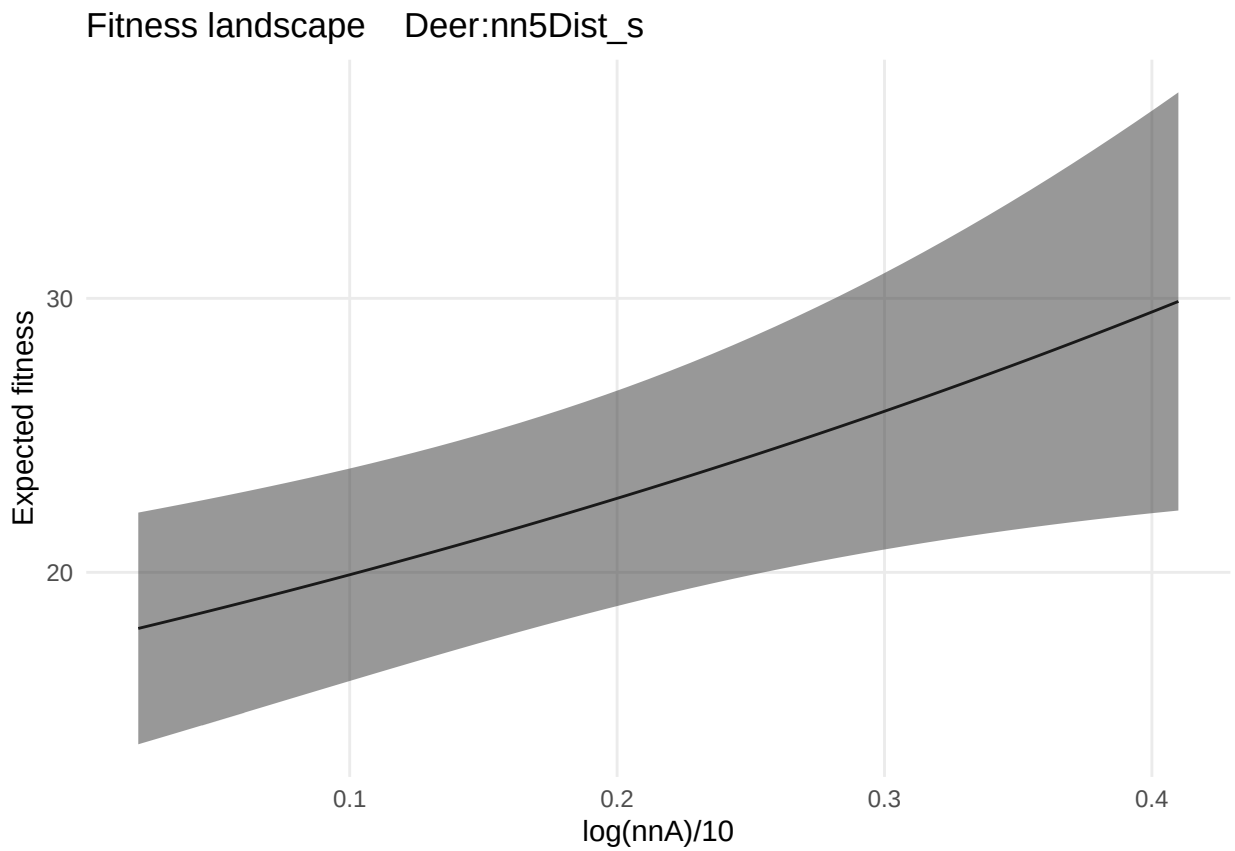
```r
cand$id <- data[1:nInd, 'id']

# Transform fake data into long format
cand_long <- reshape(cand, varying = list(vars), direction="long", timevar="varb",
                times = as.factor(vars), v.names="resp")

cand_long <- data.frame(cand_long)
cand_long$fit <- as.numeric(cand_long$varb == "embryoCt")
cand_long$Nid <- as.numeric(gsub("[^0-9.-]", "", cand_long$id))

cand_long$Deer <- as.numeric(cand_long$varb=='flCtNotConsumed')
cand_long$Pollination <- as.numeric(is.element(cand_long$varb,
                                c("capsuleCt", "isHarvested", "ovuleCt", "embryoCt")))
#cand_long$Rtail <- as.numeric(cand_long$nn5Dist_s > quantile(cand_long$nn5Dist_s, 0.975))
#cand_long$Ltail <- as.numeric(cand_long$nn5Dist_s < quantile(cand_long$nn5Dist_s, 0.025))

# Get conditional mean value parameters
pred <- predict(model, cand_long, varvar=varb, idvar=id, root=root,
                    se.fit = TRUE, model.type='conditional',
                is.always.parameter = TRUE, info.tol=1e-8)

xi_parm <- pred$fit
xi_parm_se <- pred$se.fit

names(xi_parm) <- paste0(cand_long$id,'.',cand_long$varb)
names(xi_parm_se) <- paste0(cand_long$id,'.',cand_long$varb)

xi_parm_grad <- pred$gradient
rownames(xi_parm_grad) <- paste0(cand_long$id,'.',cand_long$varb)
colnames(xi_parm_grad) <- names(model$coefficients)

# Expected fitness
Ids <- unique(cand_long$id)
exp_fitness <- rep(0,nInd)
names(exp_fitness) <- Ids
for (id in Ids) {
  exp_fitness[id] <- prod(xi_parm[grep(paste0(id,'\\.'), names(xi_parm))][-5])
}

# Get covariance matrix of xi
xi_parm_var = xi_parm_grad %*% solve(model$fisher) %*% t(xi_parm_grad)

# Delta Method
var_expfit <- rep(0,nInd)
names(var_expfit) <- Ids
for (id in Ids) {
  ind <- paste0(id,'\\.')
  xi <- xi_parm[grep(ind, names(xi_parm))][-5]
  var <- xi_parm_var[grep(ind, rownames(xi_parm_var))[-5],
                     grep(ind, colnames(xi_parm_var))[-5]]
  grad <- c(prod(xi[-1]), prod(xi[-2]), prod(xi[-3]), prod(xi[-4]), prod(xi[-5]), prod(xi[-6]))
  var_expfit[id] <- t(grad) %*% var %*% grad
}
```

```
  se_expfit = sqrt(var_expfit)
  xlabel = if (covariate == 'nn5Dist_s') 'log(nnA)/10' else 'log(nnB)/10'
  cand_block <- cand %>% mutate(exp_fitness, exp_fitness, lower = exp_fitness - 2 * se_expfit, upper = e
  if (scale_back == TRUE) {
    cand_block[, as.character(covariate)] <- exp(10*cand_block[,as.character(covariate)])
    xlabel = if (covariate == 'nn5Dist_s') 'nnA' else 'nnB'
  }
  #cand_block <- cand_block[1:40,]
  plt <- ggplot(data = cand_block) + geom_line(mapping = aes(x = cand_block[,as.character(covariate)], y
  if (observation == TRUE) {
    obs <- test %>% filter((!!sym(covariate)) > lwr & (!!sym(covariate)) < upr)
    if (scale_back == TRUE) {
      plt <- plt + geom_point(data=obs, mapping = aes(x =exp(10*obs[,as.character(covariate)]),
                                                      y = obs[,'fecundity']))
    } else {
      plt <- plt + geom_point(data=obs, mapping = aes(x =obs[,as.character(covariate)], y = obs[,'fecun
    }
  }
  plt <- plt + labs(x=xlabel, y="Expected fitness", title = paste0("Fitness landscape    ", substr(forma
    #annotate(geom='text', x=c(0, 0.1, 0.3, 0.5), y=750, label=c('nnA', '2.718282', '20.085537', '148.4
    theme_minimal() + scale_x_continuous(minor_breaks = NULL) +scale_y_continuous(minor_breaks = NULL)

  print(plt)

}
```

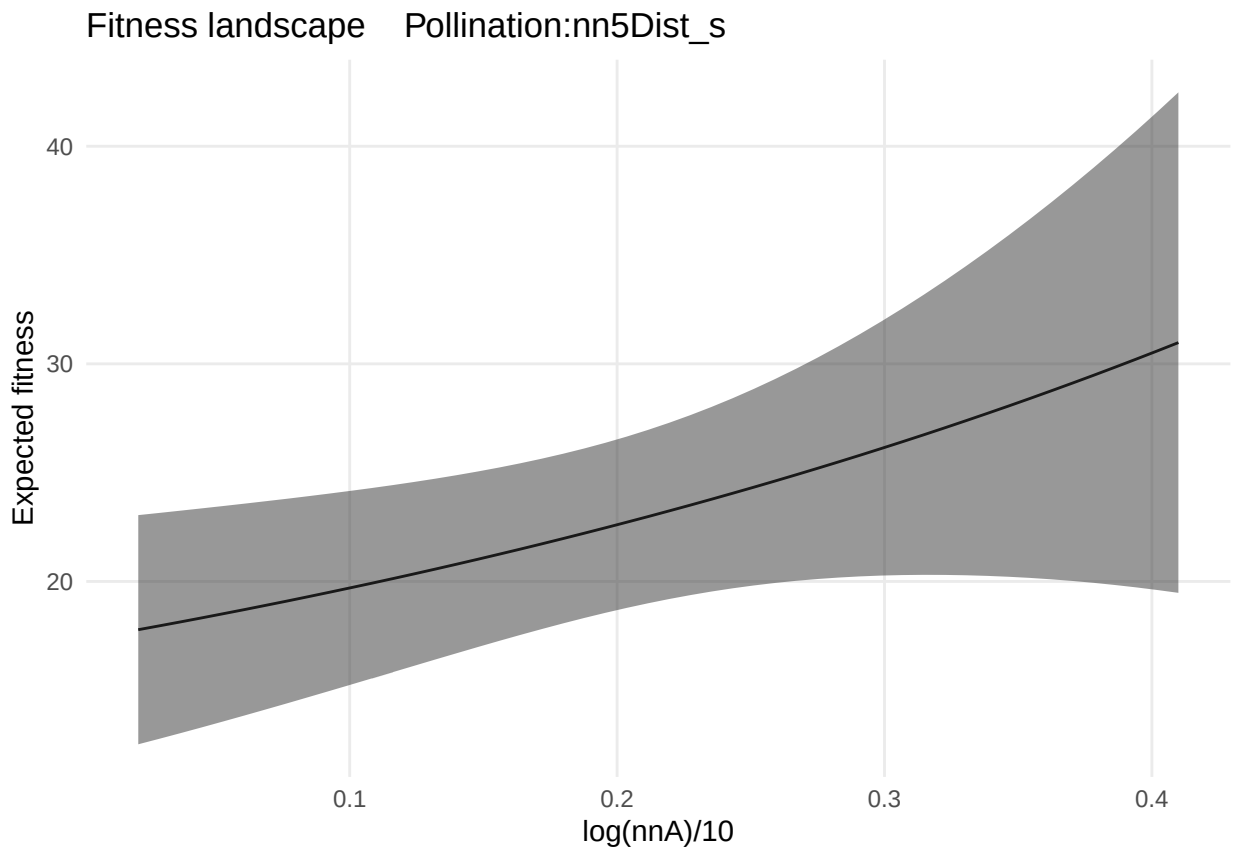
```
#par(mar = c(4, 4, .1, .1))
fitness_landscape(model1, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5Di
```



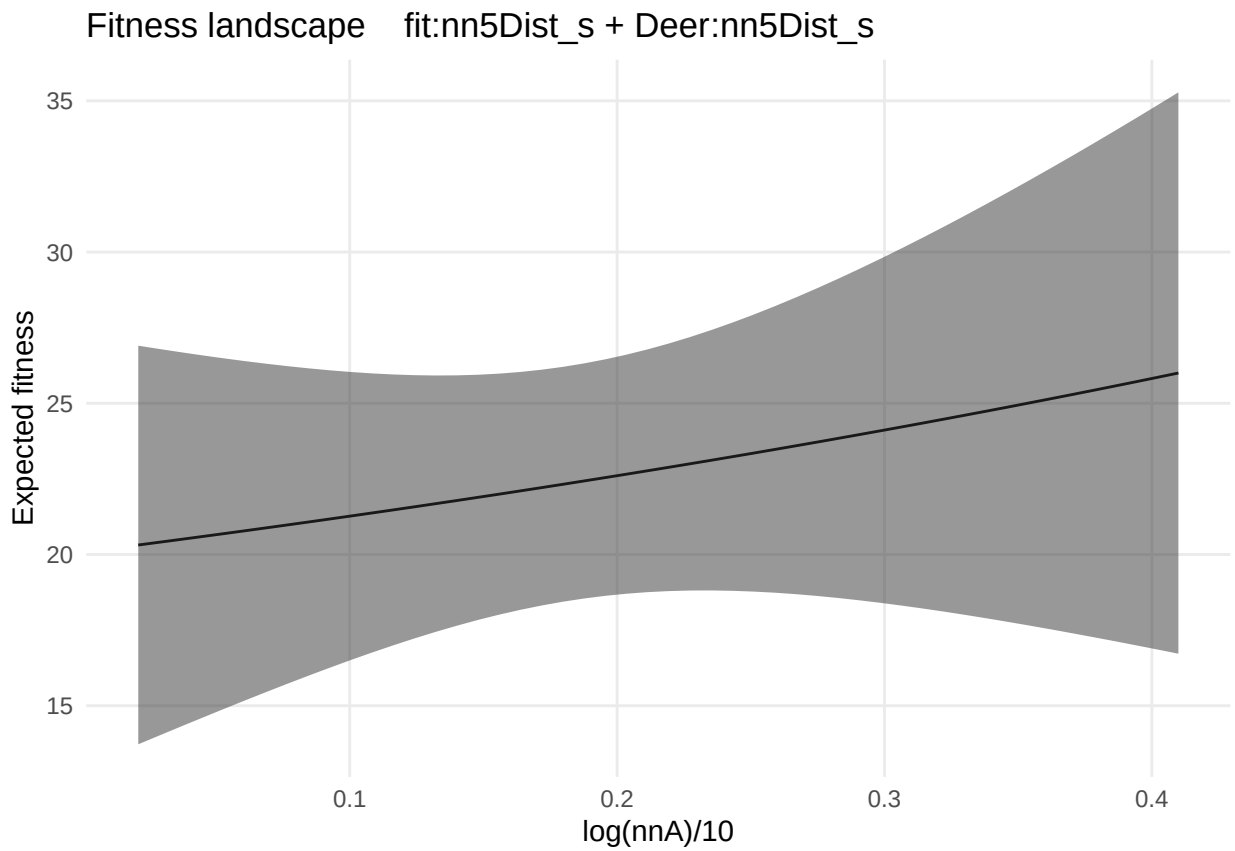

```
fitness_landscape(model2, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



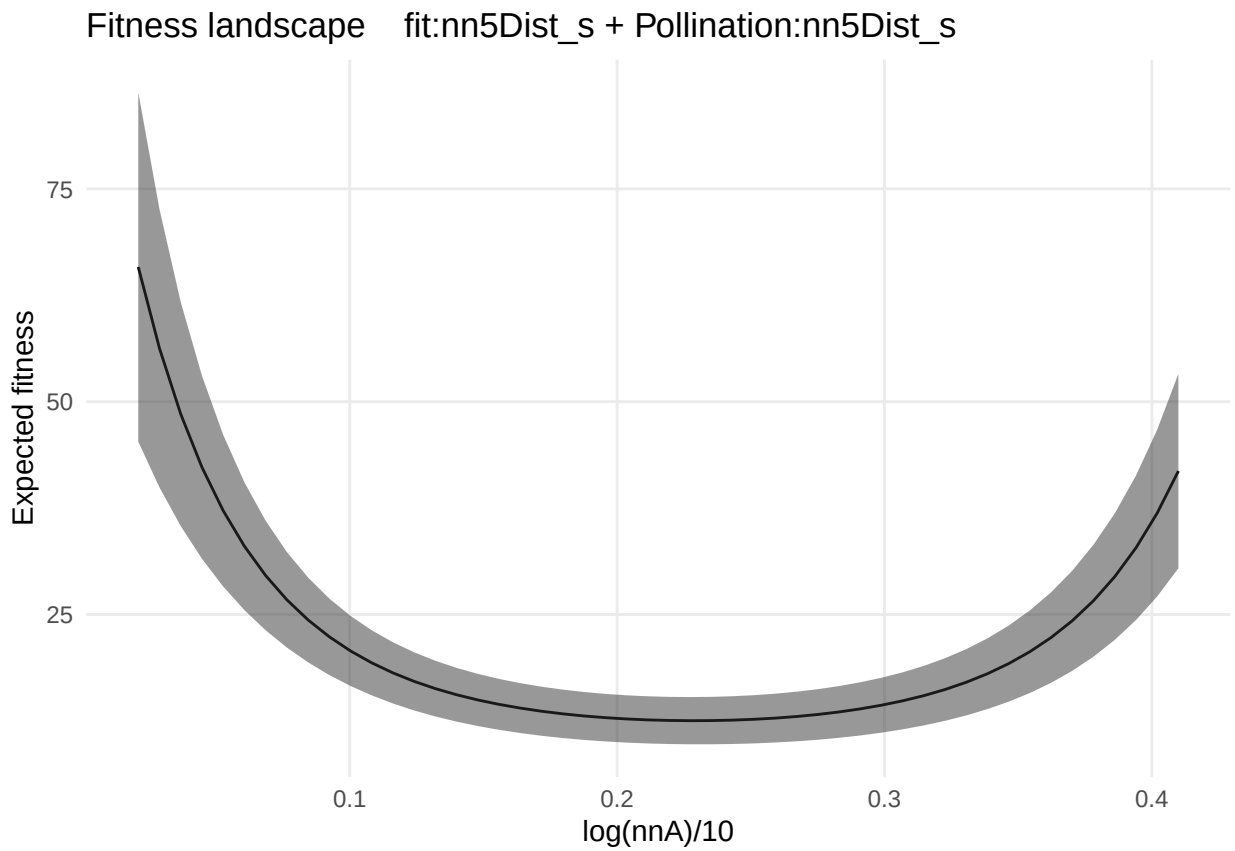

```
fitness_landscape(model3, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



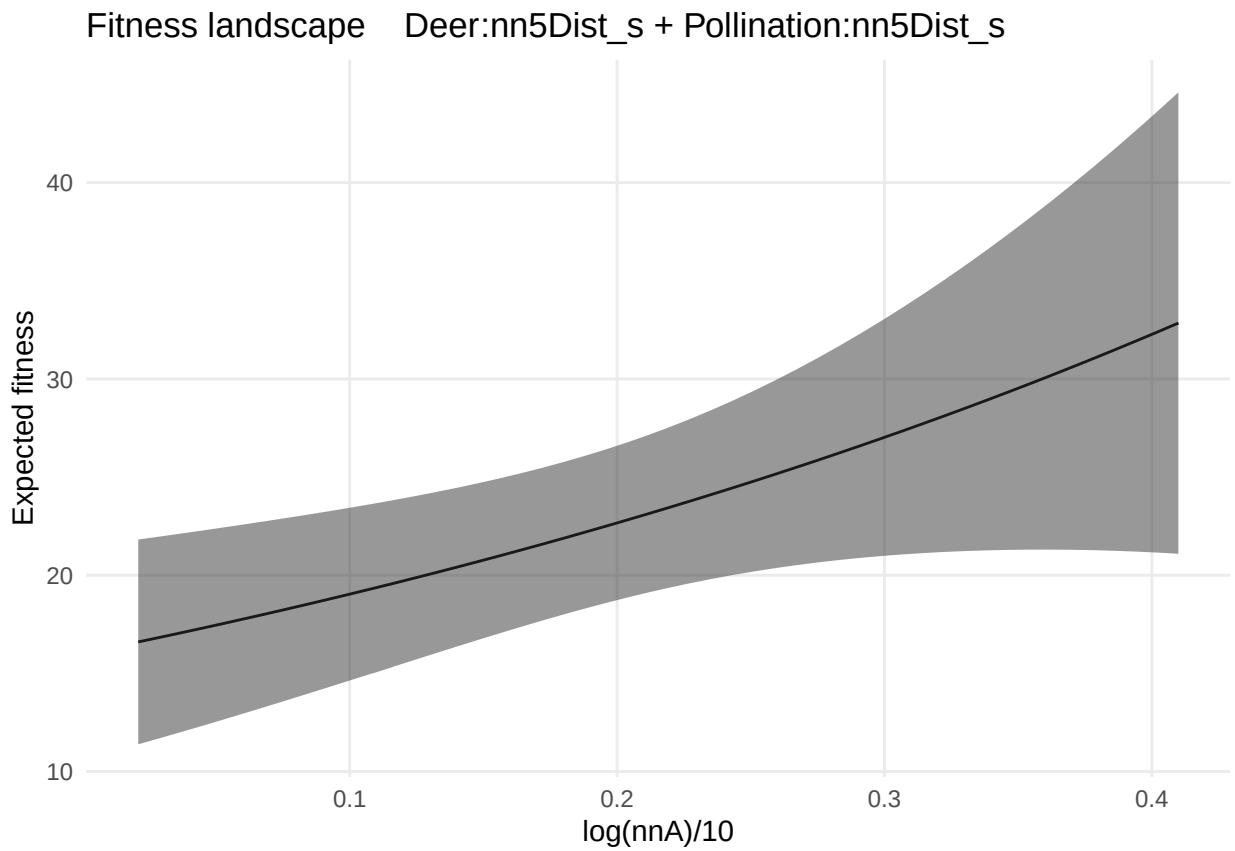

```
fitness_landscape(model4, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



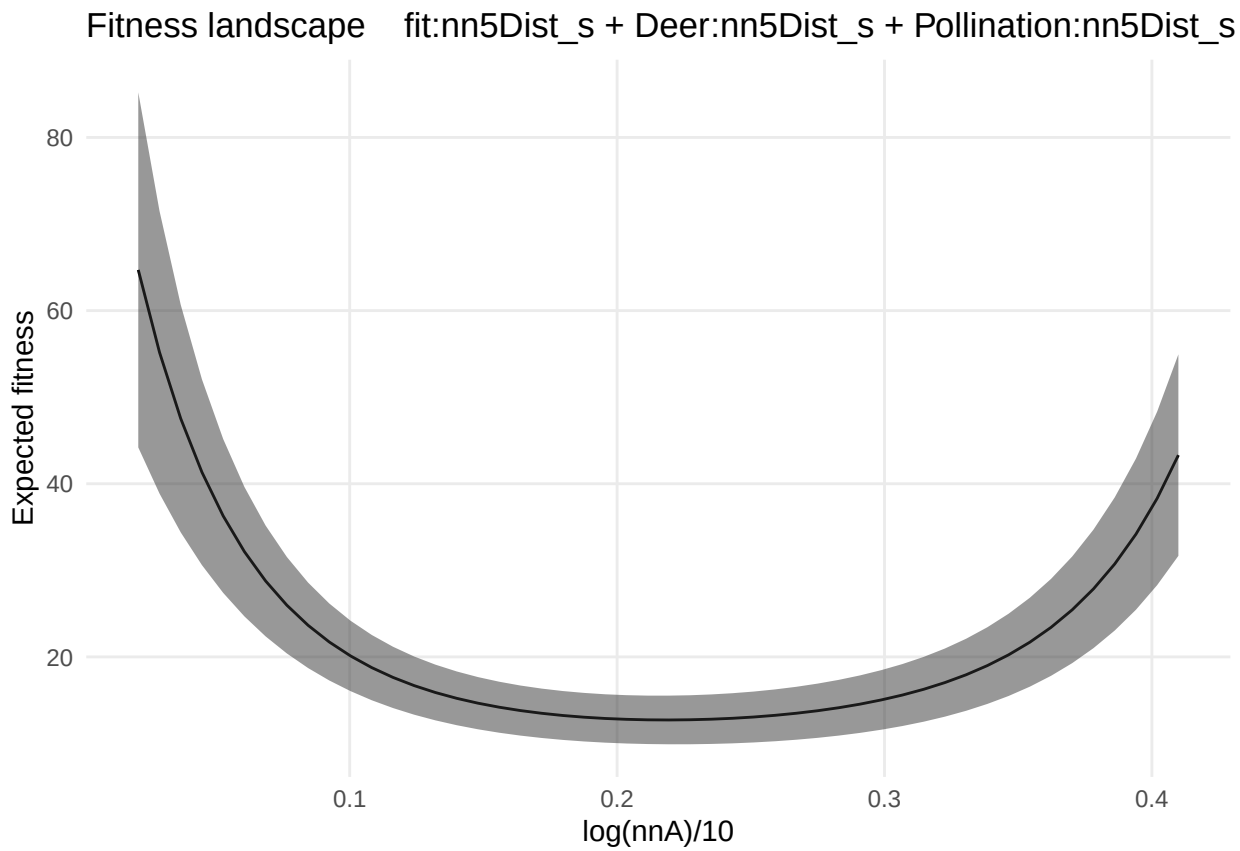

```
fitness_landscape(model5, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



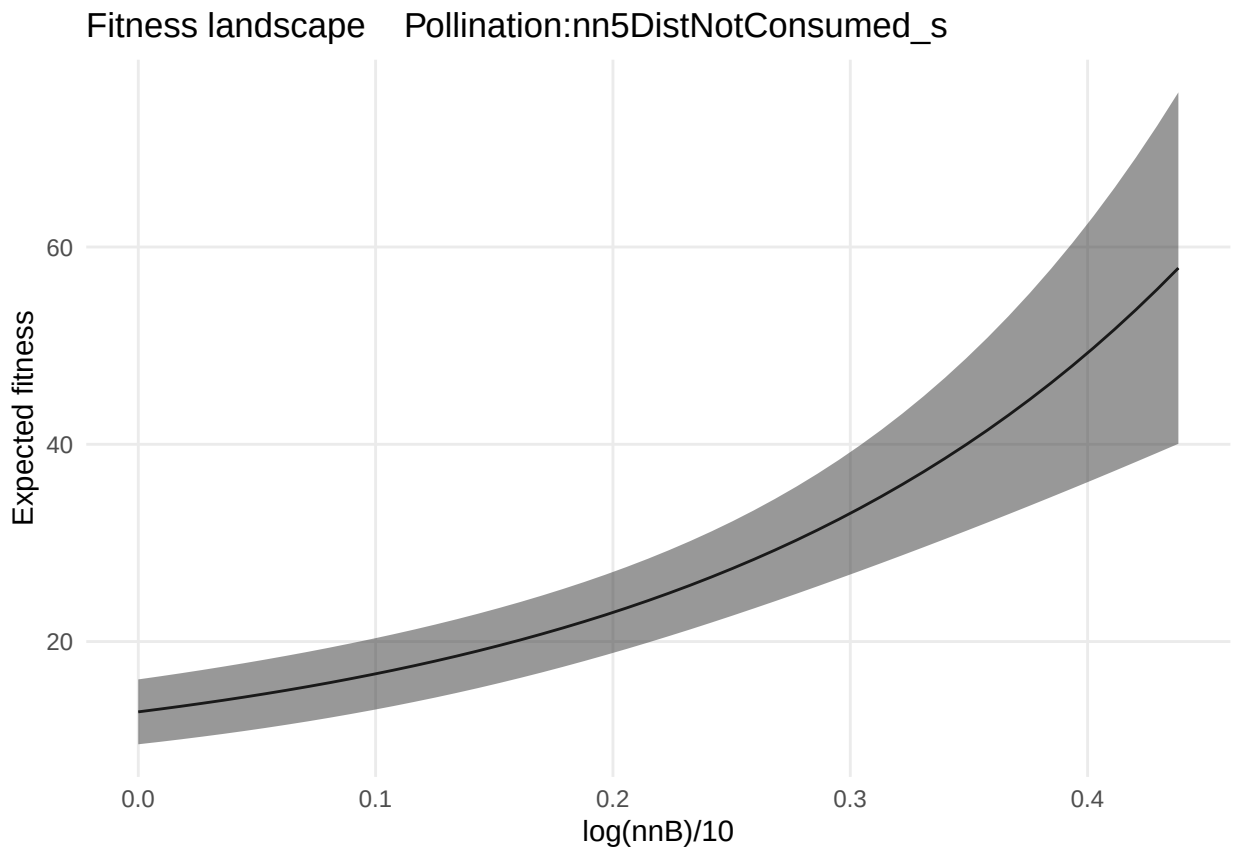

```
fitness_landscape(model6, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



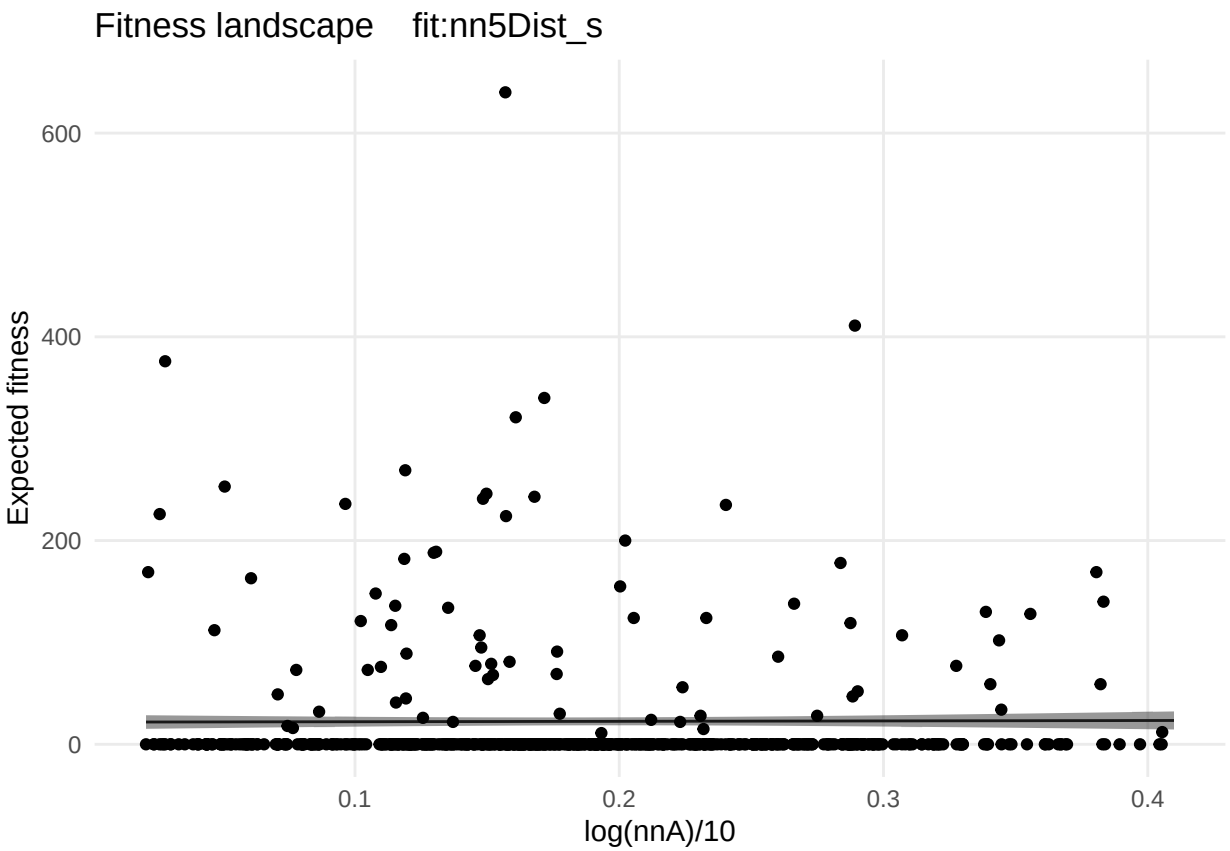

```
fitness_landscape(model7, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



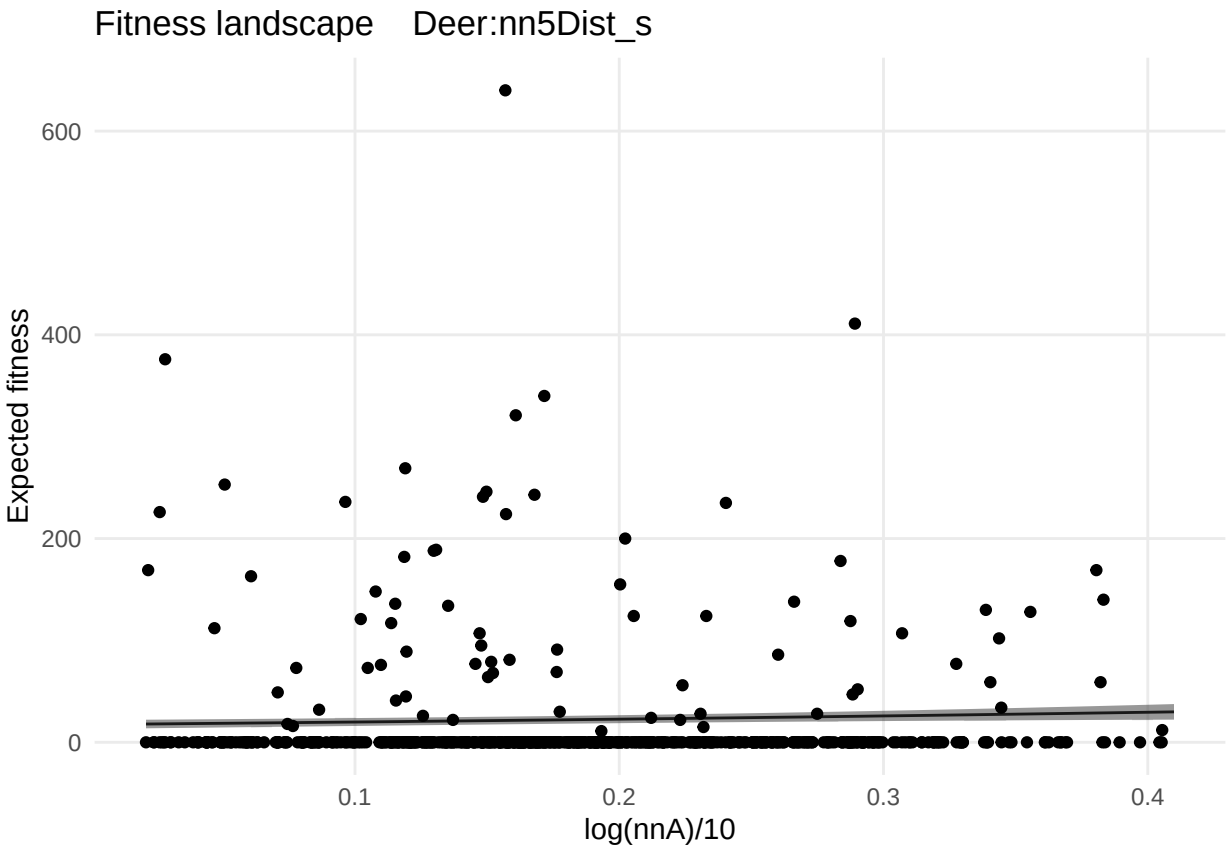

```
fitness_landscape(model8, covariate = 'nn5DistNotConsumed_s', lower = quantile(test$nn5DistNotConsumed_s
```



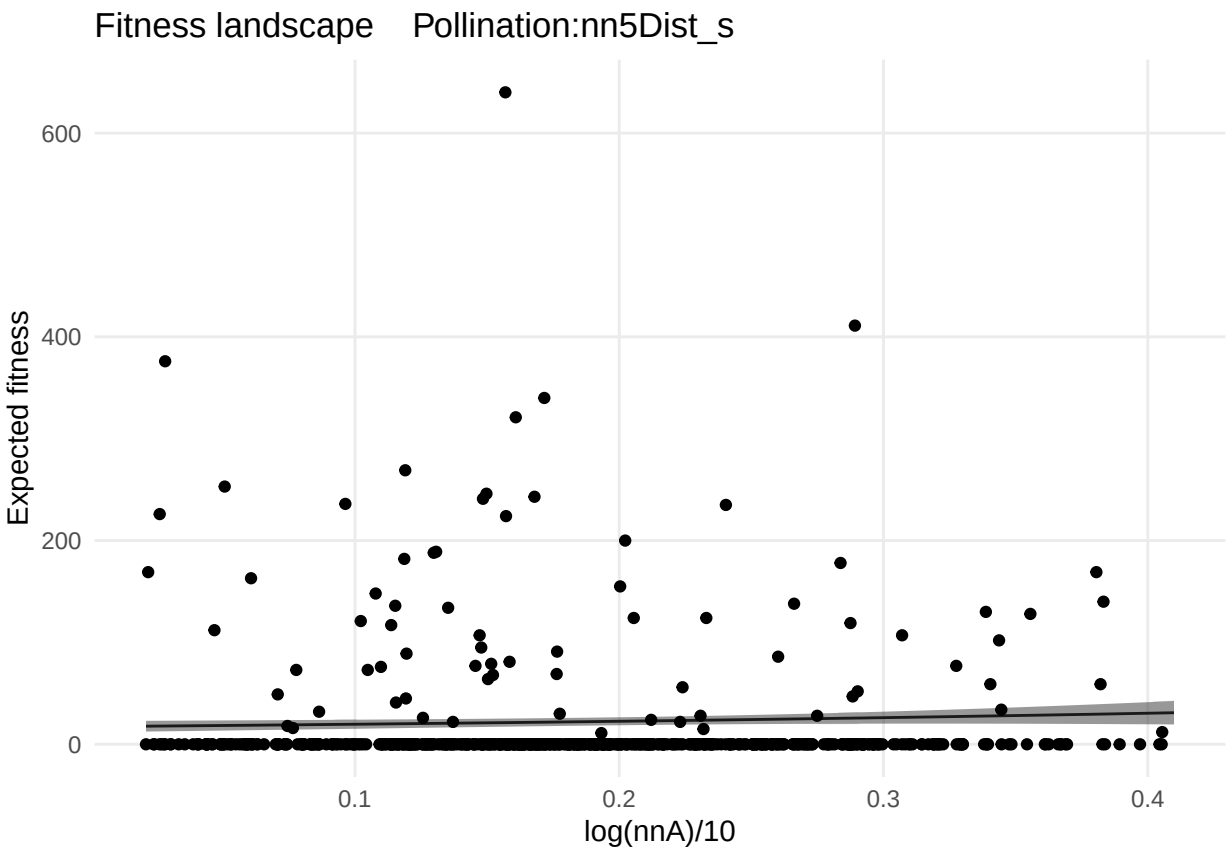

```
#par(mar = c(4, 4, .1, .1))
fitness_landscape(model1, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



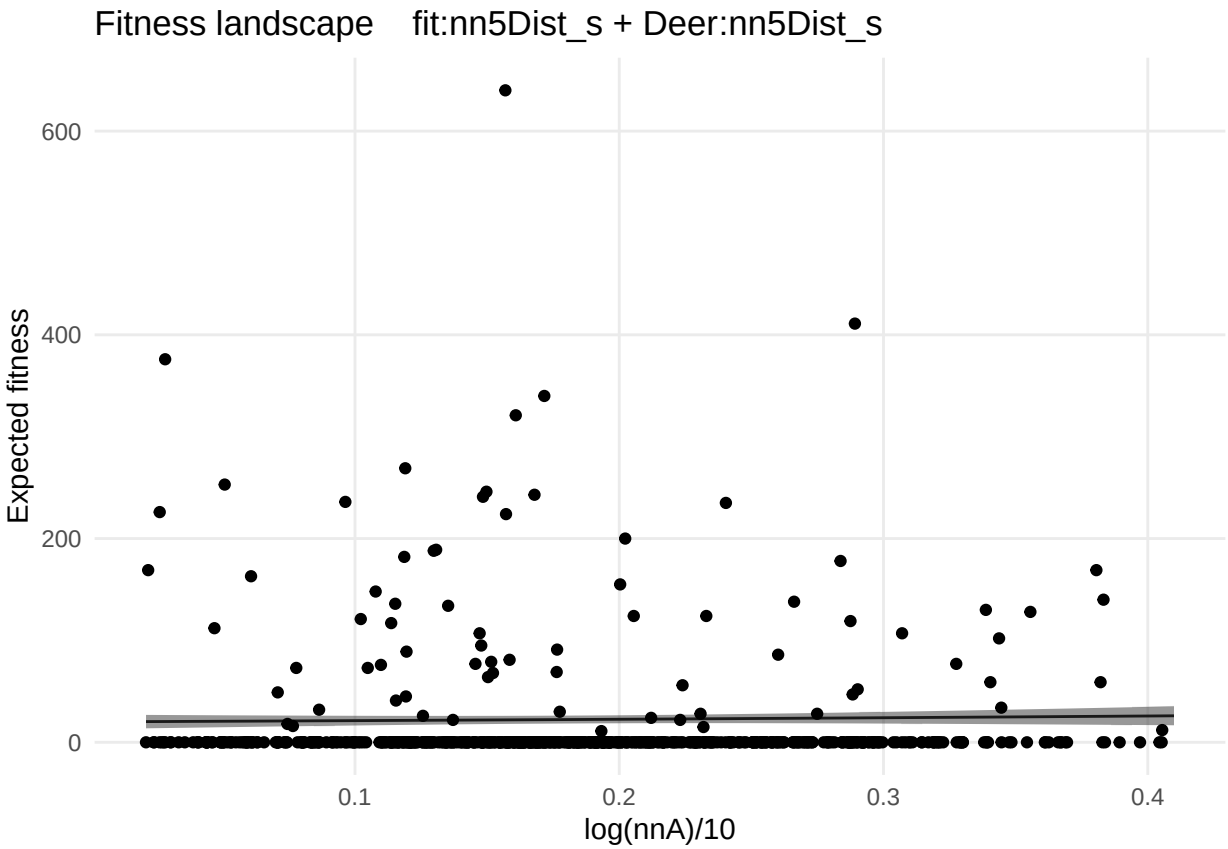

```
fitness_landscape(model2, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



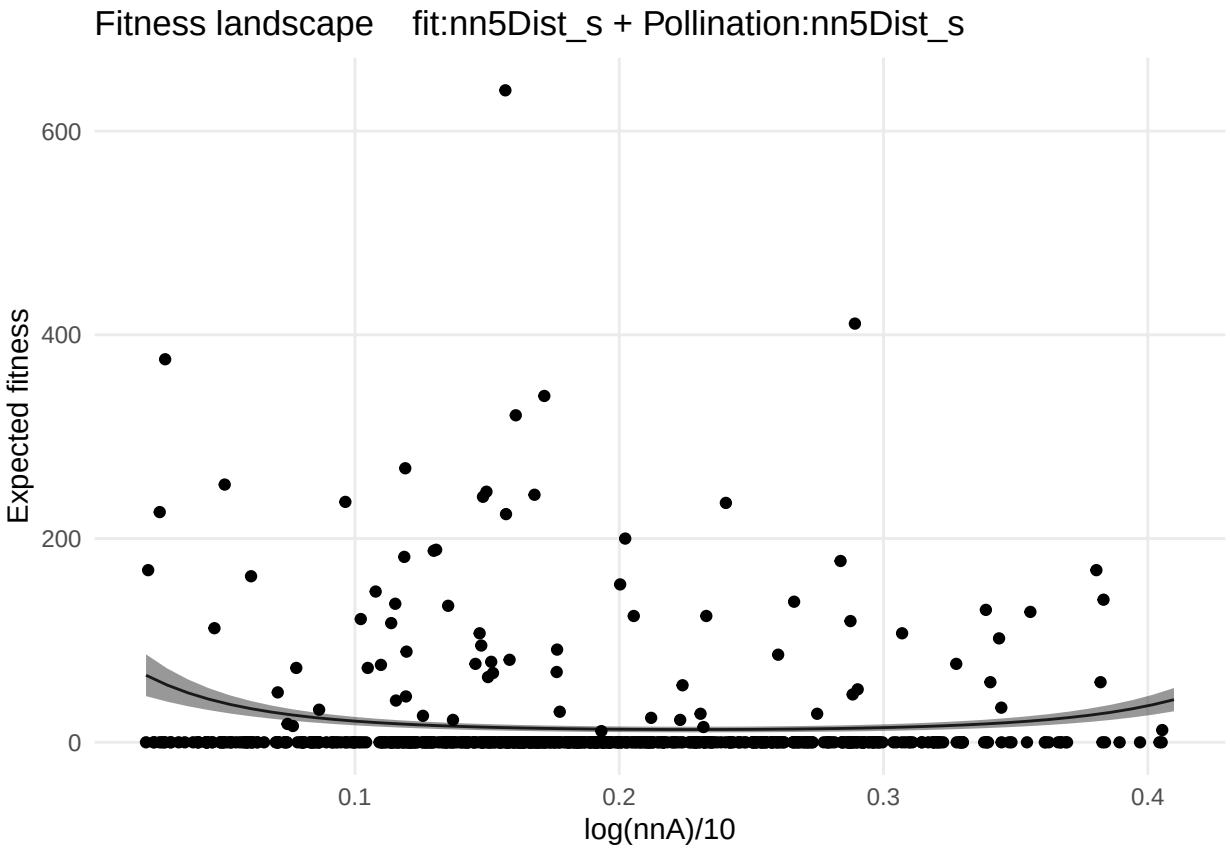

```
fitness_landscape(model3, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



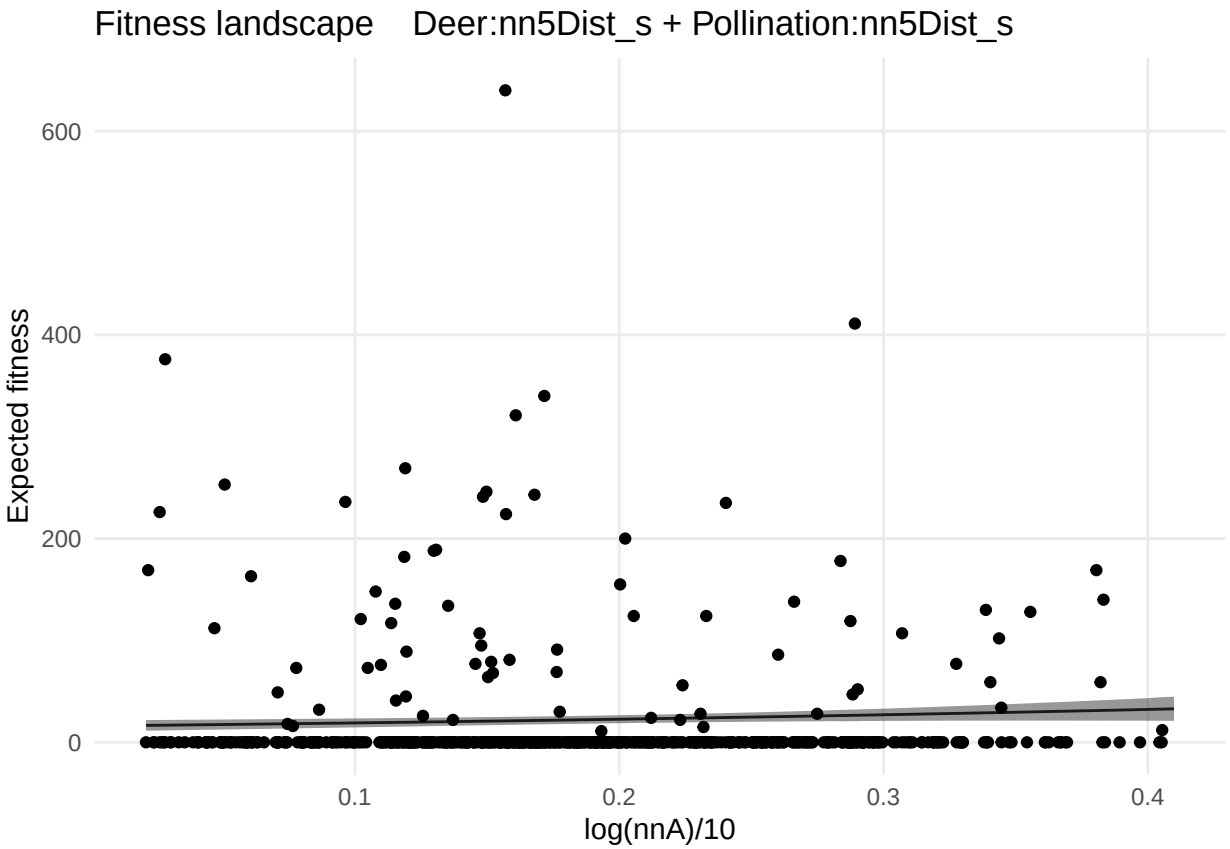

```
fitness_landscape(model4, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



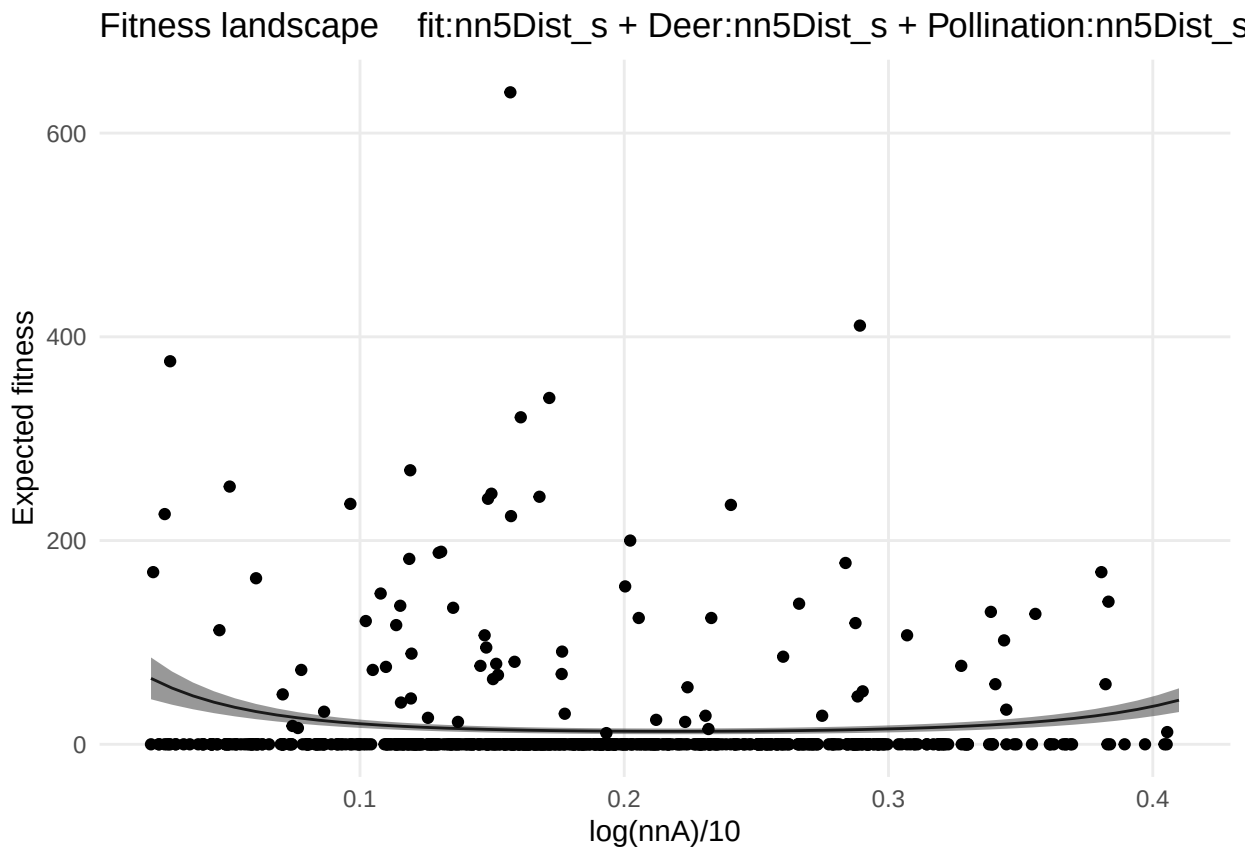

```
fitness_landscape(model5, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



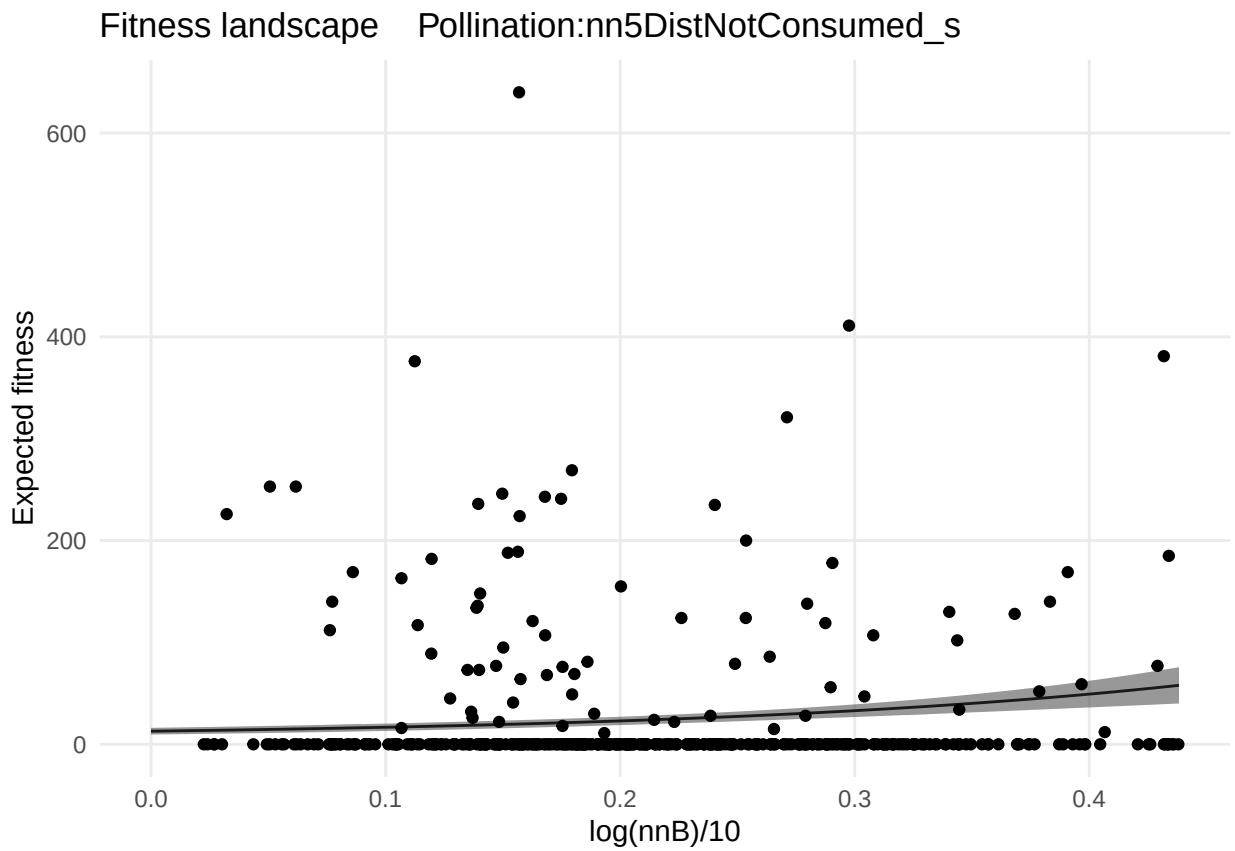

```
fitness_landscape(model6, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



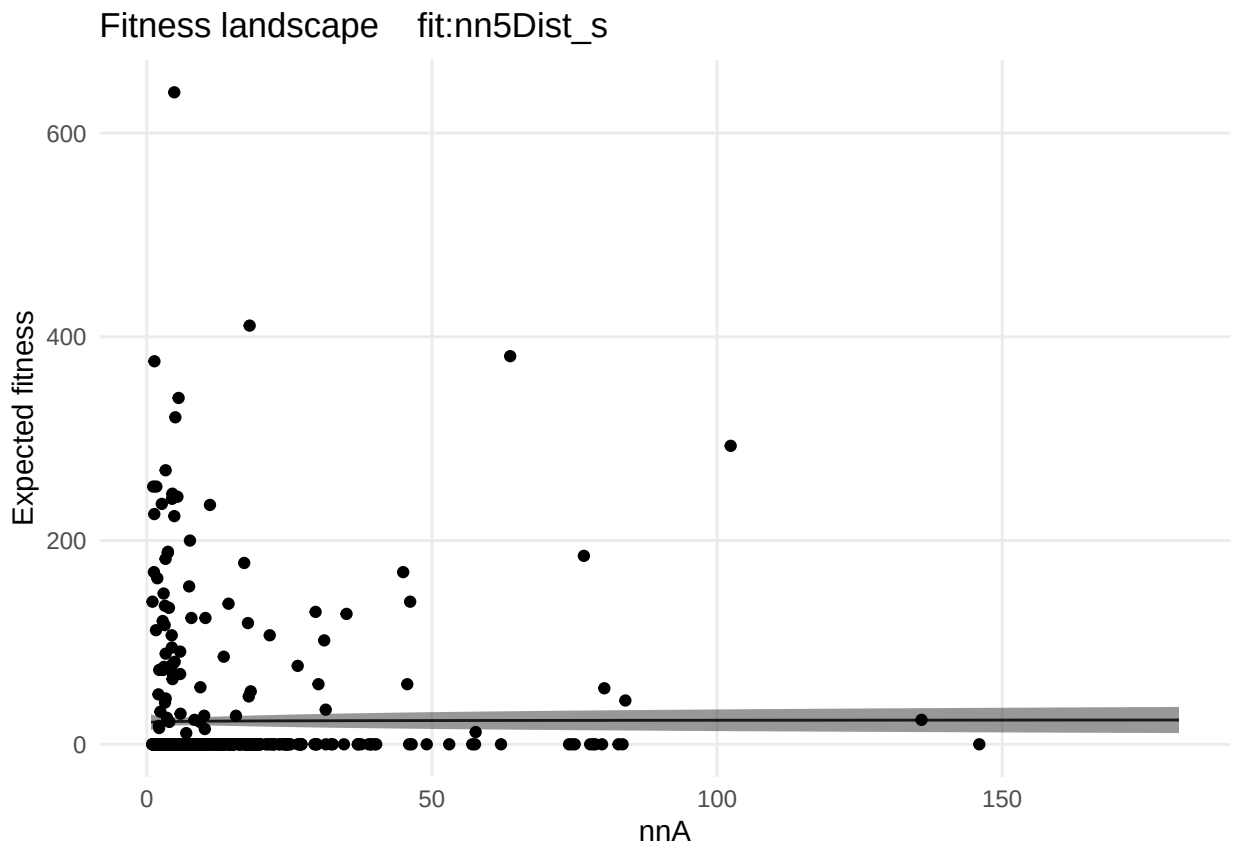

```
fitness_landscape(model7, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



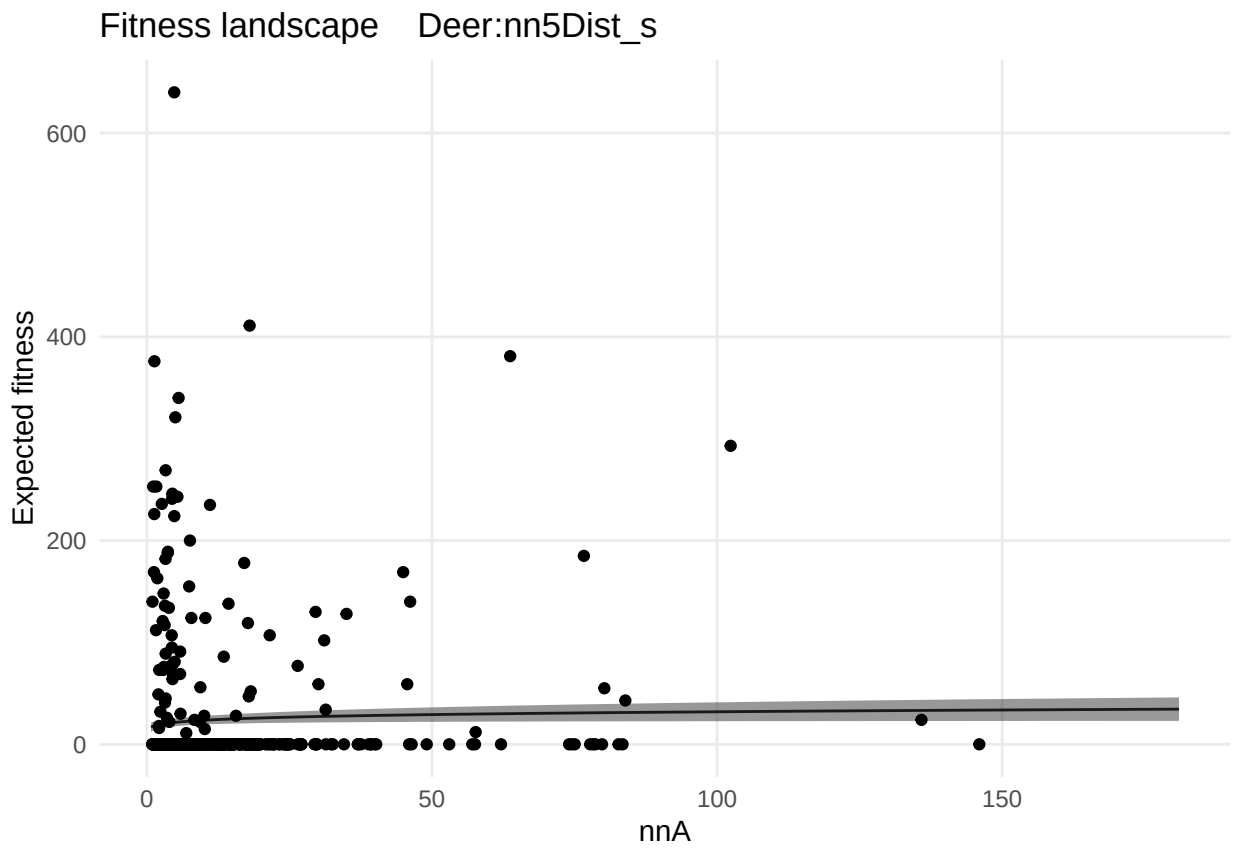

```
fitness_landscape(model8, covariate = 'nn5DistNotConsumed_s', lower = quantile(test$nn5DistNotConsumed_s
```



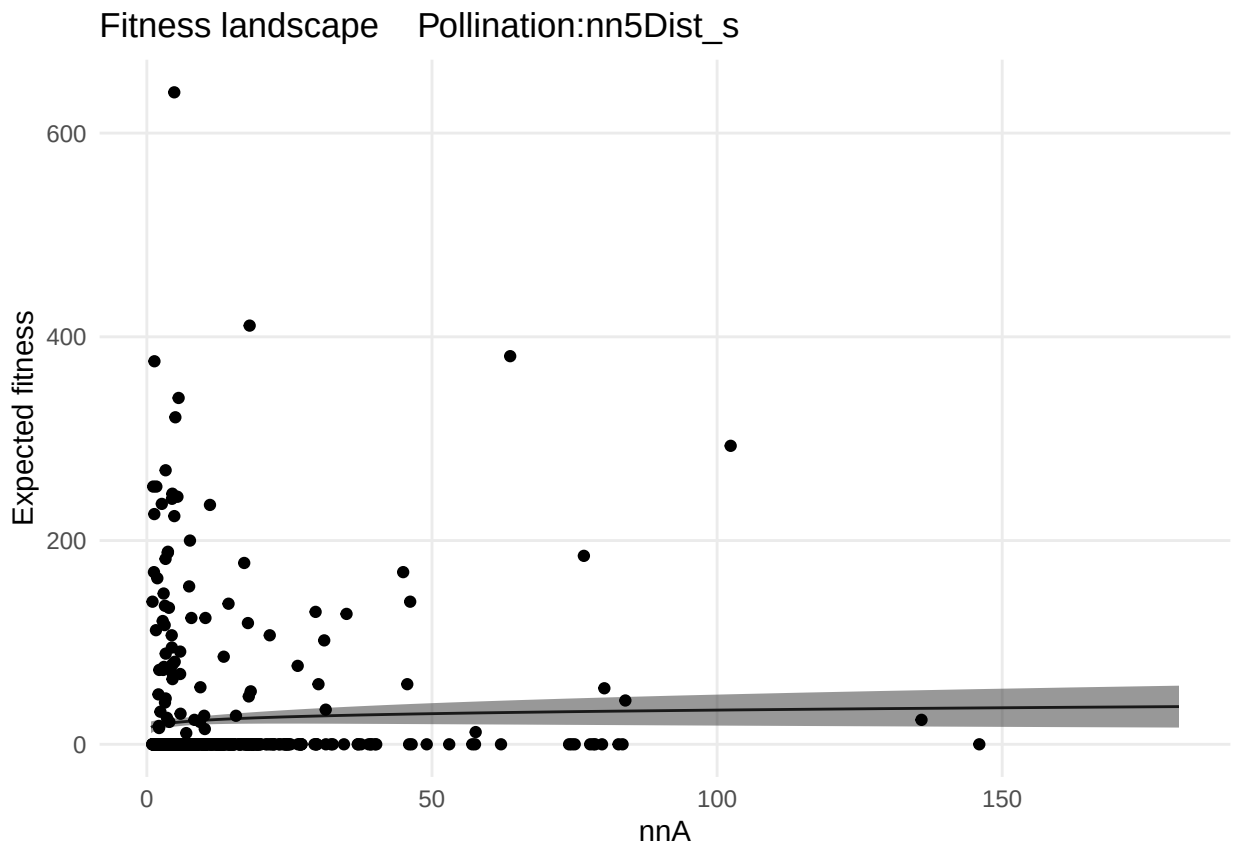

```
#par(mar = c(4, 4, .1, .1))
fitness_landscape(model1, observation = TRUE, scale_back = TRUE)
```



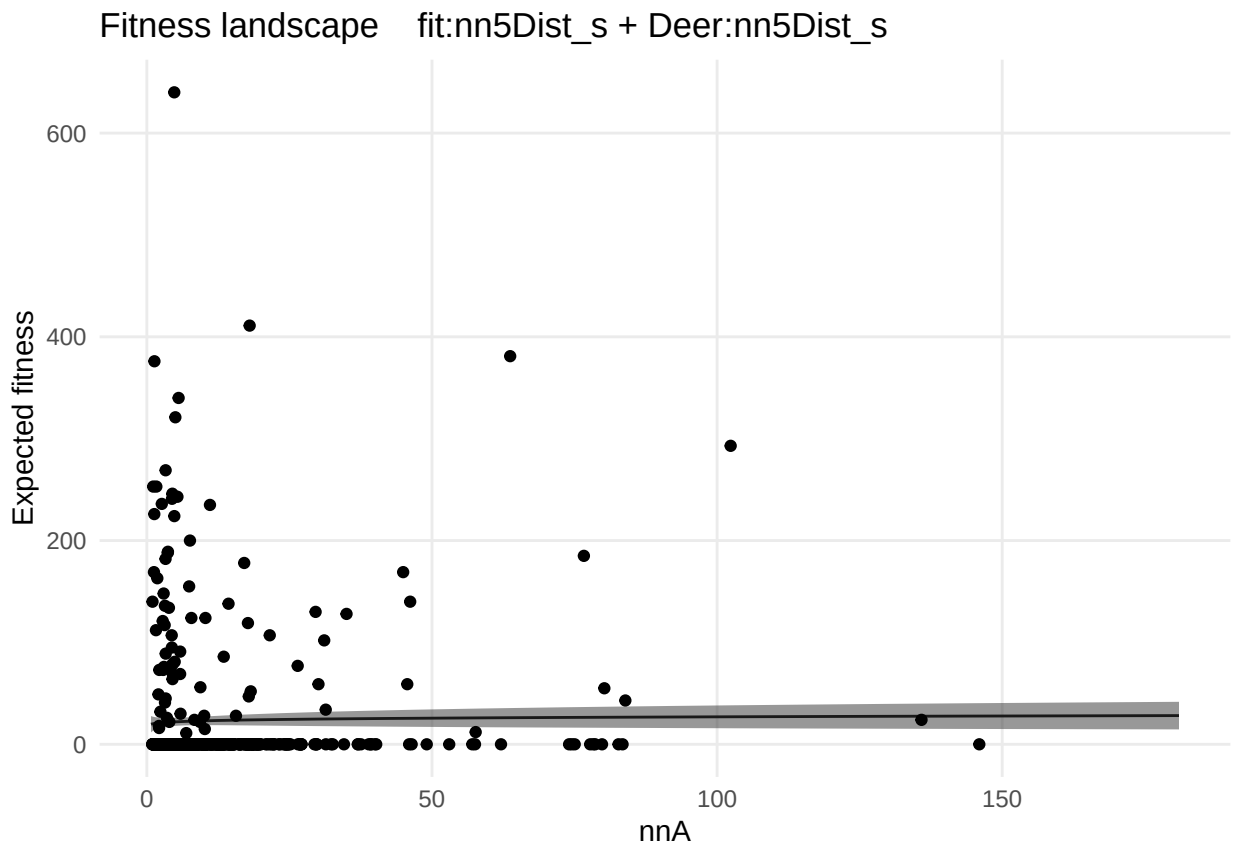

```
fitness_landscape(model2, observation = TRUE, scale_back = TRUE)
```



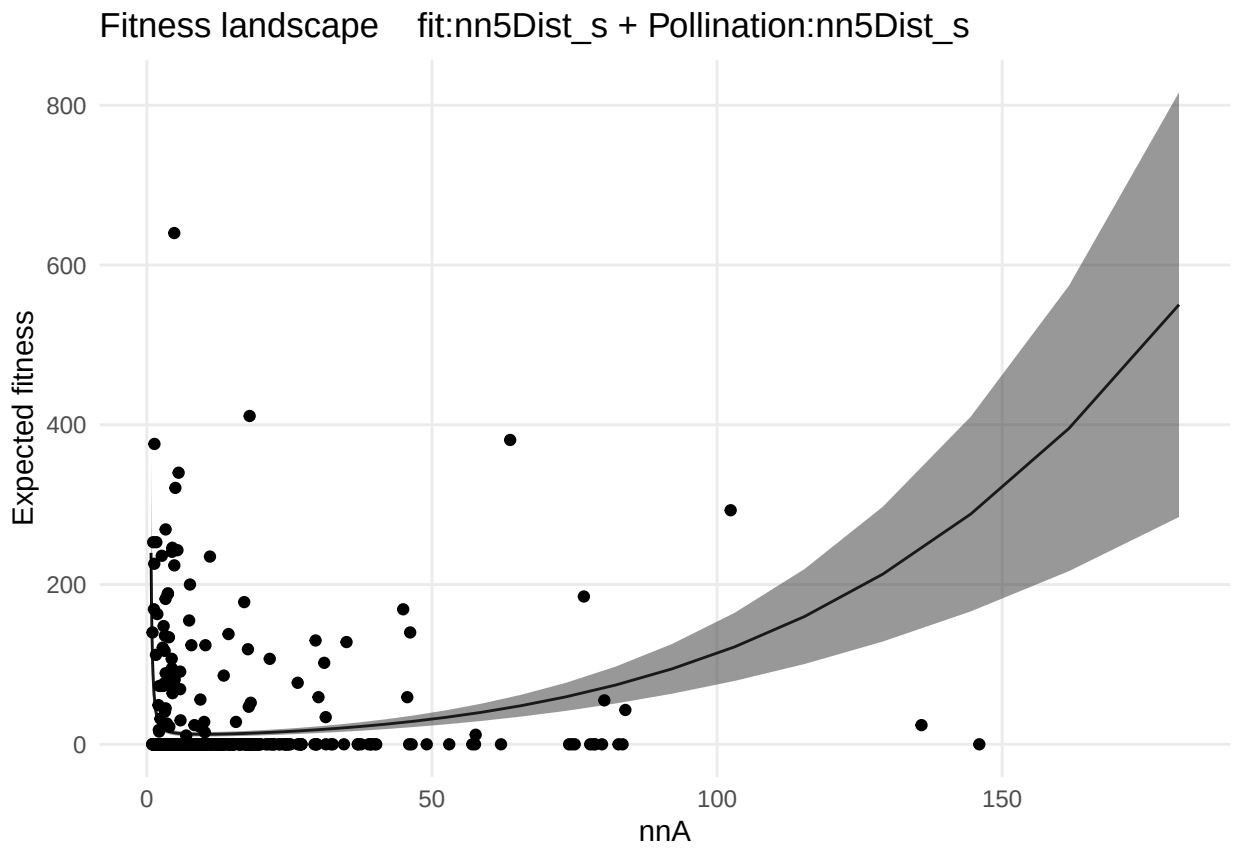

```
fitness_landscape(model3, observation = TRUE, scale_back = TRUE)
```



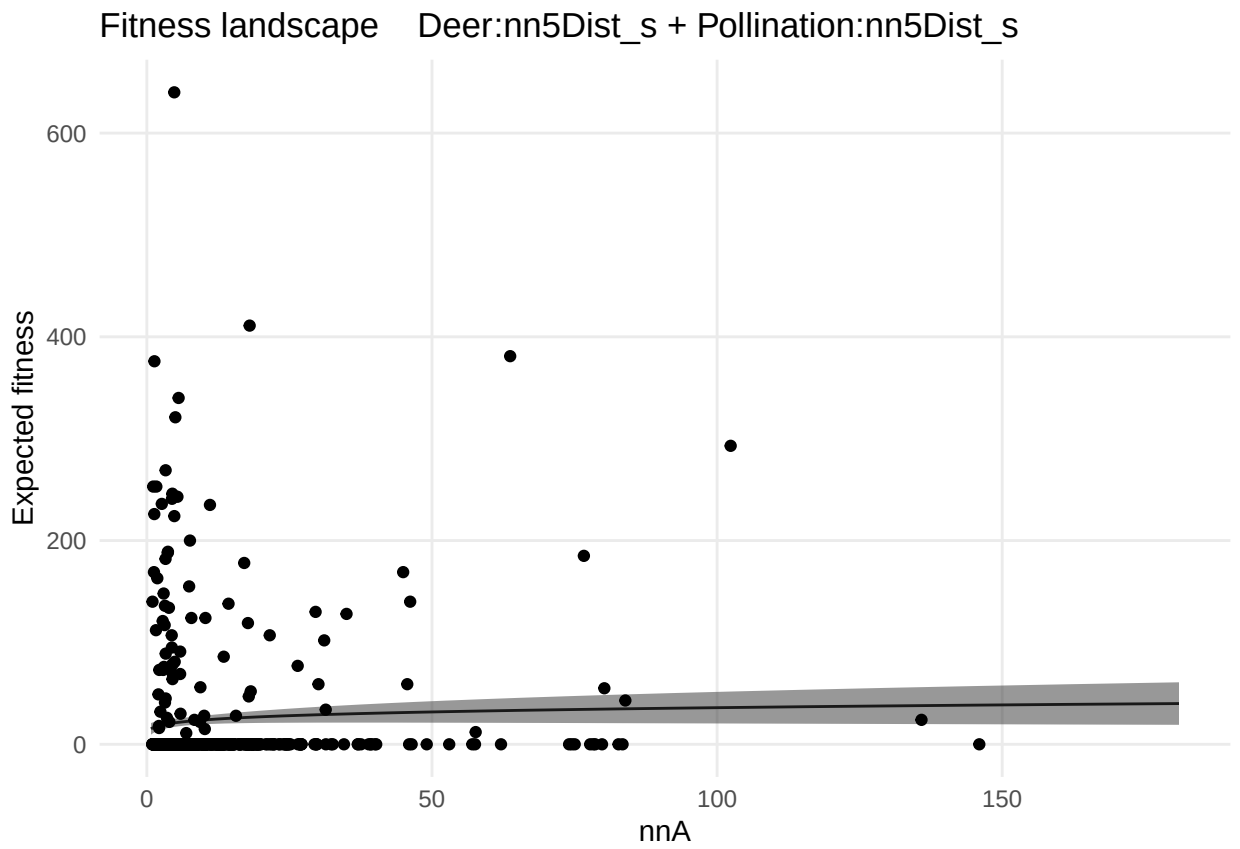

```
fitness_landscape(model4, observation = TRUE, scale_back = TRUE)
```



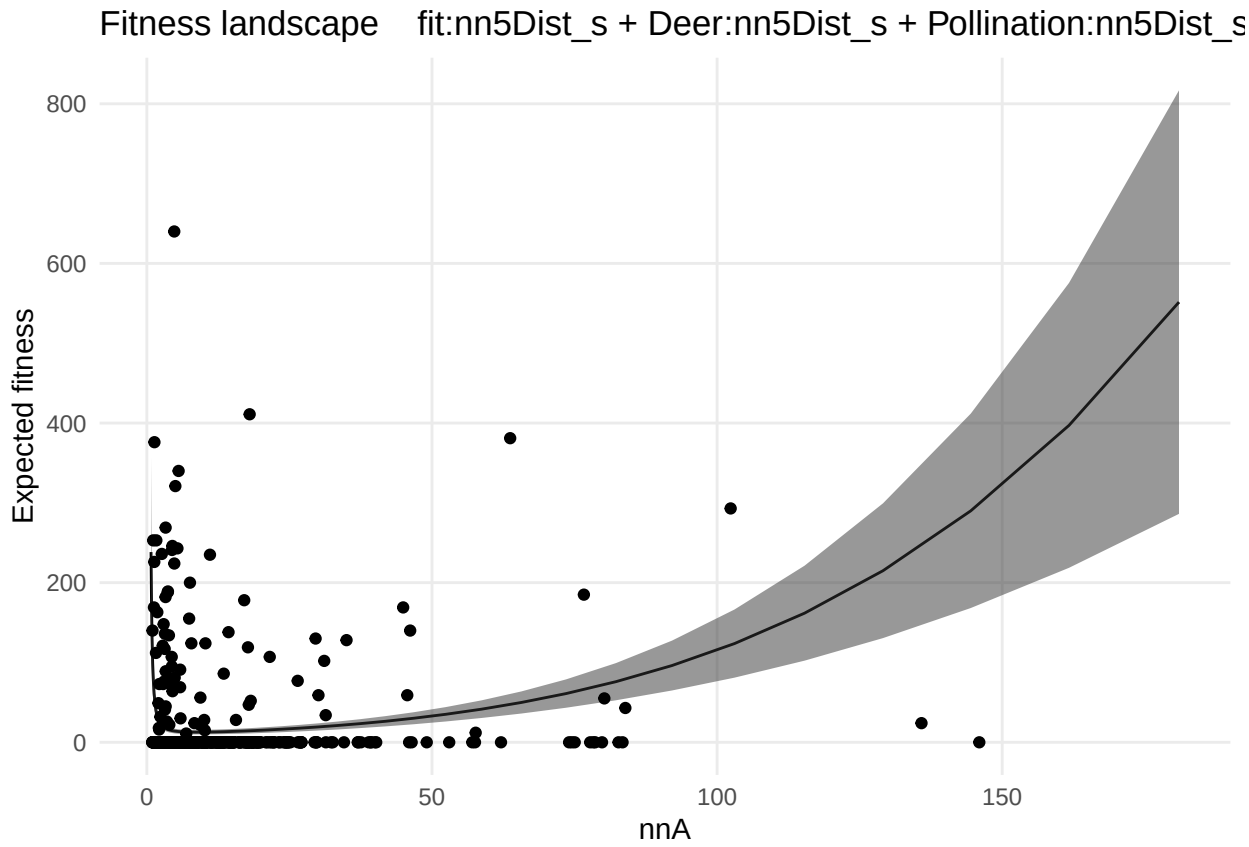

```
fitness_landscape(model5, observation = TRUE, scale_back = TRUE)
```



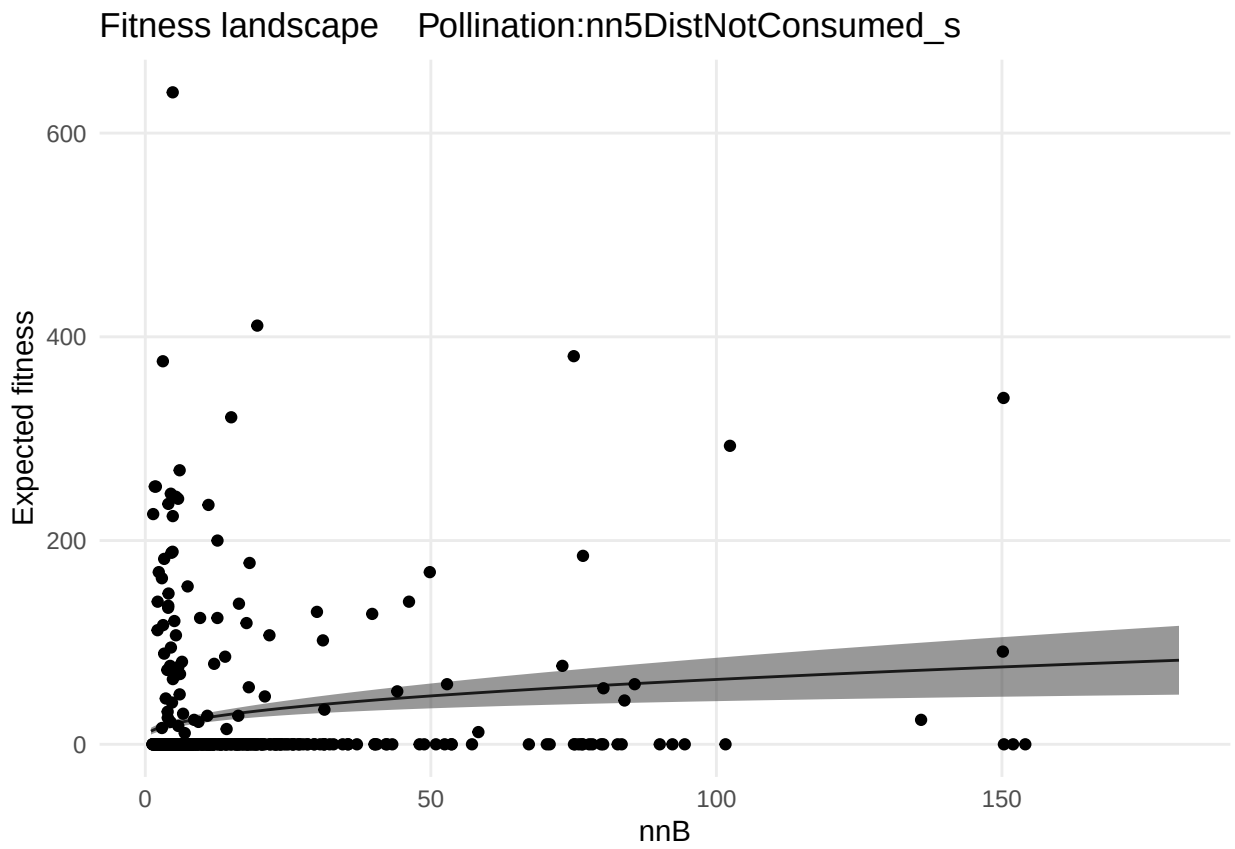

```
fitness_landscape(model6, observation = TRUE, scale_back = TRUE)
```




```
fitness_landscape(model7, observation = TRUE, scale_back = TRUE)
```




```
fitness_landscape(model8, covariate = 'nn5DistNotConsumed_s', observation = TRUE, scale_back = TRUE)
```



2-D Fitness landscape

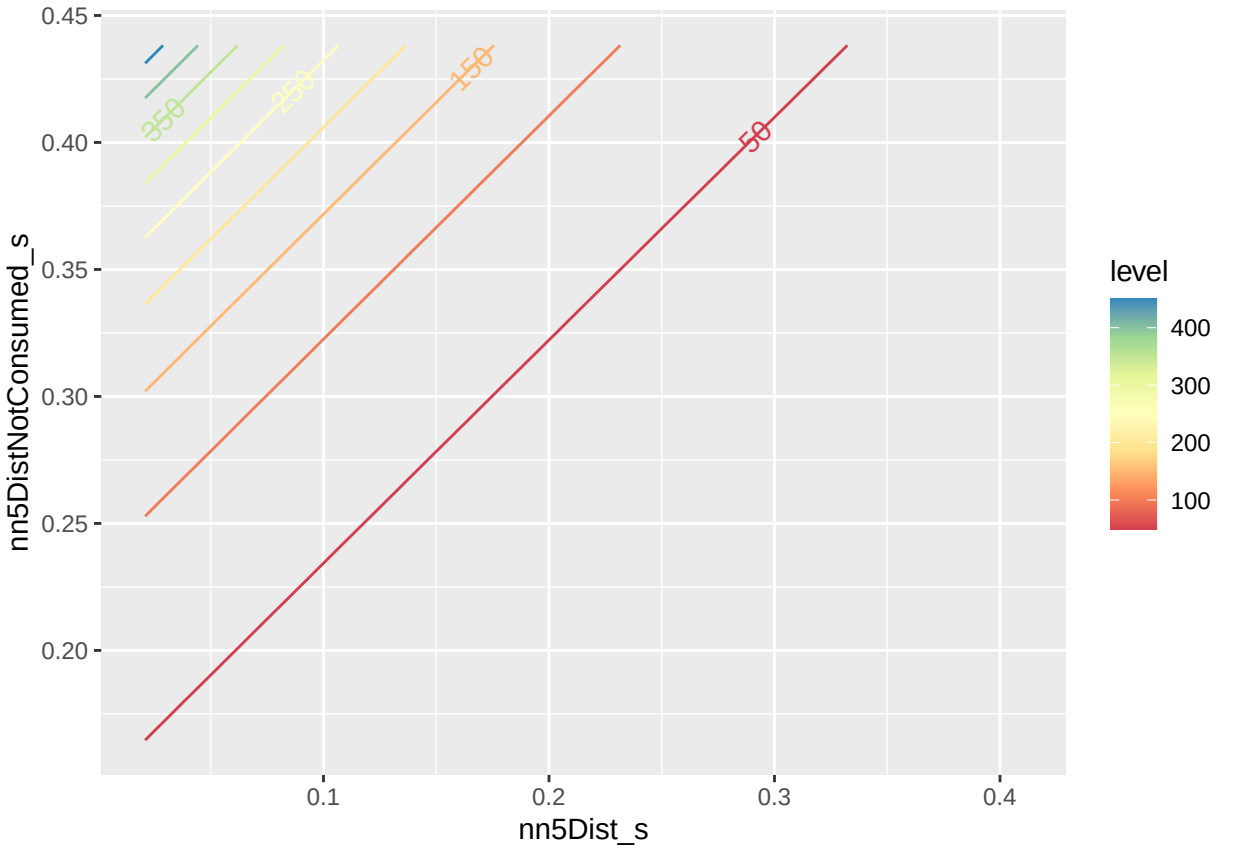
```
fitness_landscape_2d <- function(model) {
  # Make fake individuals
  nInd <- 50
  lwr_1 <-quantile(log(data$nn5Dist)/10, 0.025)
  upr_1 <-quantile(log(data$nn5Dist)/10, 0.975)
  lwr_2 <-quantile(test$nn5DistNotConsumed_s, 0.025)
  upr_2 <-quantile(test$nn5DistNotConsumed_s, 0.975)
  cand.nnA <- seq(from = lwr_1, to = upr_1, length = nInd)
  cand.nnB <- seq(from = lwr_2, to = upr_2, length = nInd)
  cand <- expand.grid(cand.nnA, cand.nnB)
  colnames(cand) <- c('nn5Dist_s', 'nn5DistNotConsumed_s')
  cand <- as.data.frame(cand)
  cand$root <- 1
  blah <- data[1:nInd, colnames(data) %in% vars]
  blah <- do.call("rbind", replicate(nInd, blah, simplify = FALSE))
  cand <- cbind(cand, blah)

  cand$id <- paste0('LP', seq(nInd^2))

  # Transform fake data into long format
  cand_long <- reshape(cand, varying = list(vars), direction="long", timevar="varb",
                  times = as.factor(vars), v.names="resp")
```

```
cand_long <- data.frame(cand_long)
cand_long$fit <- as.numeric(cand_long$varb == "embryoCt")
cand_long$Nid <- as.numeric(gsub("[^0-9.-]", "", cand_long$id))

cand_long$Deer <- as.numeric(cand_long$varb=='flCtNotConsumed')
cand_long$Pollination <- as.numeric(is.element(cand_long$varb,
                              c("capsuleCt", "isHarvested", "ovuleCt", "embryoCt")))

# Get conditional mean value parameters
pred <- predict(model, cand_long, varvar=varb, idvar=id, root=root,
                    se.fit = TRUE, model.type='conditional',
                is.always.parameter = TRUE, info.tol=1e-8)

xi_parm <- pred$fit
xi_parm_se <- pred$se.fit

names(xi_parm) <- paste0(cand_long$id,'.',cand_long$varb)
names(xi_parm_se) <- paste0(cand_long$id,'.',cand_long$varb)

xi_parm_grad <- pred$gradient
rownames(xi_parm_grad) <- paste0(cand_long$id,'.',cand_long$varb)
colnames(xi_parm_grad) <- names(model$coefficients)

# Expected fitness
Ids <- unique(cand_long$id)
exp_fitness <- rep(0,nInd^2)
names(exp_fitness) <- Ids
for (id in Ids) {
  exp_fitness[id] <- prod(xi_parm[grep(paste0(id,'\\.'), names(xi_parm))][-5])
}

# Get covariance matrix of xi
xi_parm_var = xi_parm_grad %*% solve(model$fisher) %*% t(xi_parm_grad)

# Delta Method
var_expfit <- rep(0,nInd^2)
names(var_expfit) <- Ids
for (id in Ids) {
  ind <- paste0(id,'\\.')
  xi <- xi_parm[grep(ind, names(xi_parm))][-5]
  var <- xi_parm_var[grep(ind, rownames(xi_parm_var))[-5],
                     grep(ind, colnames(xi_parm_var))[-5]]
  grad <- c(prod(xi[-1]), prod(xi[-2]), prod(xi[-3]), prod(xi[-4]), prod(xi[-5]), prod(xi[-6]))
  var_expfit[id] <- t(grad) %*% var %*% grad
}

se_expfit = sqrt(var_expfit)

# Plots
cand_block <- cand %>% mutate(exp_fitness, exp_fitness, lower = exp_fitness - se_expfit, upper = exp_


# p1 <- ggplot(data=cand_block, aes(nn5Dist_s, nn5DistNotConsumed_s, fill=lower)) +
```

```
  # geom_tile() +ggtitle(paste0("Fitness landscape    ", substr(format(model$formula), start=20, stop=n
  # scale_fill_gradient(low="white", high="blue") +
  # theme(plot.title = element_text(size=10))

  # p2 <- ggplot(data=cand_block, aes(nn5Dist_s, nn5DistNotConsumed_s, fill=upper)) +
  # geom_tile() +ggtitle(paste0("Fitness landscape    ", substr(format(model$formula), start=20, stop=n
  # scale_fill_gradient(low="white", high="blue") +
  #  theme(plot.title = element_text(size=10))

  # p3 <- ggplot(data=cand_block, aes(nn5Dist_s, nn5DistNotConsumed_s, fill=exp_fitness)) +
  # geom_tile() +ggtitle(paste0("Fitness landscape    ", substr(format(model$formula), start=20, stop=n
  # scale_fill_gradient(low="white", high="blue") +
  #  theme(plot.title = element_text(size=10))

  p3 <- ggplot(cand_block, aes(nn5Dist_s, nn5DistNotConsumed_s, z= exp_fitness, colour=stat(level))) +
          geom_contour() +
          geom_text_contour(aes(z = exp_fitness)) +
          scale_colour_distiller(palette = 'Spectral', direction = 1) +
          scale_x_continuous(limits = c(lwr_1, upr_1))
  print(p3)



  #return(list(lwr = p1, upr = p2, exp = p3))

  #ggplotly(p3)

}
```

Fitness landscape of model 9

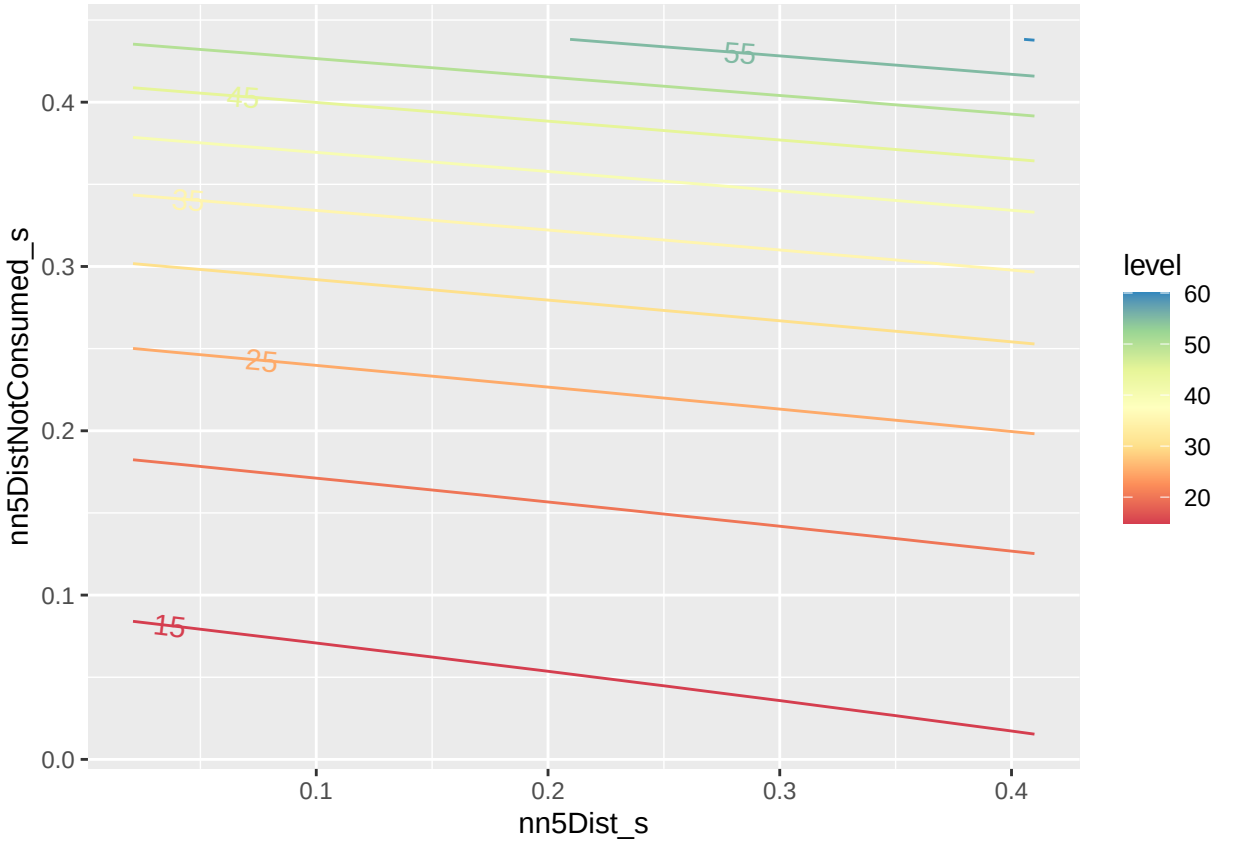
```
res <- fitness_landscape_2d(model9)
```




```
# res$lwr
# res$exp
# res$upr
```

Fitness landscape of model 10

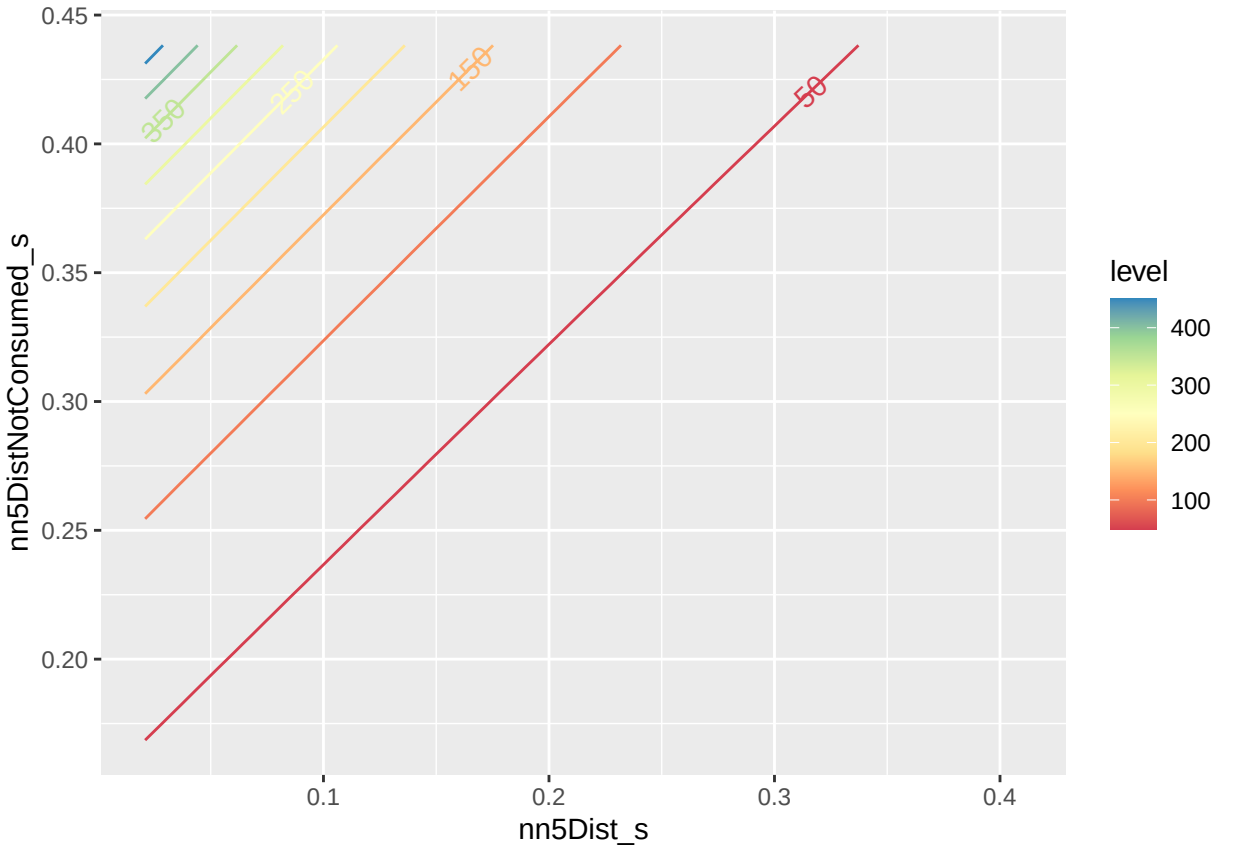
```
res <- fitness_landscape_2d(model10)
```




```
#res$lwr
#res$exp
#res$upr
#grid.arrange(res$lwr ,res$exp ,res$upr, ncol=2, nrow=2)
```

Fitness landscape of model 11

```
res <- fitness_landscape_2d(model11)
```

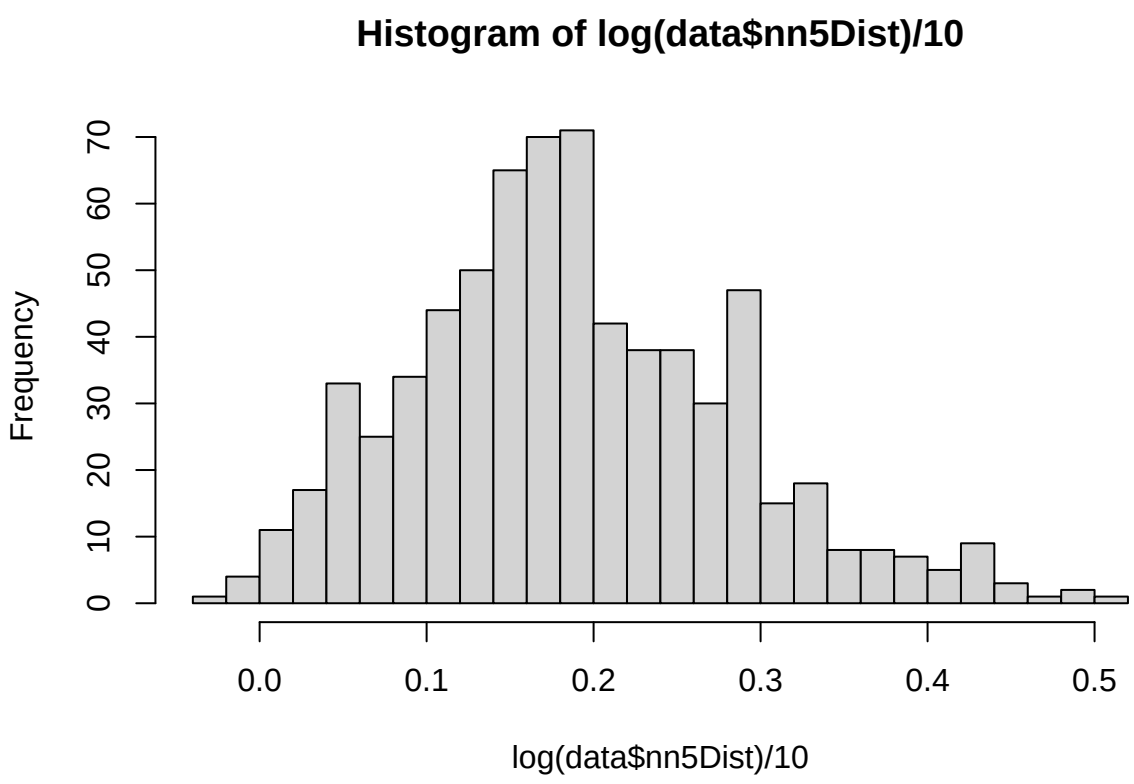



```
# res$lwr
# res$exp
# res$upr
```

Handling extreme values

We observed abnormal behaviors on fitness landscapes of `model7`, our best model chosen by AIC, at both right and left tails. So we try to handle these extreme values of `nn5Dist` study how they affect model performance.

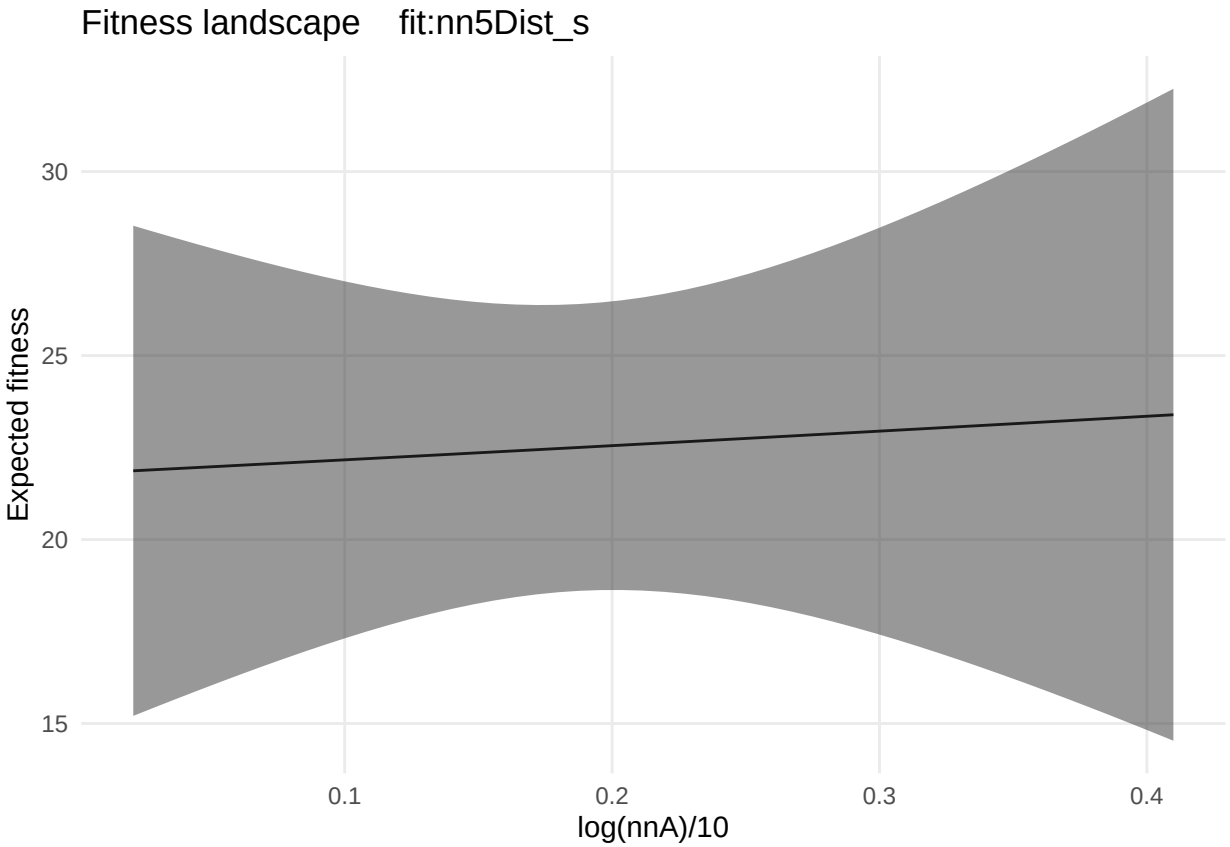
```
hist(log(data$nn5Dist)/10, breaks=20)
```



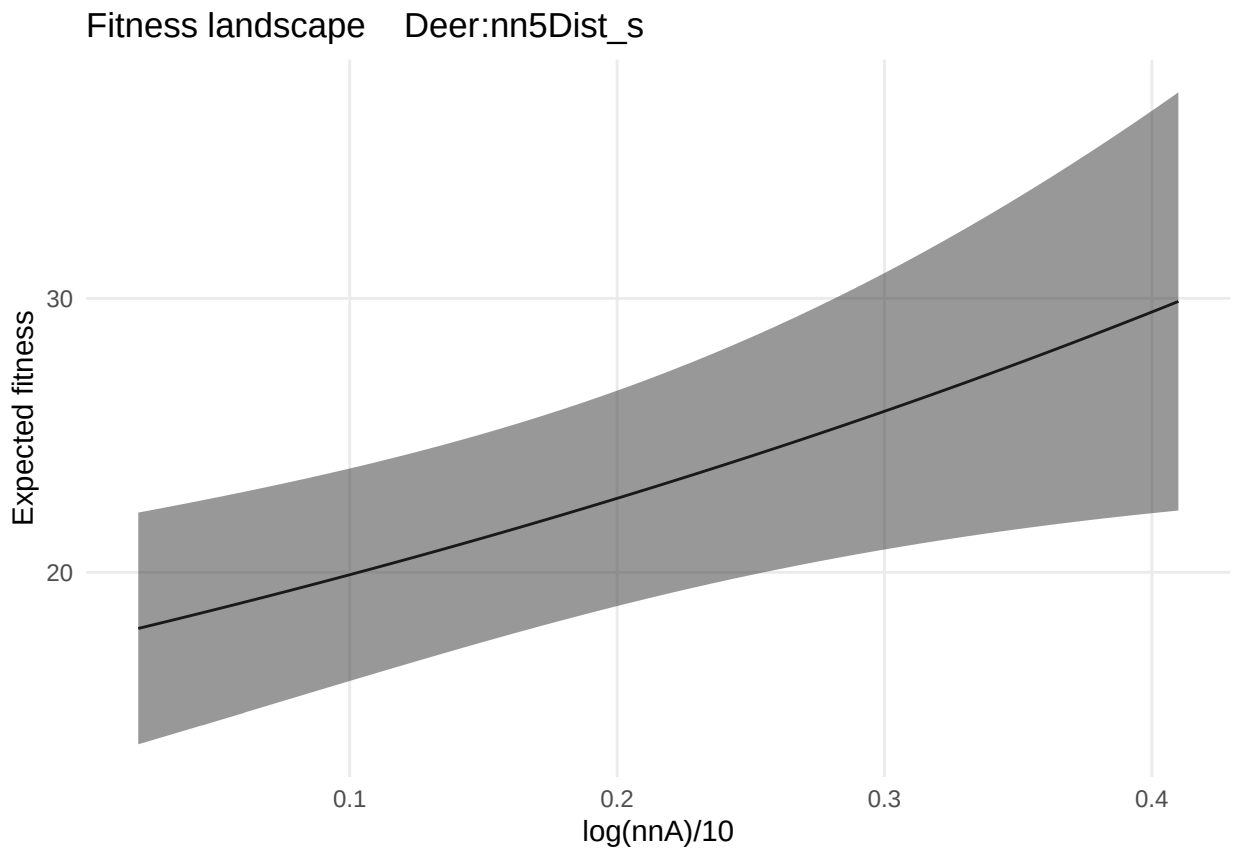

```
wide2long <- function(org_data) {
  pred <- c(0,1,2,3,4,5,6)
  fam <- c(2,1,1,1,1,2,1)
  vars <- c("flCt", "flCtNotConsumed","flCtUndamaged", "capsuleCt",
            "isHarvested", "ovuleCt", "embryoCt")

  test <- org_data %>% mutate(nn5Dist_s = log(nn5Dist)/10,
                          nn5DistNotConsumed =replace_na(nn5DistNotConsumed, 1)) %>%
    mutate(nn5DistNotConsumed_s = log(nn5DistNotConsumed)/10) %>%
    mutate(Ltail = ifelse(nn5Dist_s < quantile(nn5Dist_s, 0.025), 1, 0),
                        Rtail = ifelse(nn5Dist_s > quantile(nn5Dist_s, 0.975), 1, 0)) %>%
    mutate(middle = ifelse(Ltail+Rtail == 0, 1, 0))

  redata <- reshape(test, varying = list(vars), direction="long", timevar="varb",
                    times = as.factor(vars), v.names="resp")

  redata <- data.frame(redata, root = 1)
  redata$fit <- as.numeric(redata$varb == "embryoCt")
  redata$Nid <- as.numeric(gsub("[^0-9.-]", "", redata$id))
  redata$Deer <- as.numeric(redata$varb=='flCtNotConsumed')
  redata$Pollination <- as.numeric(is.element(redata$varb,
                                c("capsuleCt", "isHarvested", "ovuleCt", "embryoCt")))


  return(redata)
}
```

Zooming-in to the middle range of original landscapes

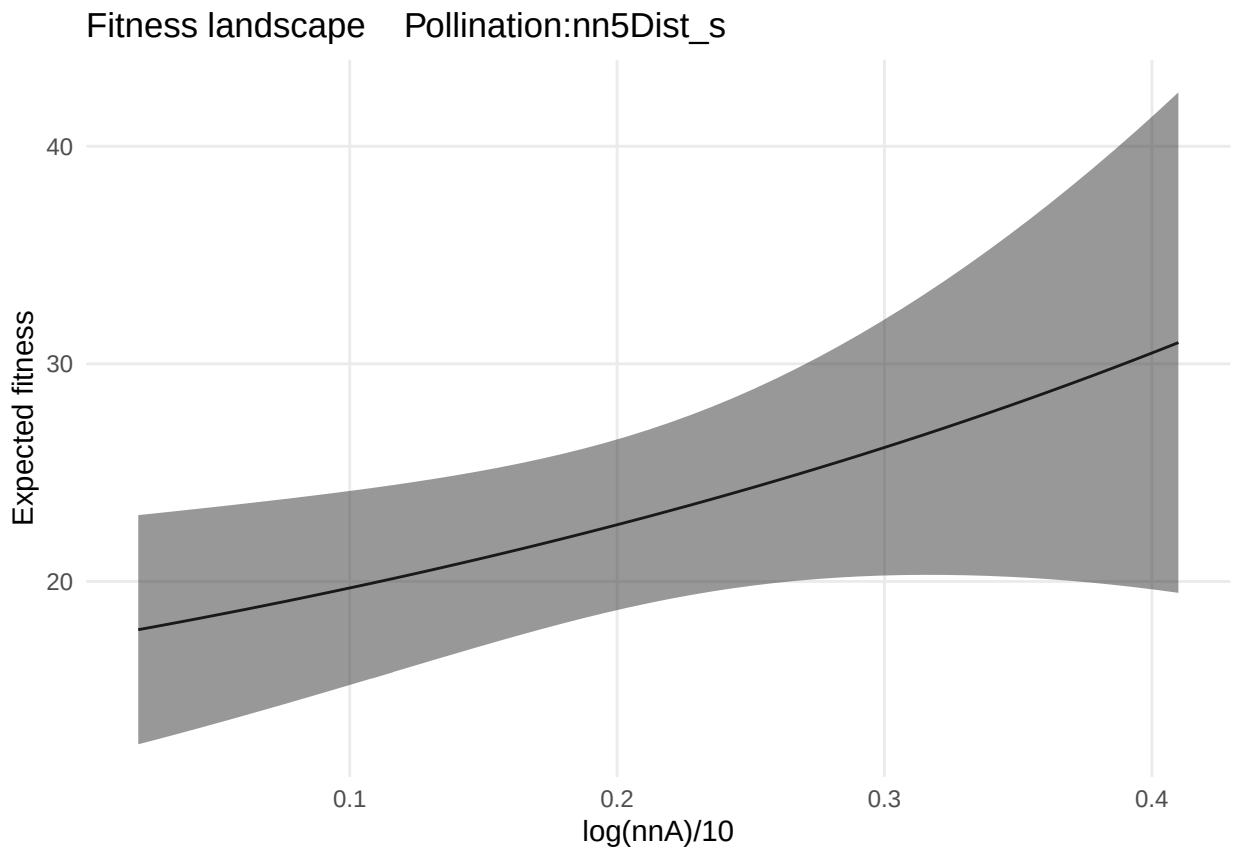
```
#par(mar = c(4, 4, .1, .1))
fitness_landscape(model1, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



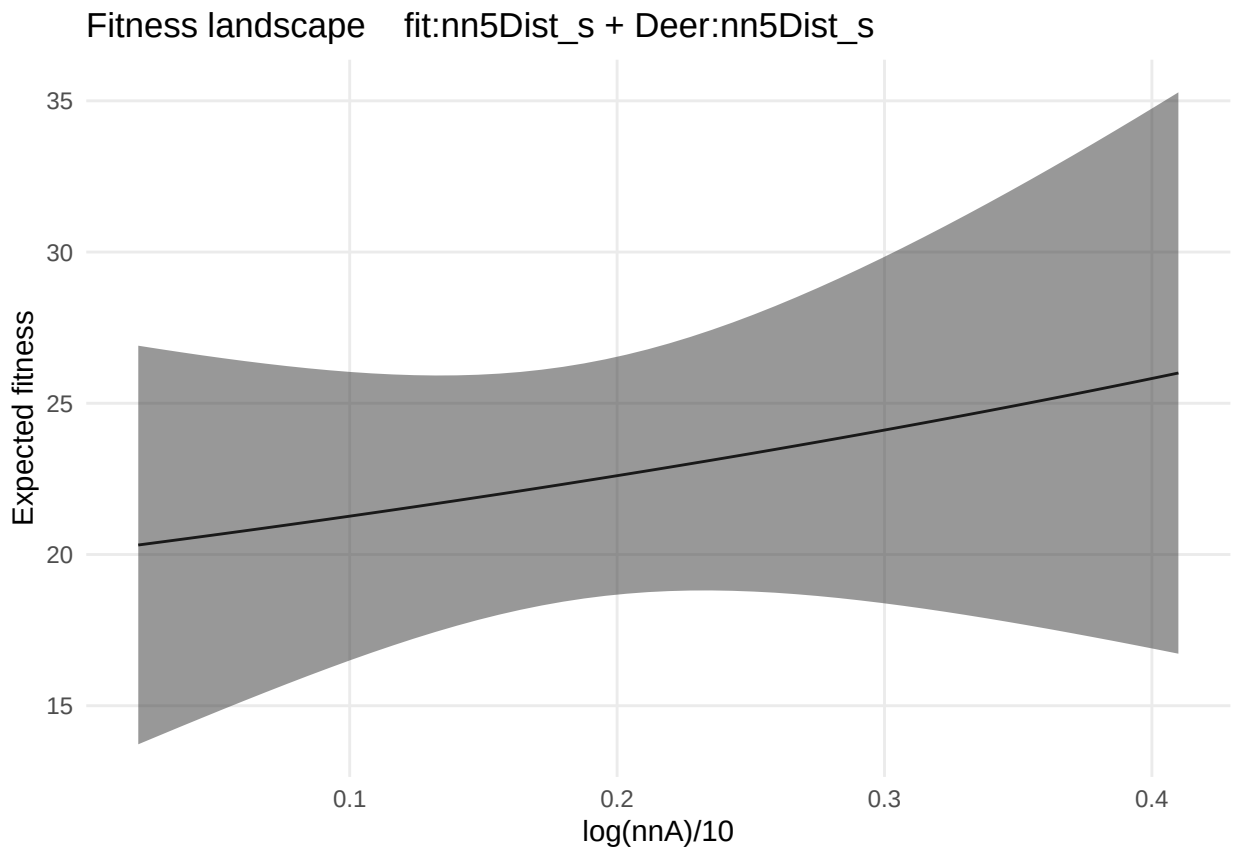

```
fitness_landscape(model2, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



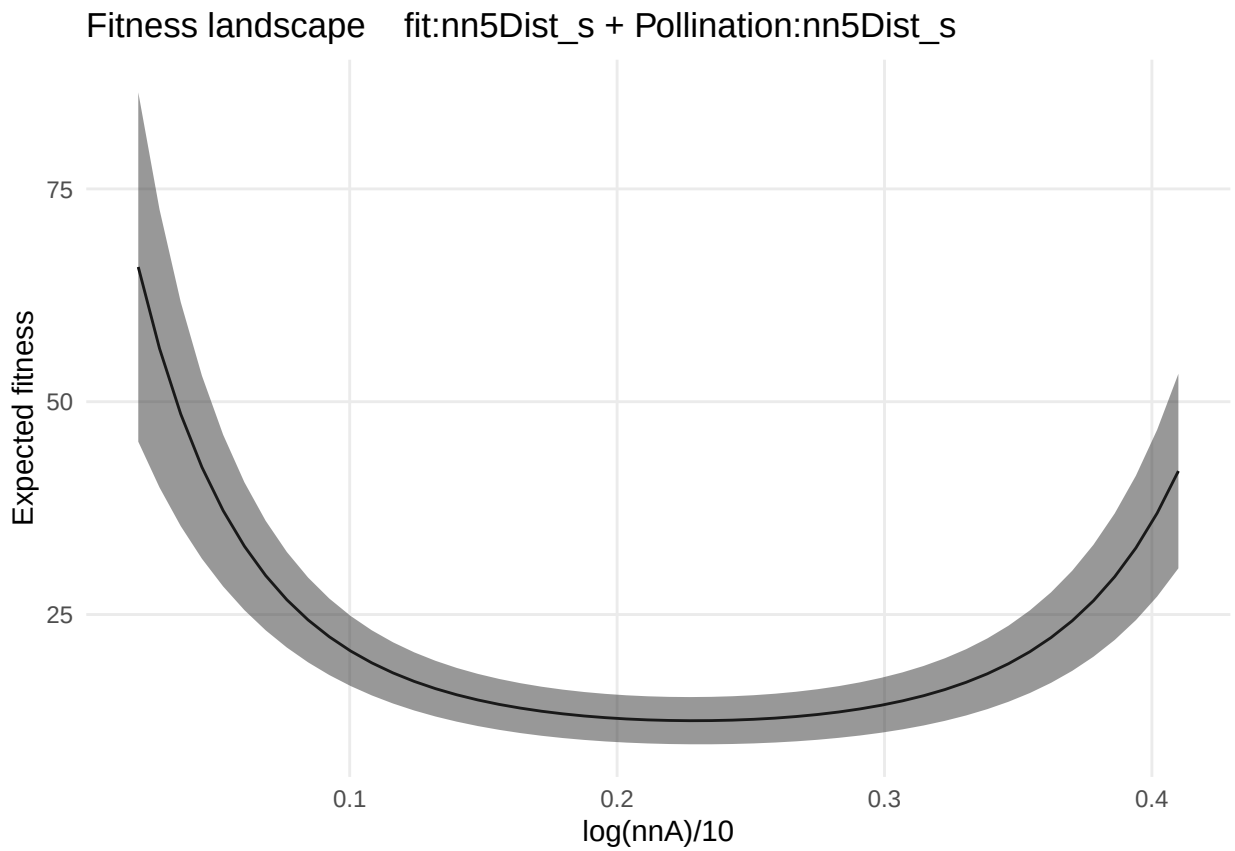

```
fitness_landscape(model3, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



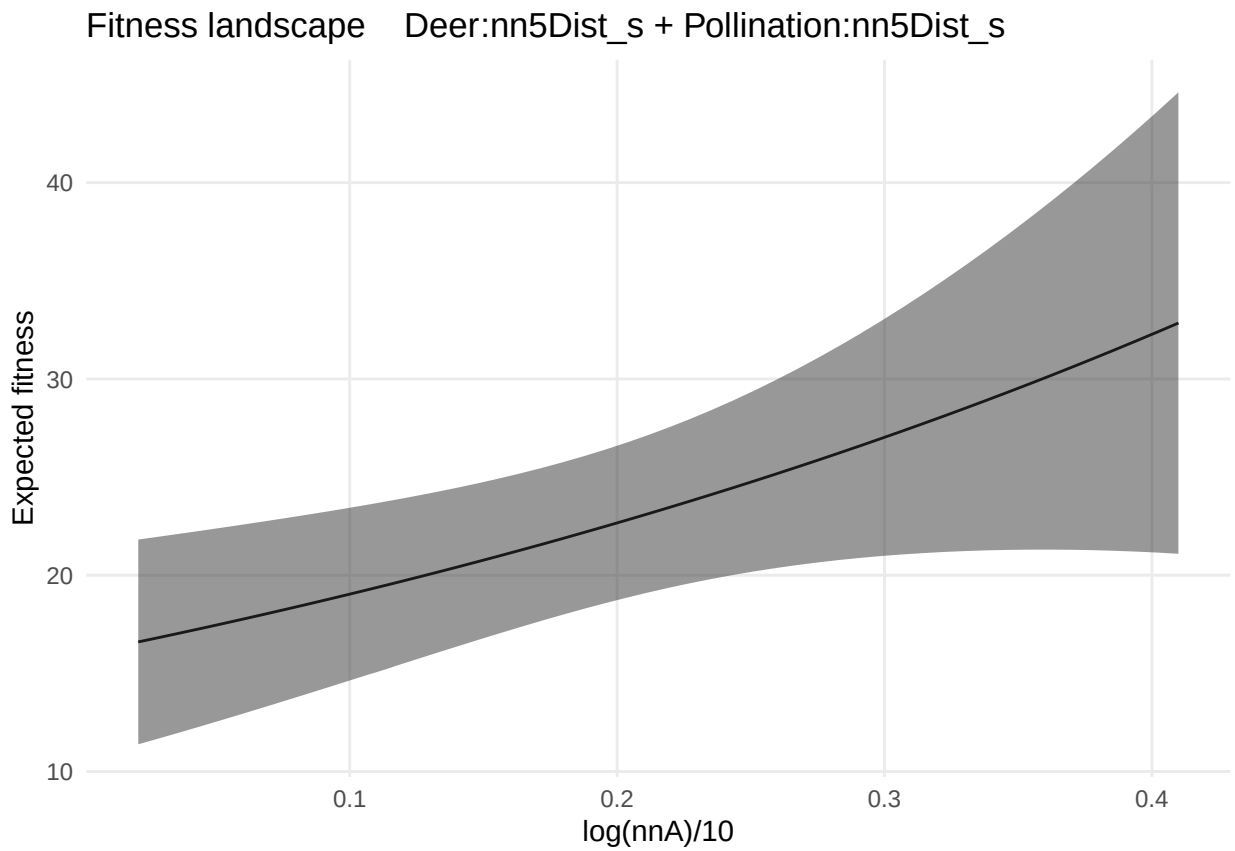

```
fitness_landscape(model4, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



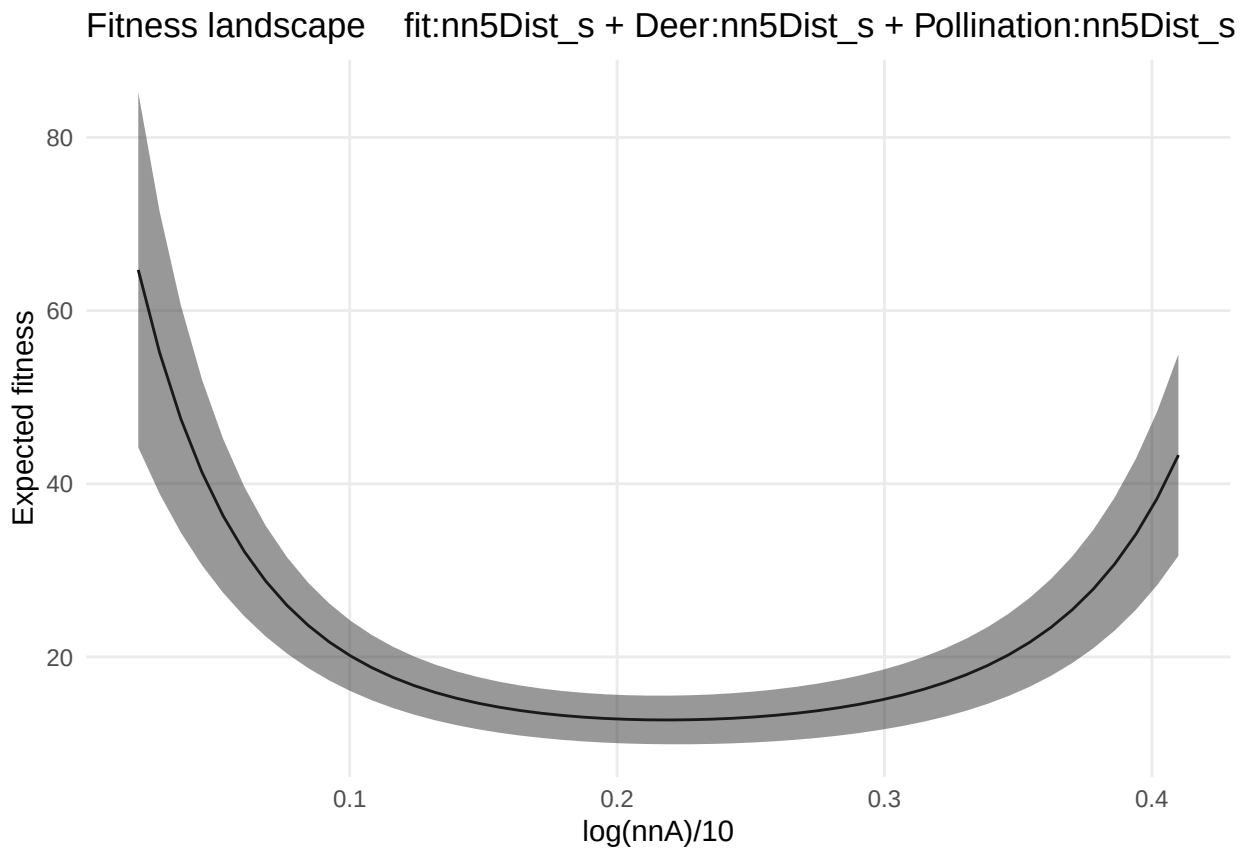

```
fitness_landscape(model5, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



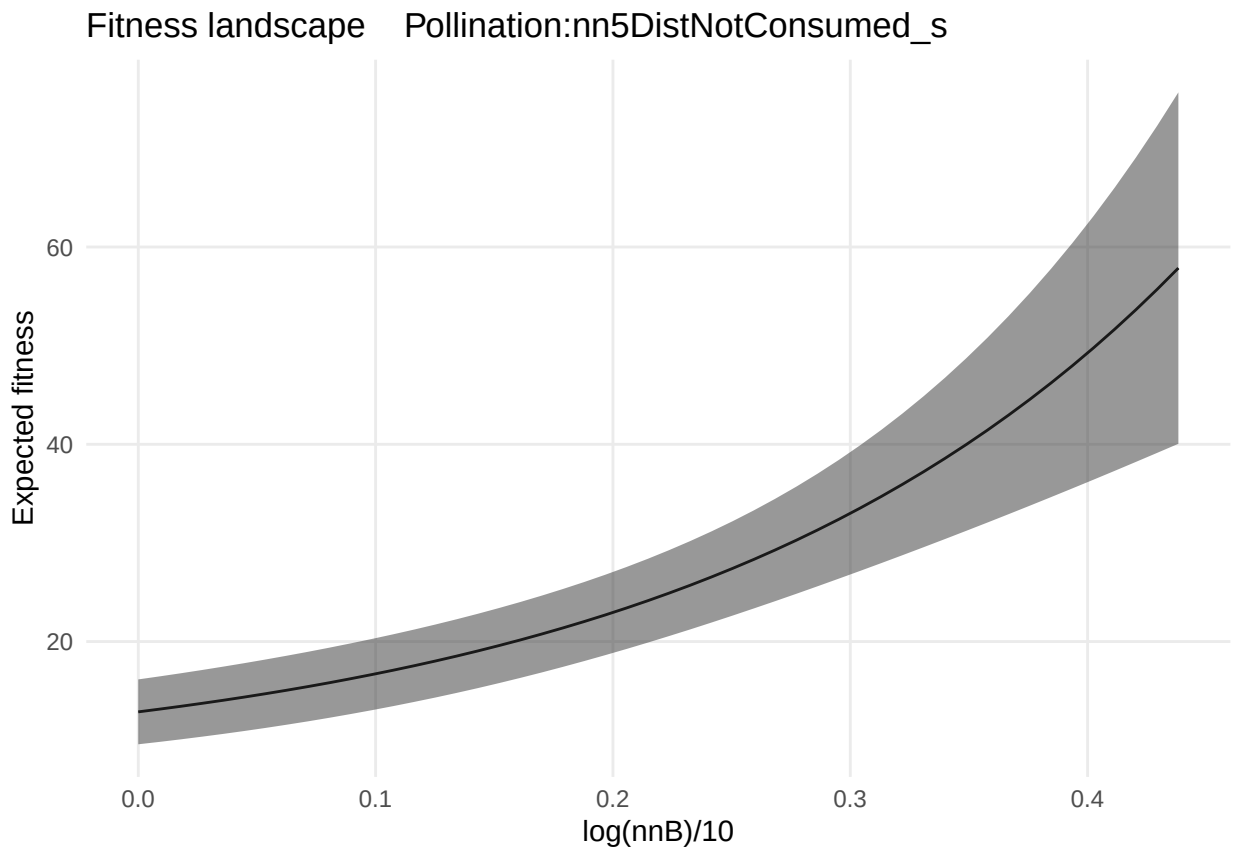

```
fitness_landscape(model6, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



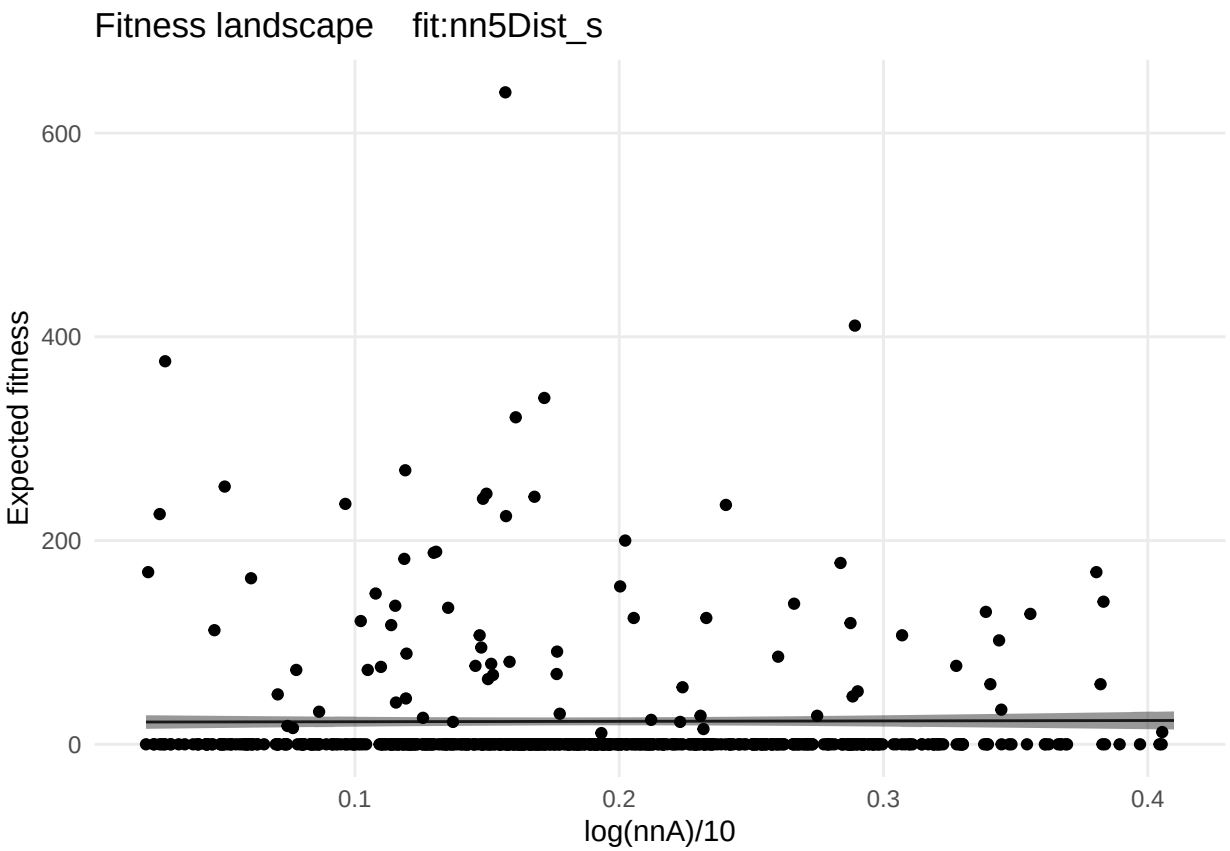

```
fitness_landscape(model7, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



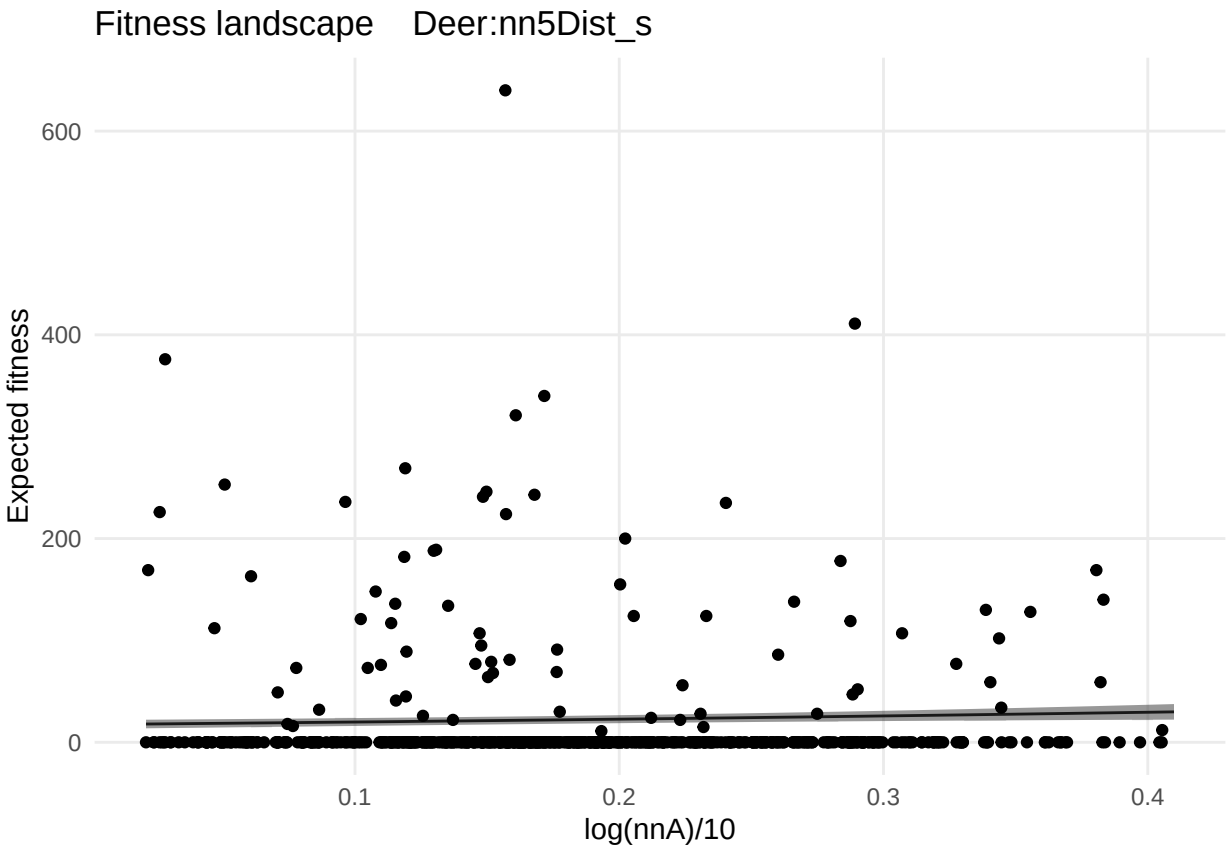

```
fitness_landscape(model8, covariate = 'nn5DistNotConsumed_s', lower = quantile(test$nn5DistNotConsumed_s
```



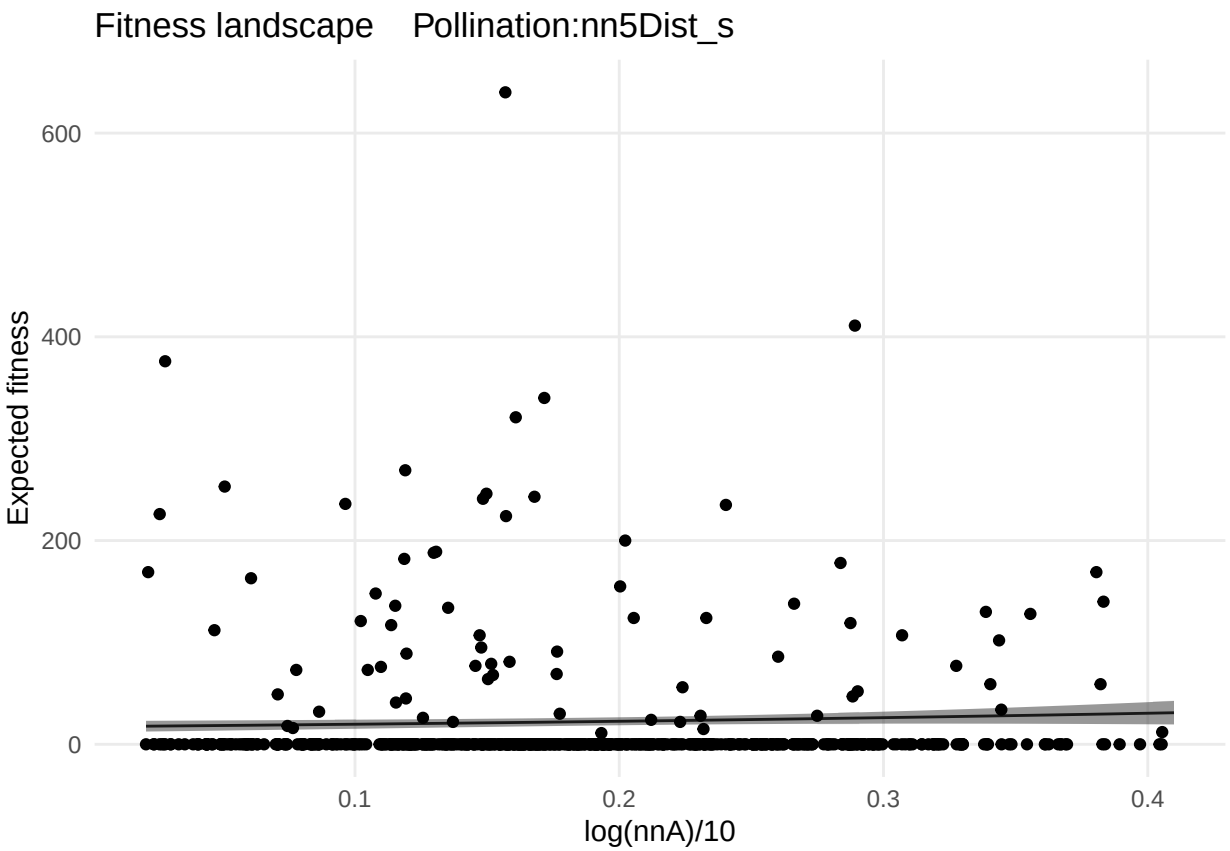

```
#par(mar = c(4, 4, .1, .1))
fitness_landscape(model1, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



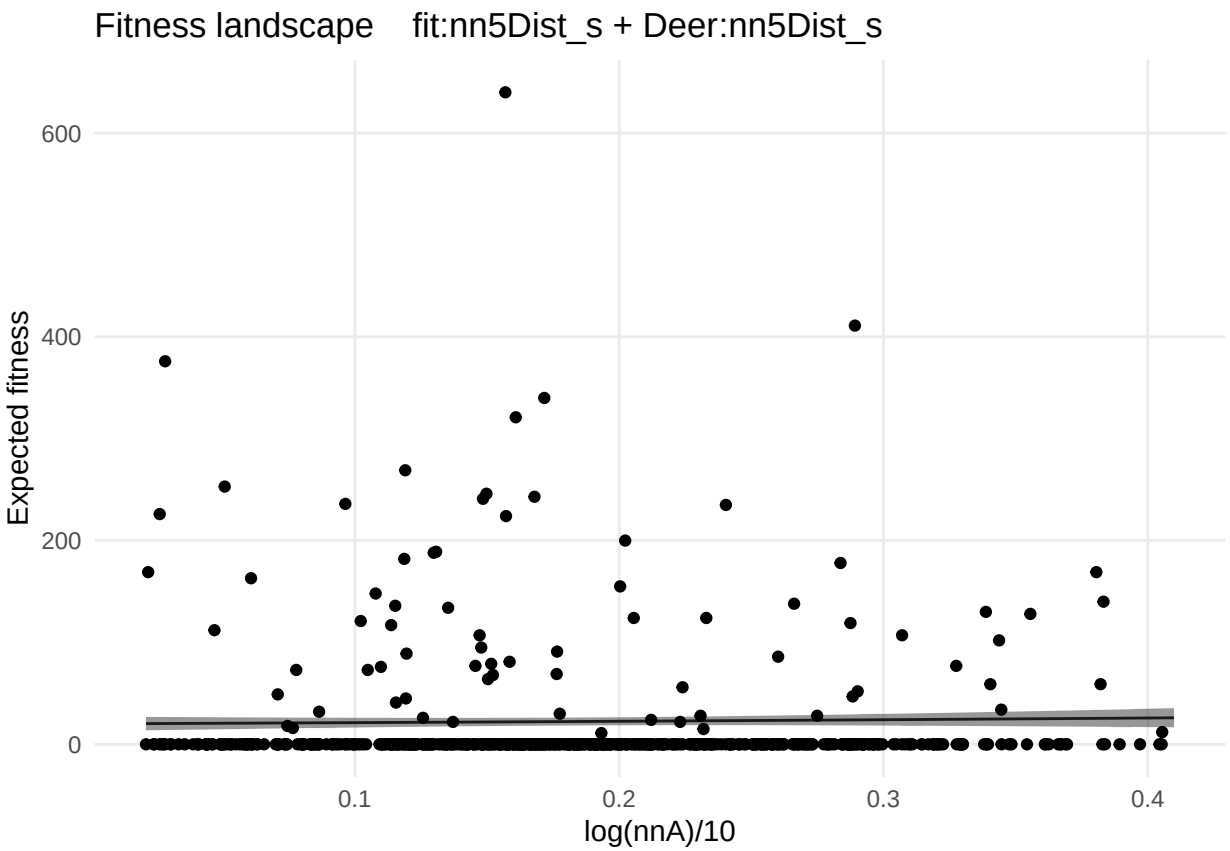

```
fitness_landscape(model2, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



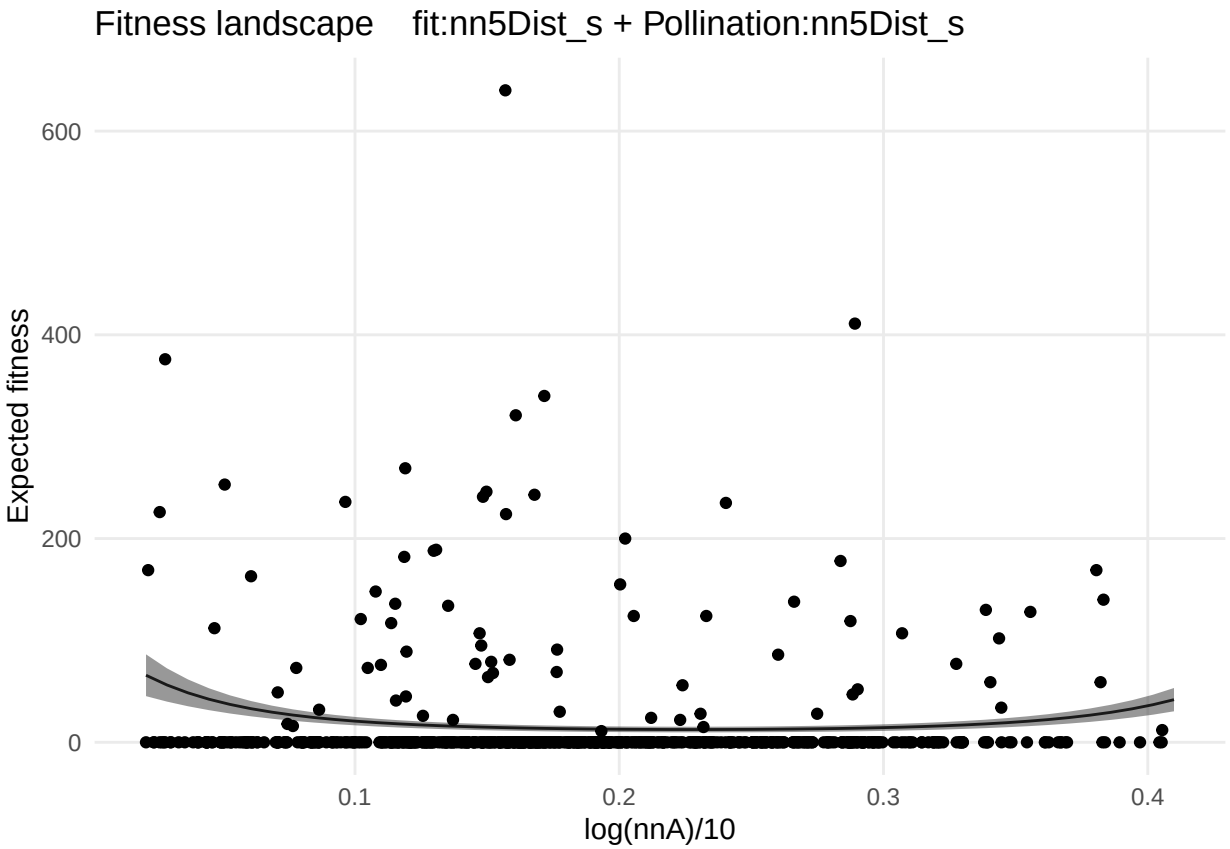

```
fitness_landscape(model3, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



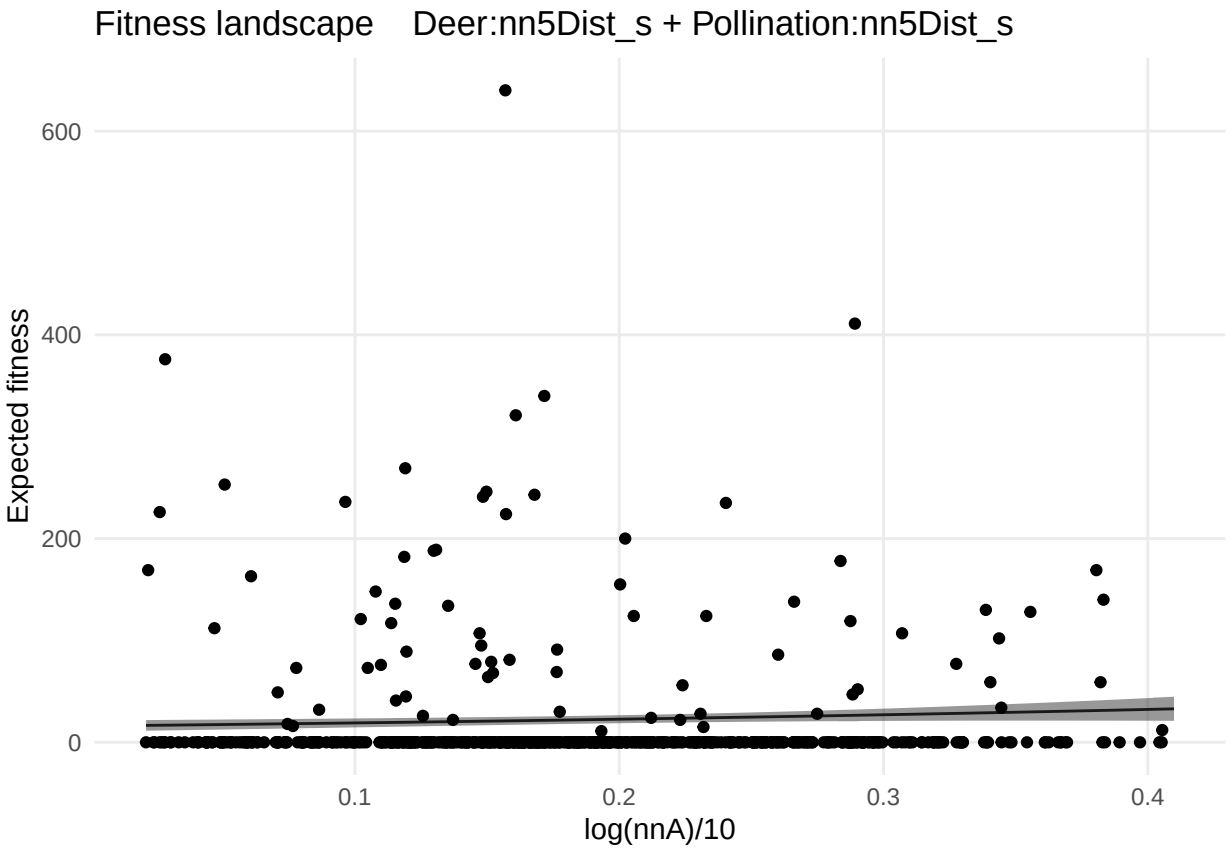

```
fitness_landscape(model4, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



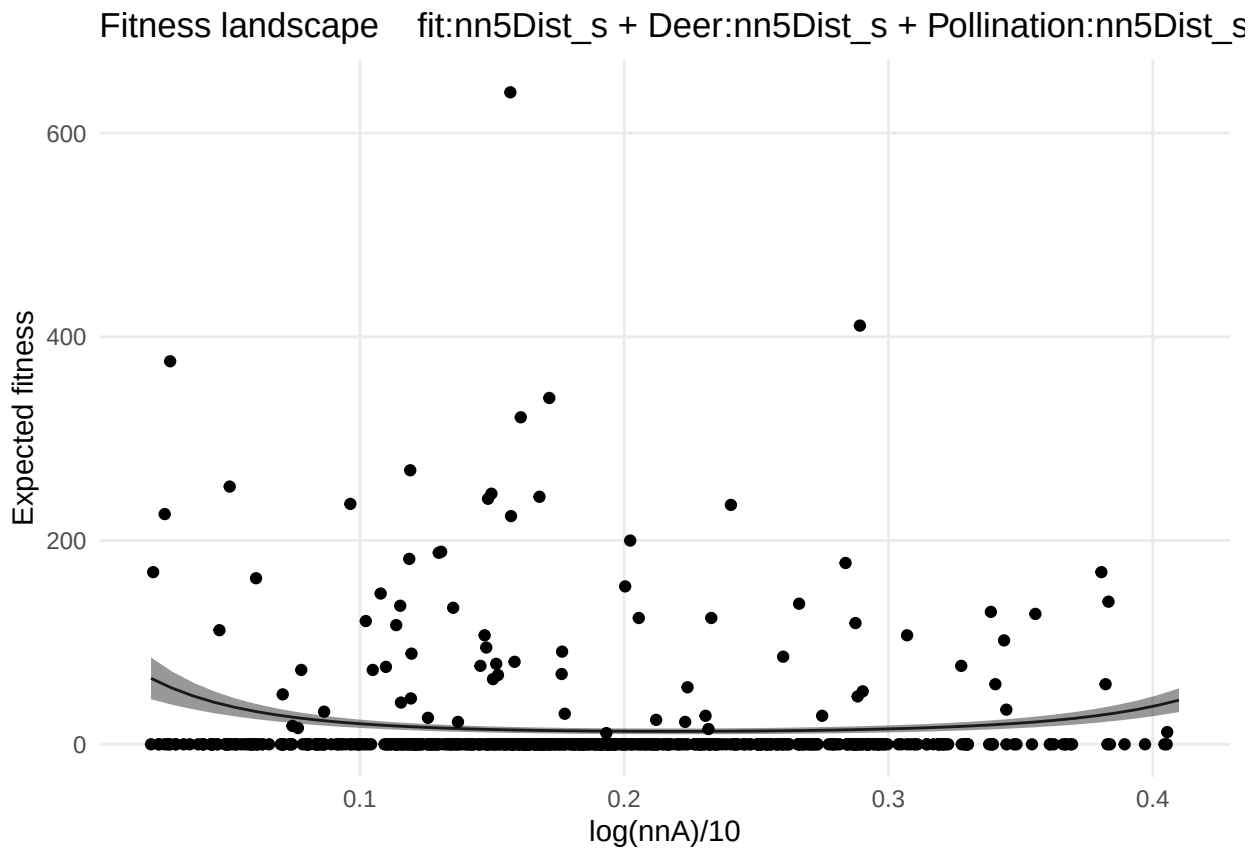

```
fitness_landscape(model5, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



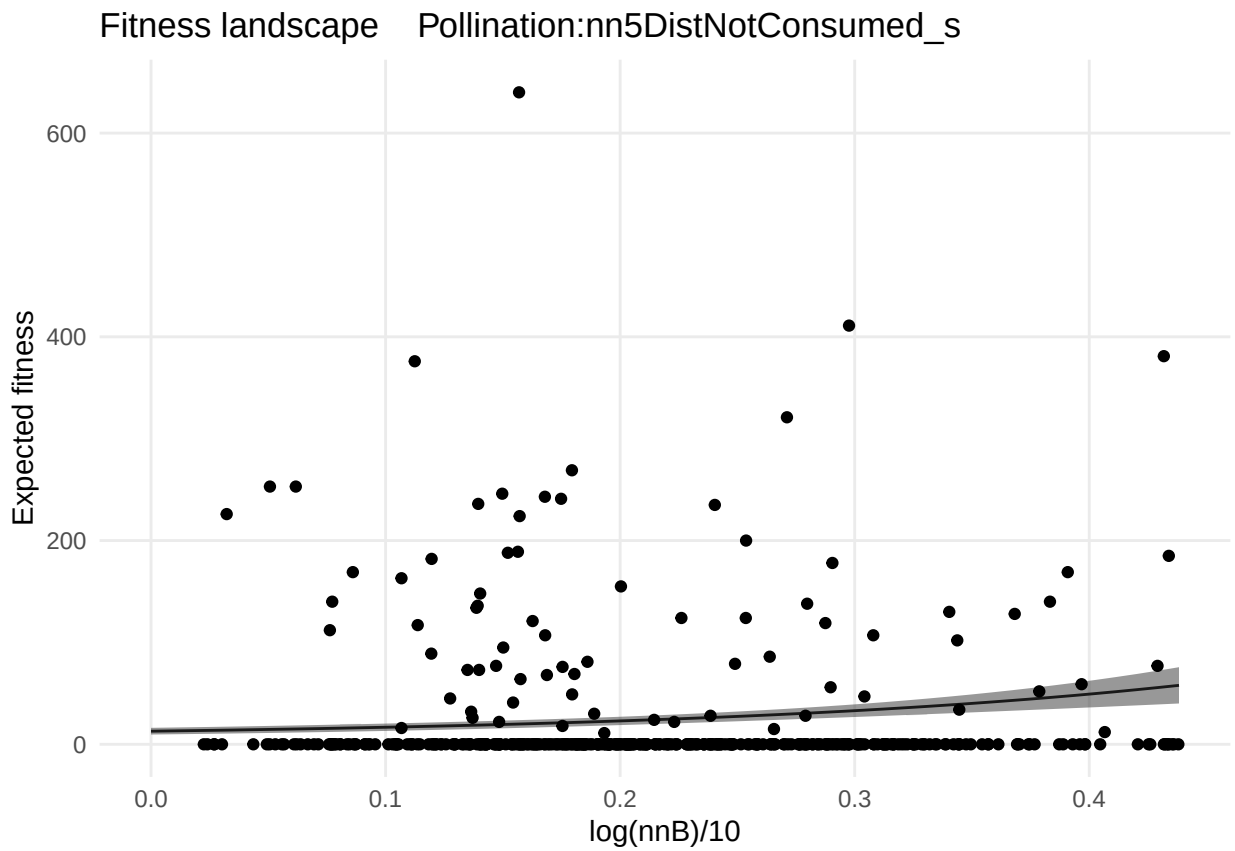

```
fitness_landscape(model6, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



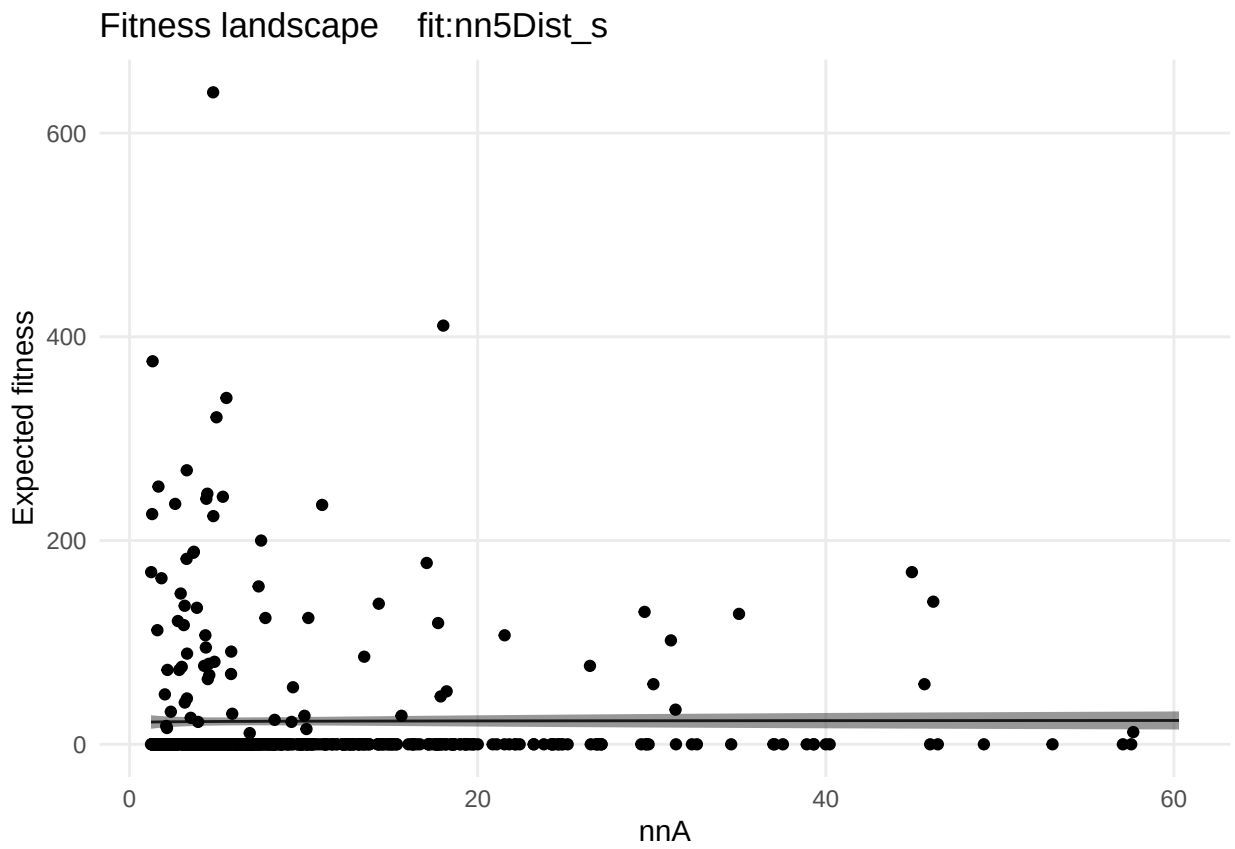

```
fitness_landscape(model7, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



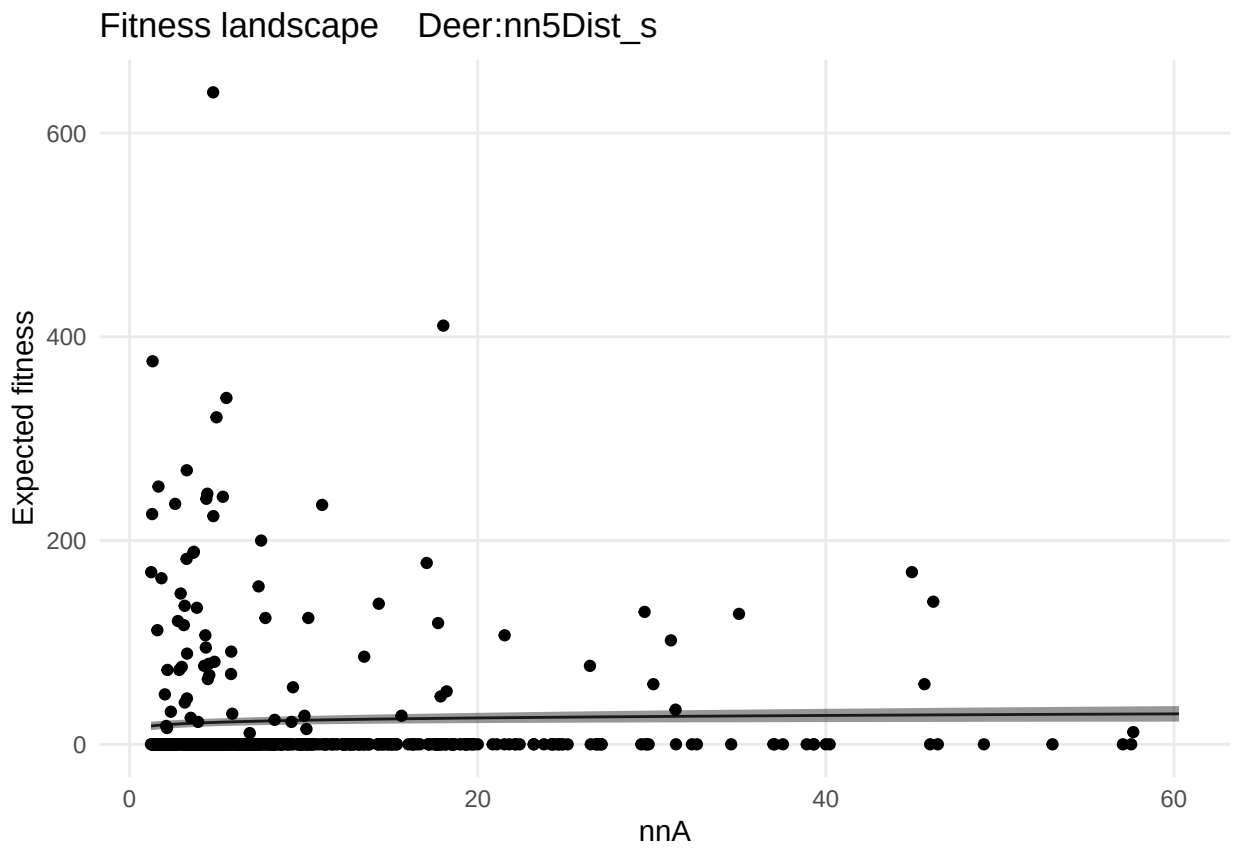

```
fitness_landscape(model8, covariate = 'nn5DistNotConsumed_s', lower = quantile(test$nn5DistNotConsumed_s
```



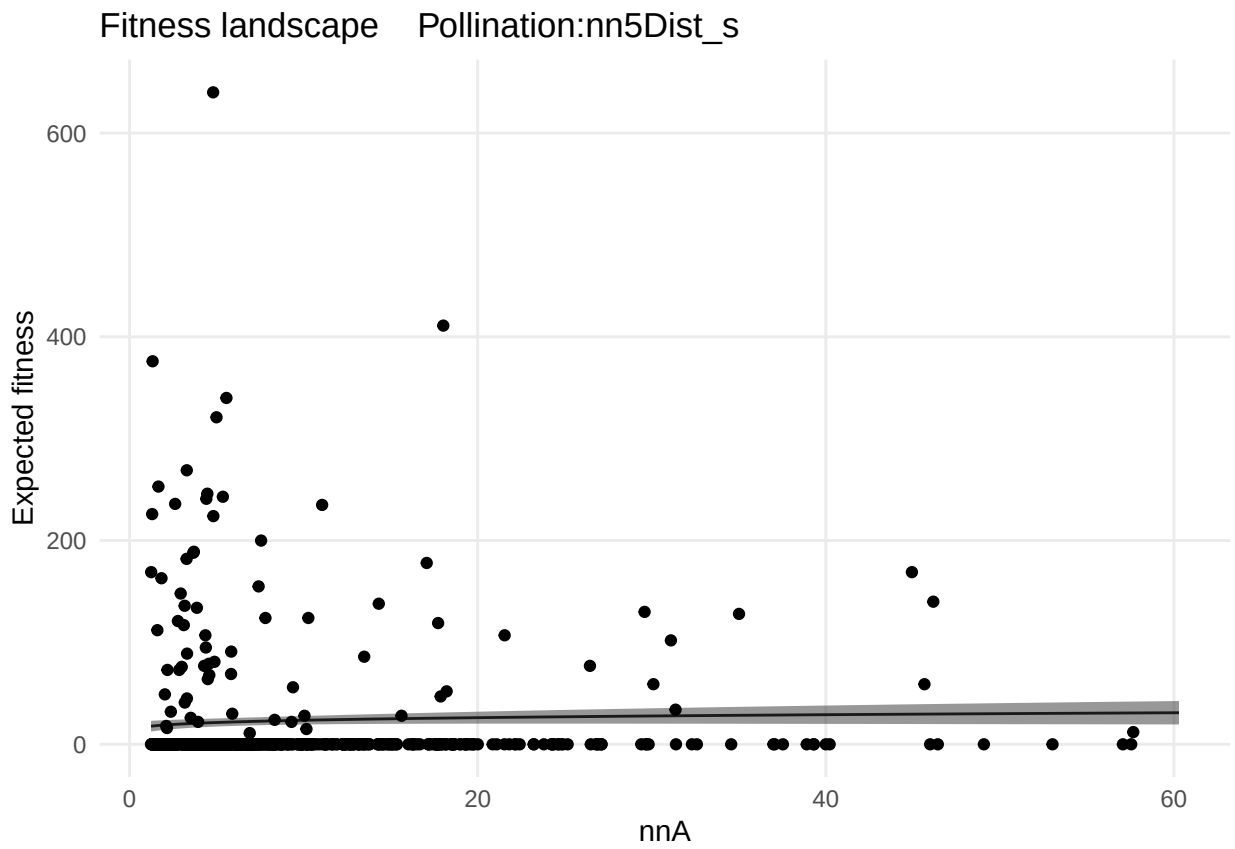

```
#par(mar = c(4, 4, .1, .1))
fitness_landscape(model1, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



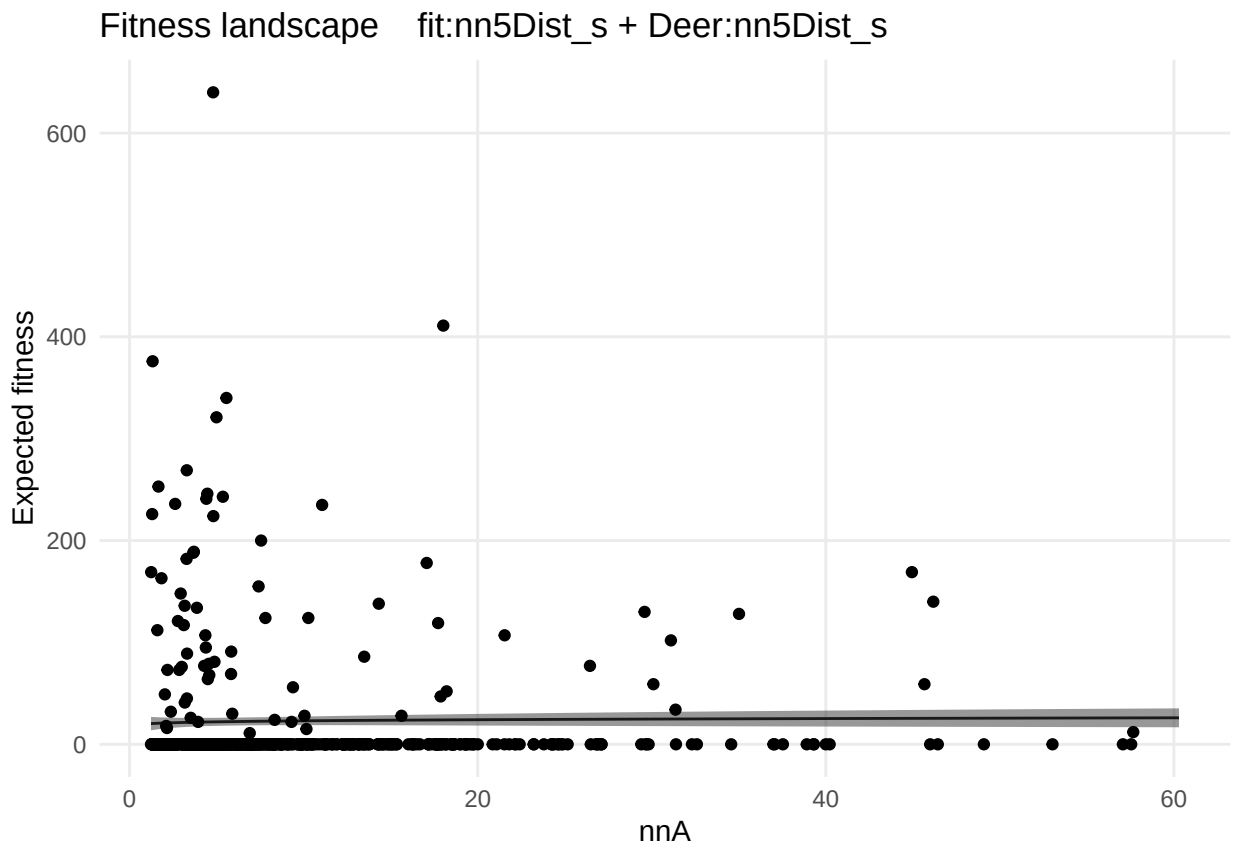

```
fitness_landscape(model2, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



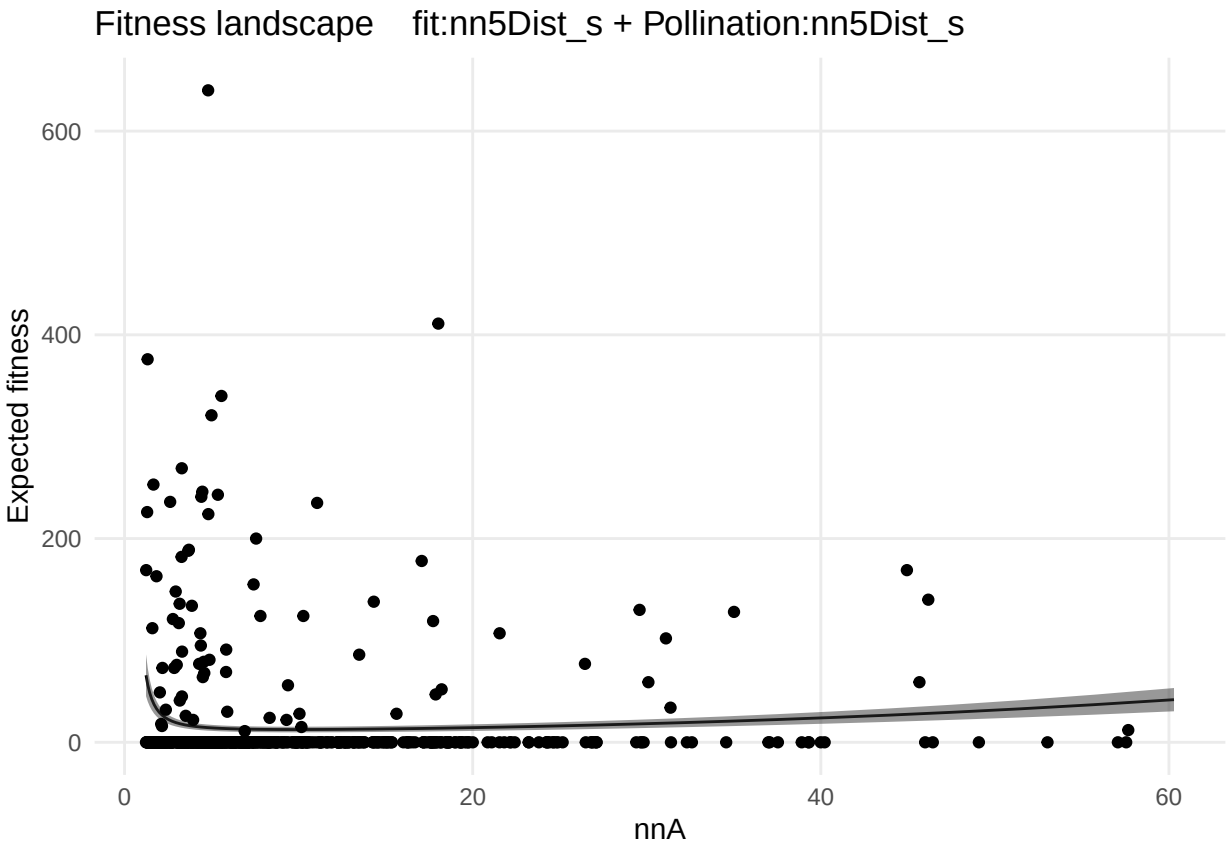

```
fitness_landscape(model3, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



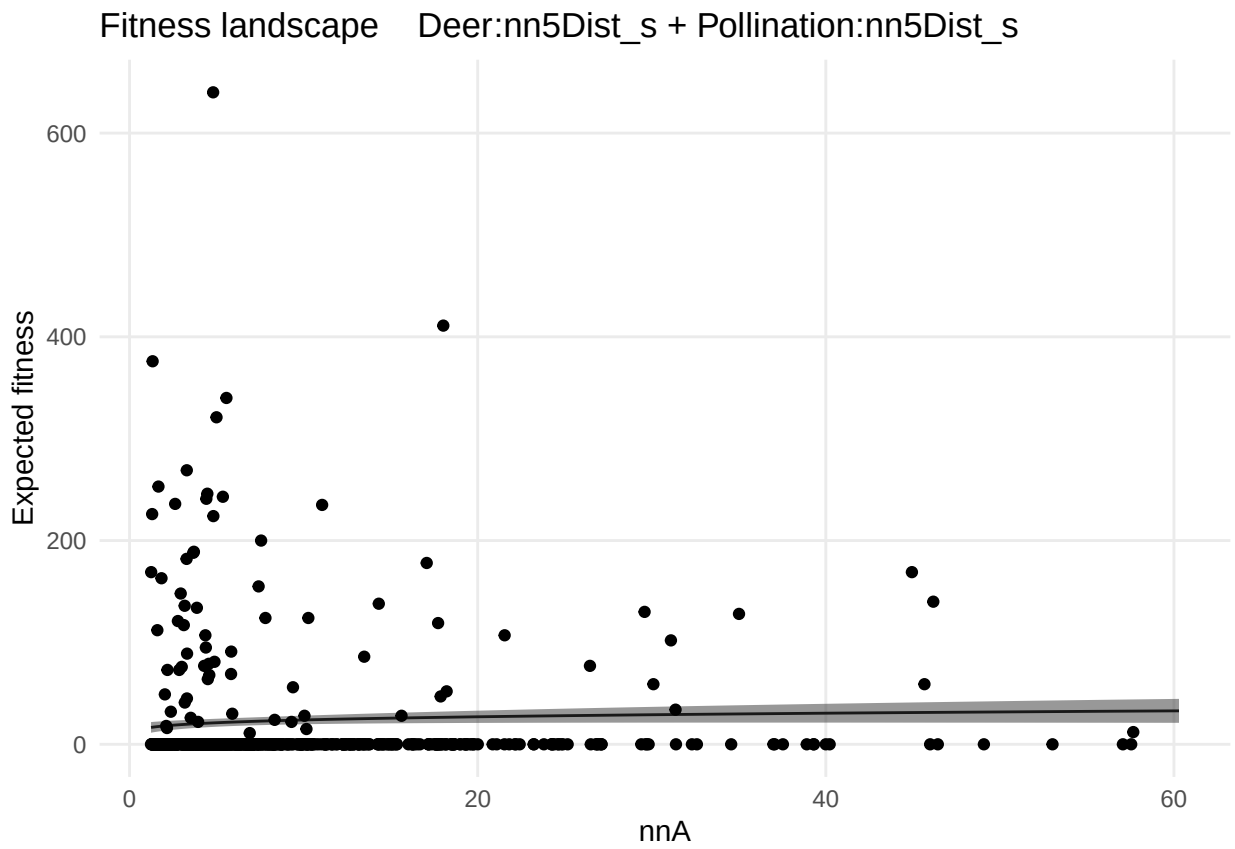

```
fitness_landscape(model4, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



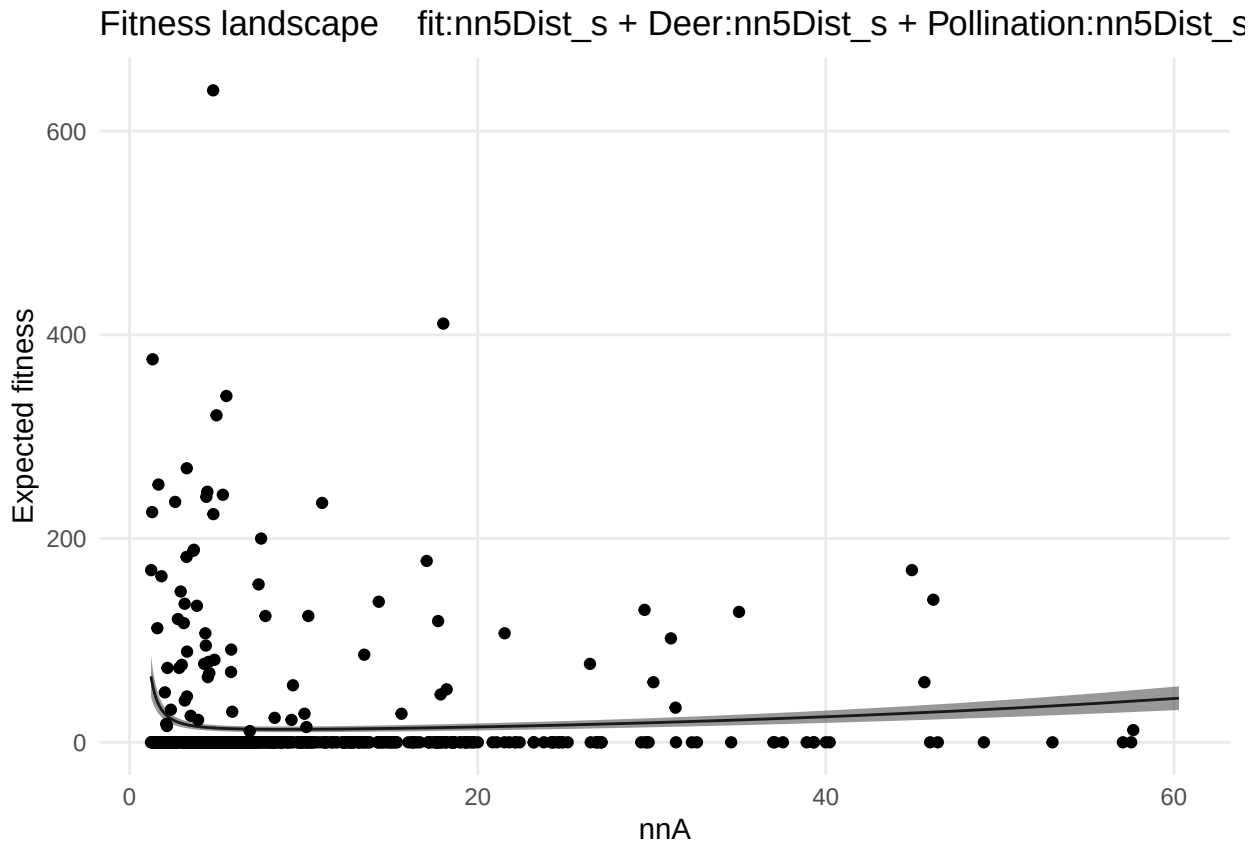

```
fitness_landscape(model5, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```



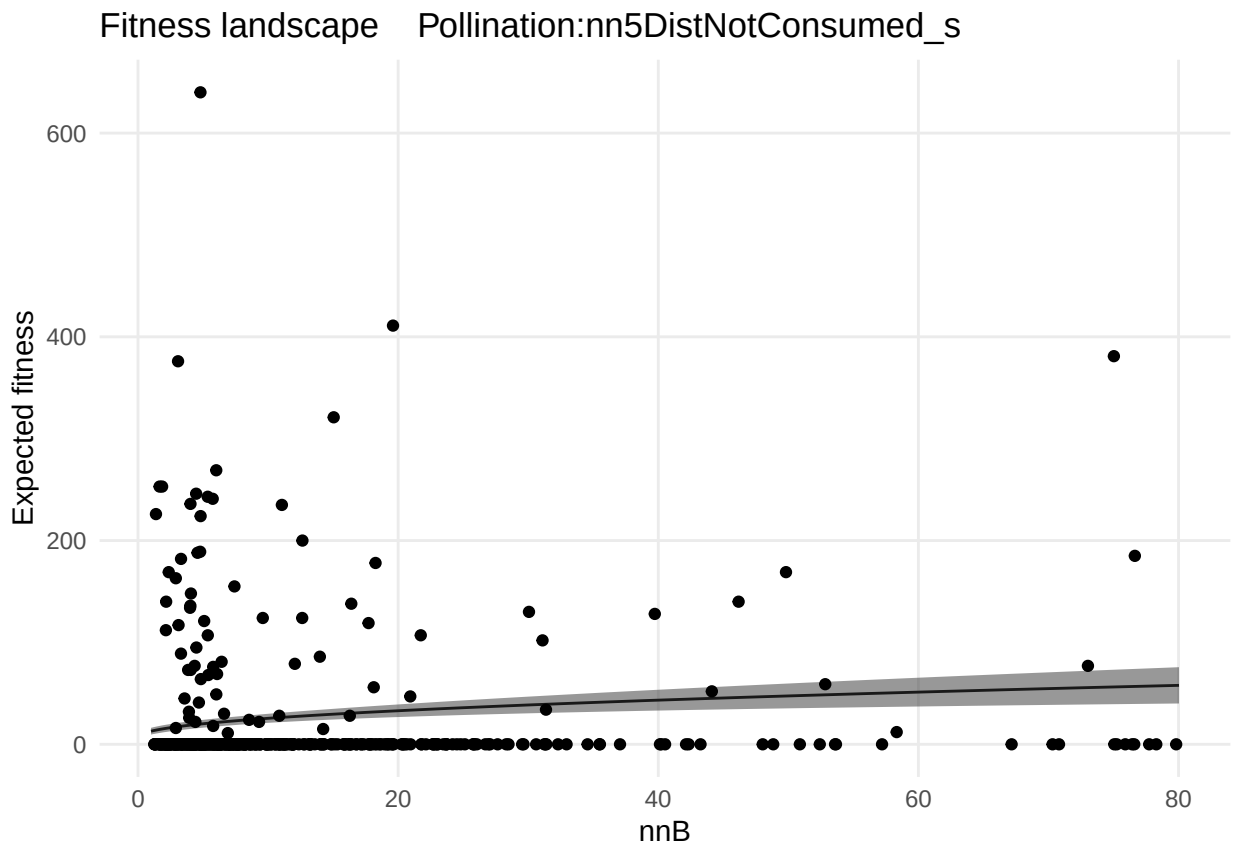

```
fitness_landscape(model6, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```




```
fitness_landscape(model7, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$nn5D
```




```
fitness_landscape(model8, covariate = 'nn5DistNotConsumed_s', lower = quantile(test$nn5DistNotConsumed_s
```

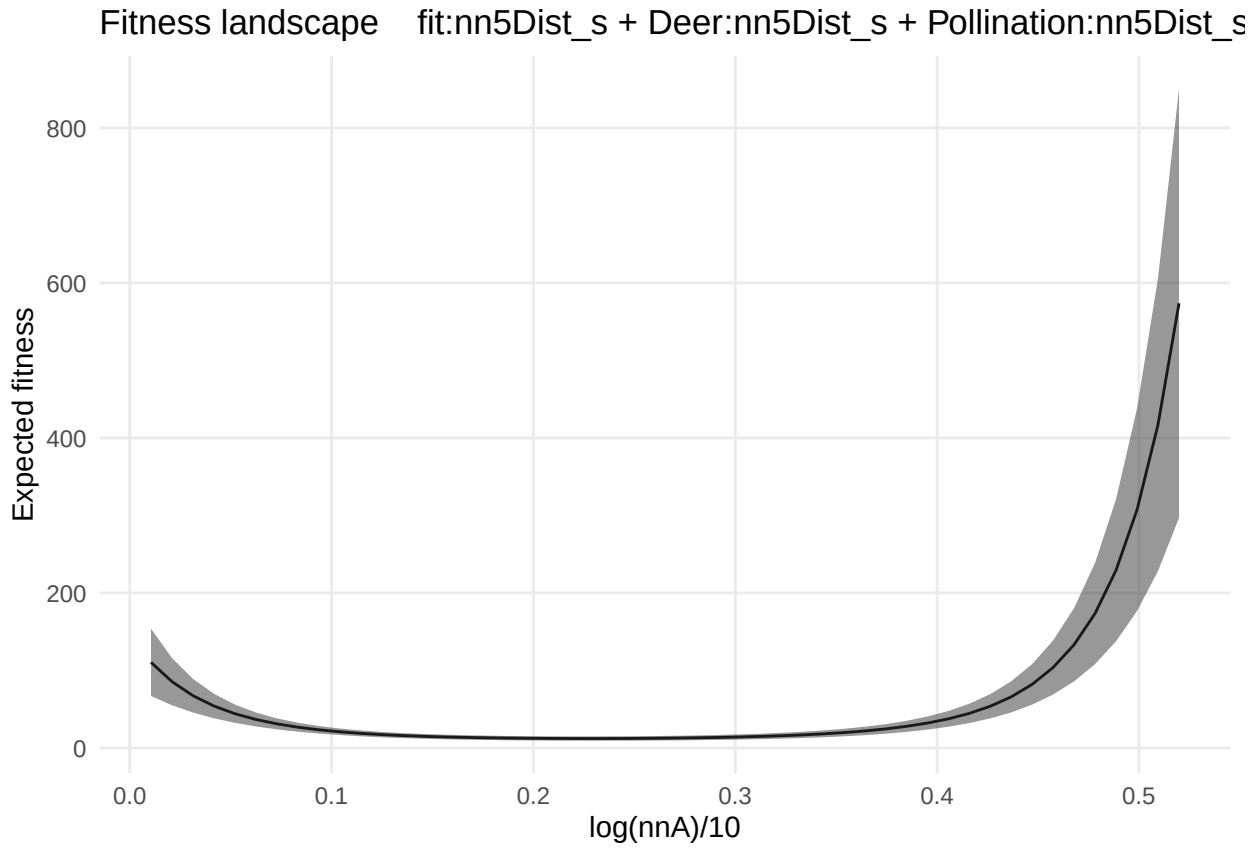


Refit models on trimmed data

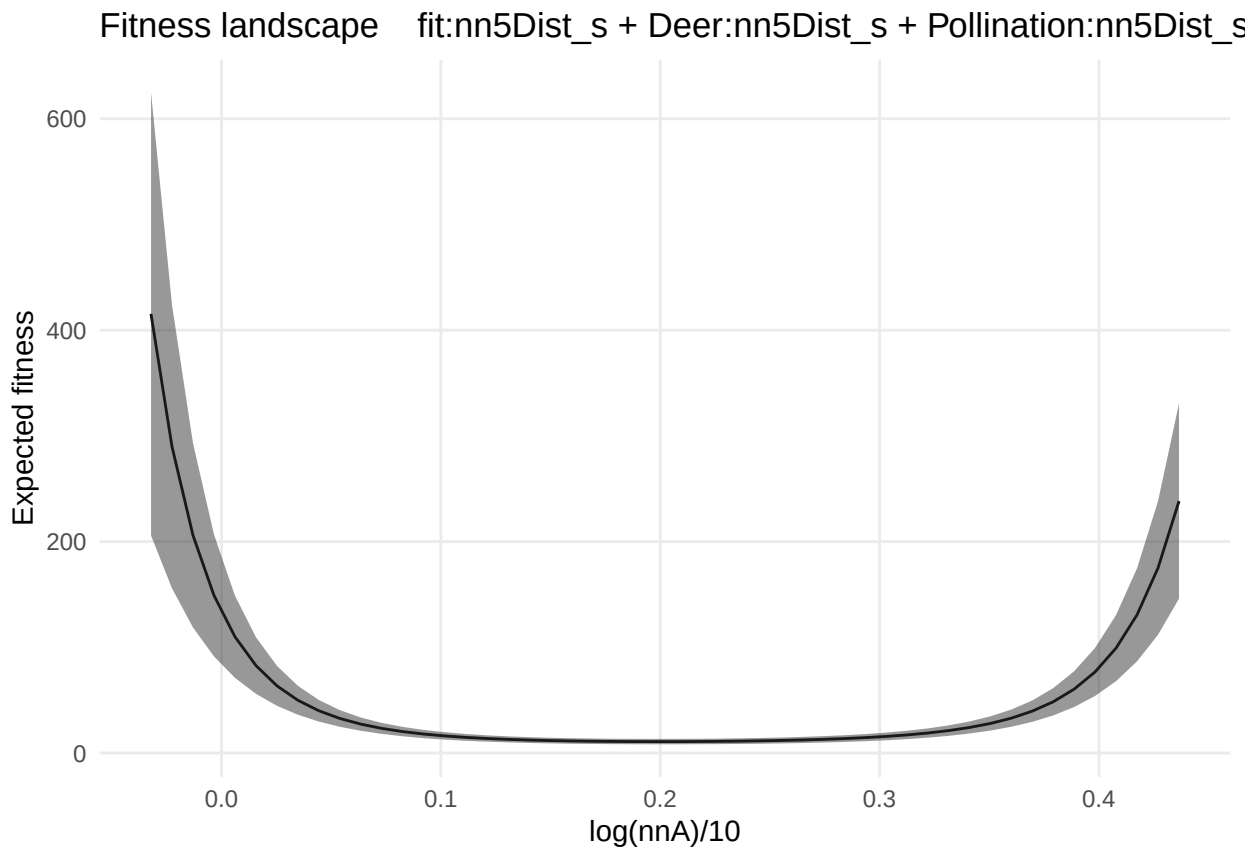
```
model_selection <- function(redata) {
  mod.null <- aster(resp ~ -1 + varb, pred, fam,varb,id,root,data=redata)
  mod1 <- aster(resp ~ -1 + varb + fit:nn5Dist_s, pred, fam,varb,id,root,data=redata)
  mod2 <- aster(resp ~ -1 + varb + Deer:nn5Dist_s, pred, fam,varb,id,root,data=redata)
  mod3 <- aster(resp ~ -1 + varb + Pollination:nn5Dist_s, pred, fam,varb,id,root,data=redata)
  mod4 <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s, pred, fam,varb,id,root,data=redata)
  mod5 <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Pollination:nn5Dist_s, pred, fam,varb,id,root,data=r
  mod6 <- aster(resp ~ -1 + varb + Deer:nn5Dist_s + Pollination:nn5Dist_s, pred, fam,varb,id,root,data=
  mod7 <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s,
                pred, fam,varb,id,root,data=redata)
  mod8 <- aster(resp ~ -1 + varb + Pollination:nn5DistNotConsumed_s, pred, fam,varb,id,root,data=redata
  AICs <- rep(0, 9)
  AICs[1] <- mod.null$deviance + 2*length(mod.null$coefficients)
  AICs[2] <- mod1$deviance + 2*length(mod1$coefficients)
  AICs[3] <- mod2$deviance + 2*length(mod2$coefficients)
  AICs[4] <- mod3$deviance + 2*length(mod3$coefficients)
  AICs[5] <- mod4$deviance + 2*length(mod4$coefficients)
  AICs[6] <- mod5$deviance + 2*length(mod5$coefficients)
  AICs[7] <- mod6$deviance + 2*length(mod6$coefficients)
  AICs[8] <- mod7$deviance + 2*length(mod7$coefficients)
  AICs[9] <- mod8$deviance + 2*length(mod8$coefficients)
  print(which.min(AICs))
}
```

```
min10 <- sort(data$nn5Dist)[1:10]
max10 <- sort(data$nn5Dist, decreasing = TRUE)[1:10]
```

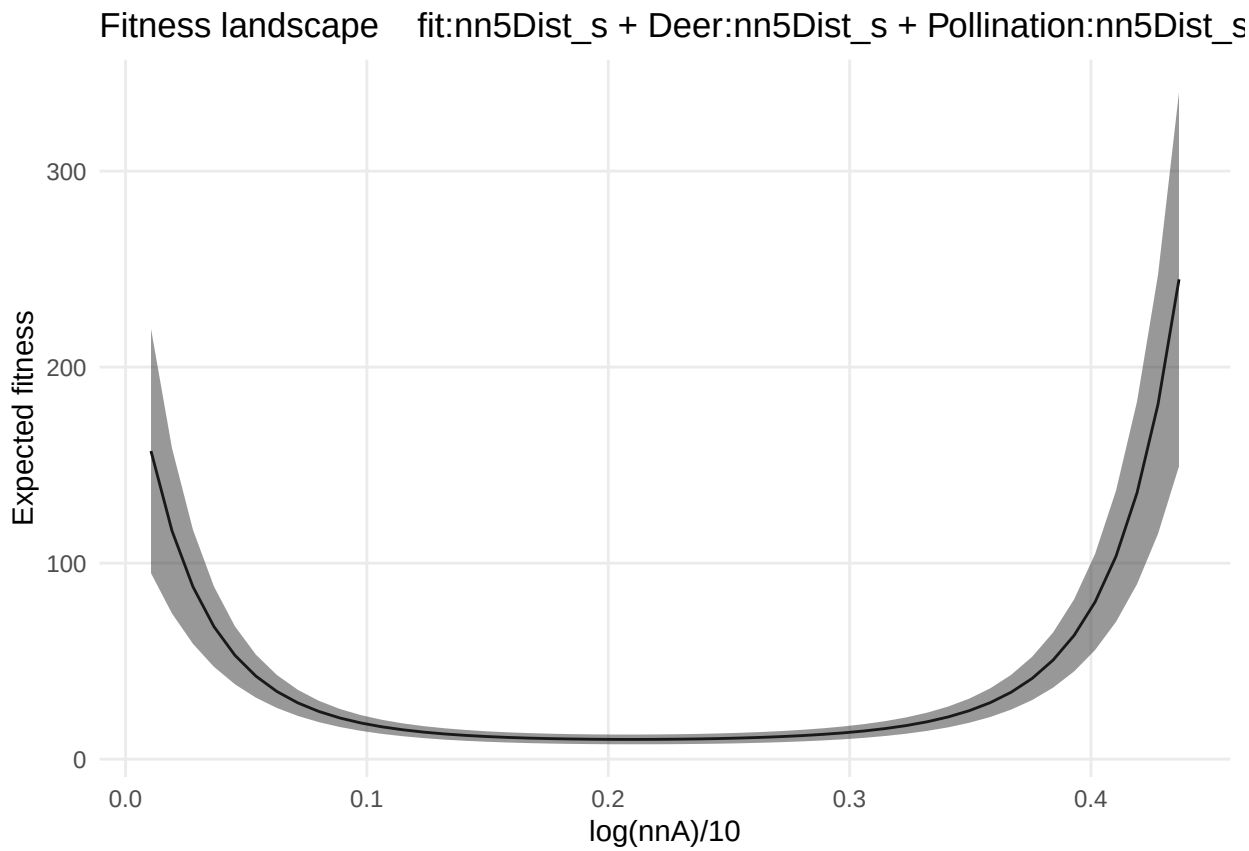
```
data_copy <- data


redata_trimmed <- wide2long(data_copy %>% filter(nn5Dist > quantile(nn5Dist, 0.0125)))
mod <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s,
             pred, fam,varb,id,root,data=redata_trimmed)
fitness_landscape(mod)
```



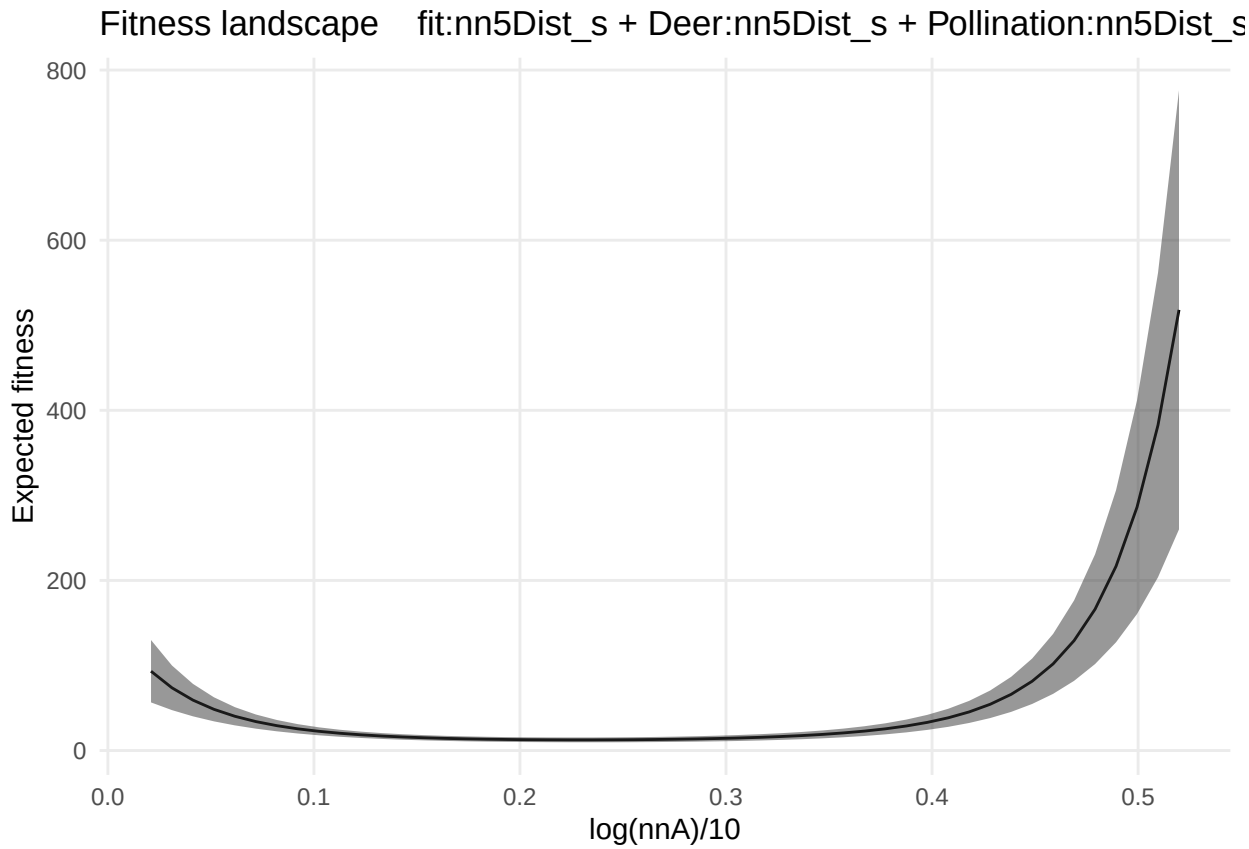

```
redata_trimmed <- wide2long(data_copy %>% filter(nn5Dist < quantile(nn5Dist, 0.9875)))
mod <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s,
             pred, fam,varb,id,root,data=redata_trimmed)
fitness_landscape(mod)
```



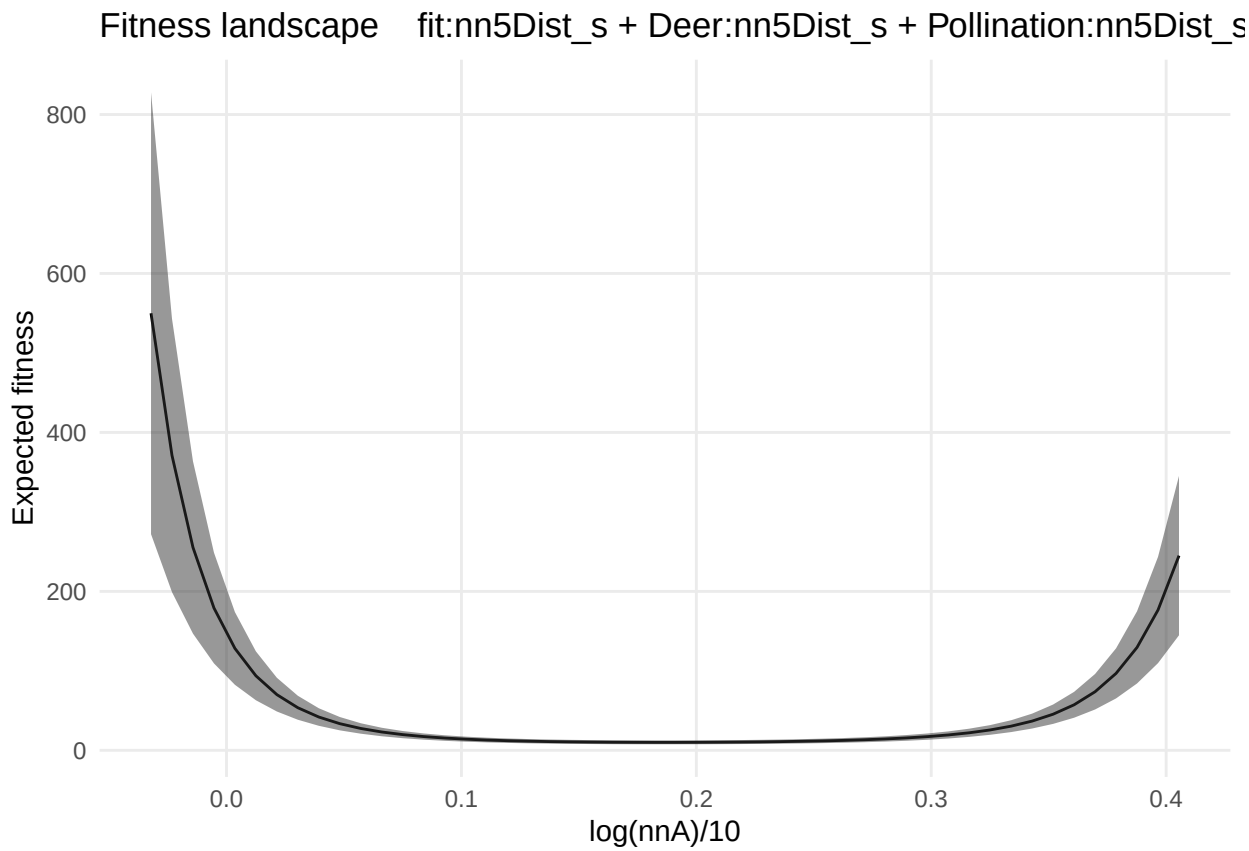

```
redata_trimmed <- wide2long(data_copy %>% filter(nn5Dist > quantile(nn5Dist, 0.0125),
                                                 nn5Dist < quantile(nn5Dist, 0.9875)))
mod <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s,
             pred, fam,varb,id,root,data=redata_trimmed)
fitness_landscape(mod)
```



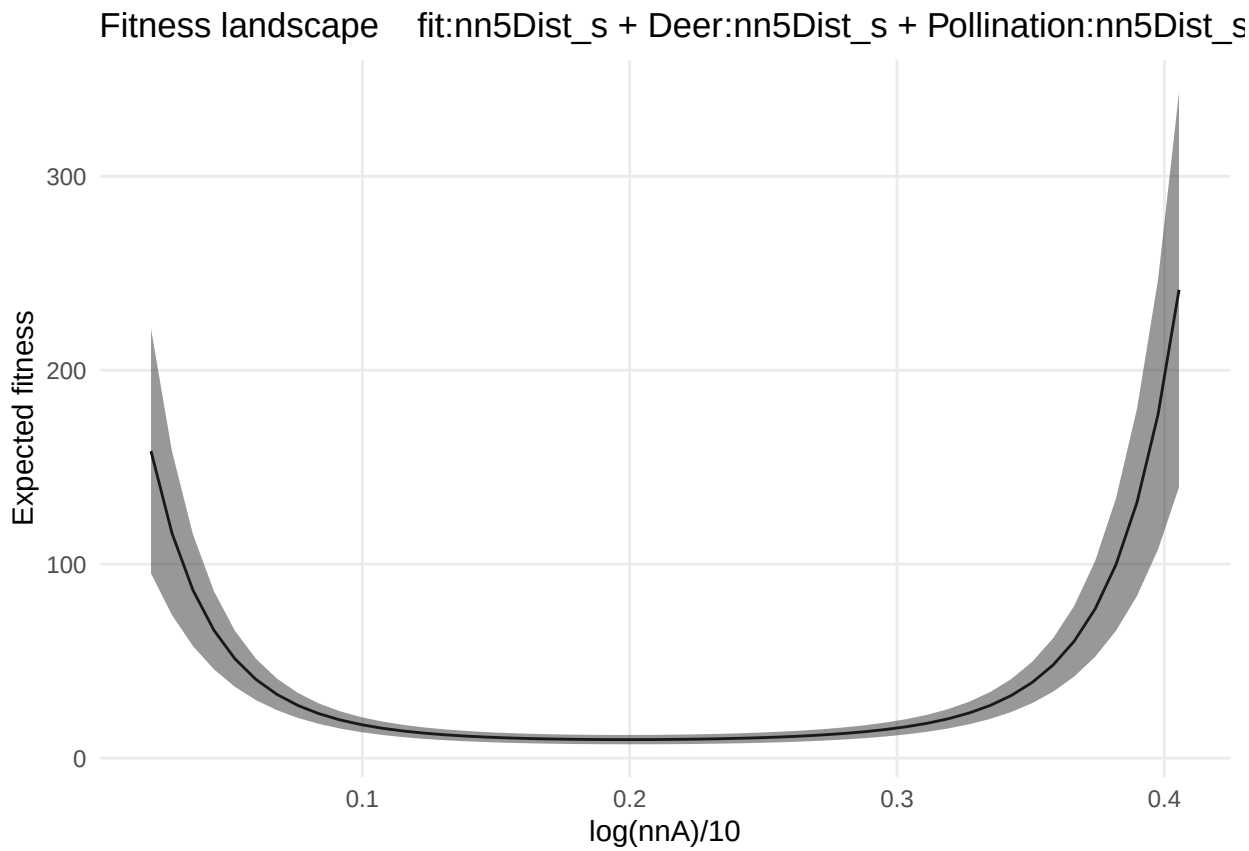

```
redata_trimmed <- wide2long(data_copy %>% filter(nn5Dist > quantile(nn5Dist, 0.025)))
mod <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s,
             pred, fam,varb,id,root,data=redata_trimmed)
fitness_landscape(mod)
```




```
redata_trimmed <- wide2long(data_copy %>% filter(nn5Dist < quantile(nn5Dist, 0.975)))
mod <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s,
             pred, fam,varb,id,root,data=redata_trimmed)
fitness_landscape(mod)
```



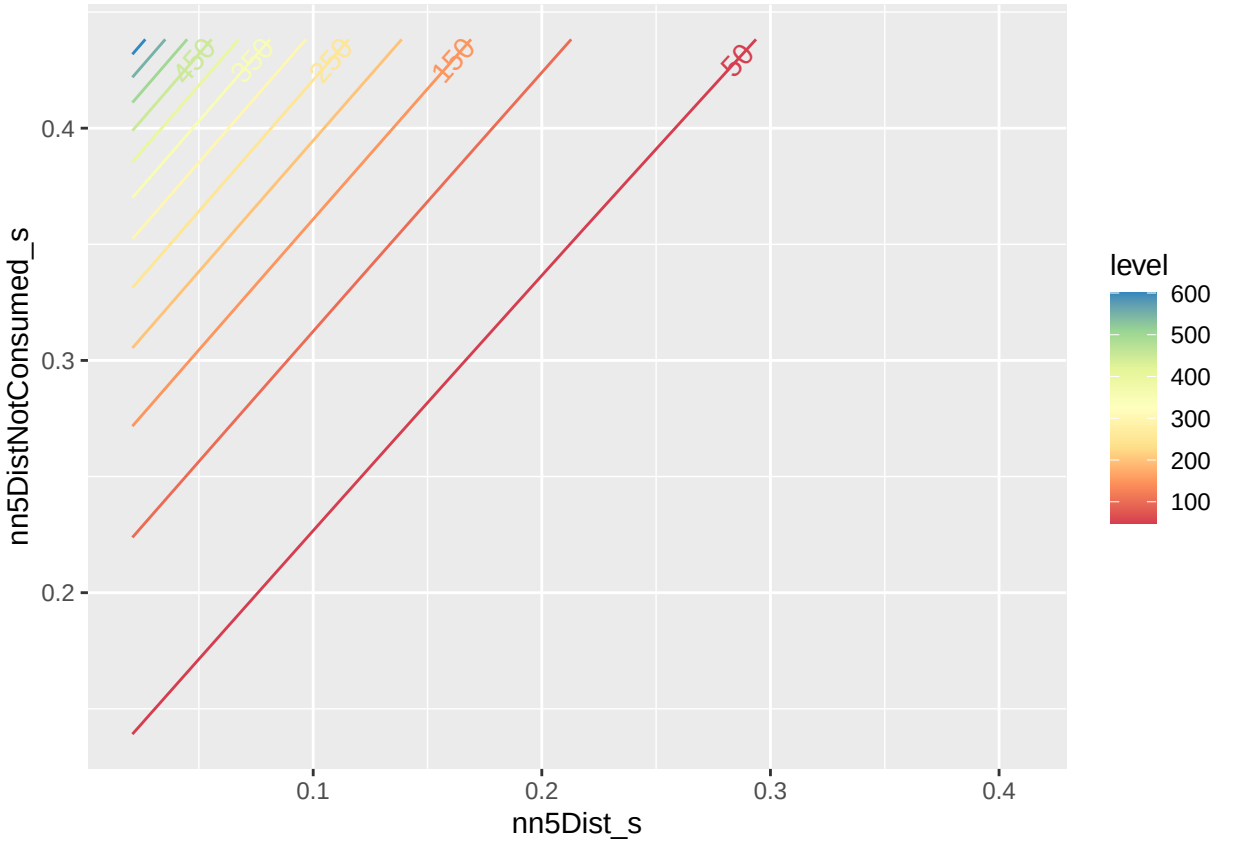

```
redata_trimmed <- wide2long(data_copy %>% filter(nn5Dist > quantile(nn5Dist, 0.025),
                                                 nn5Dist < quantile(nn5Dist, 0.975)))
mod <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s,
             pred, fam,varb,id,root,data=redata_trimmed)
fitness_landscape(mod)
```



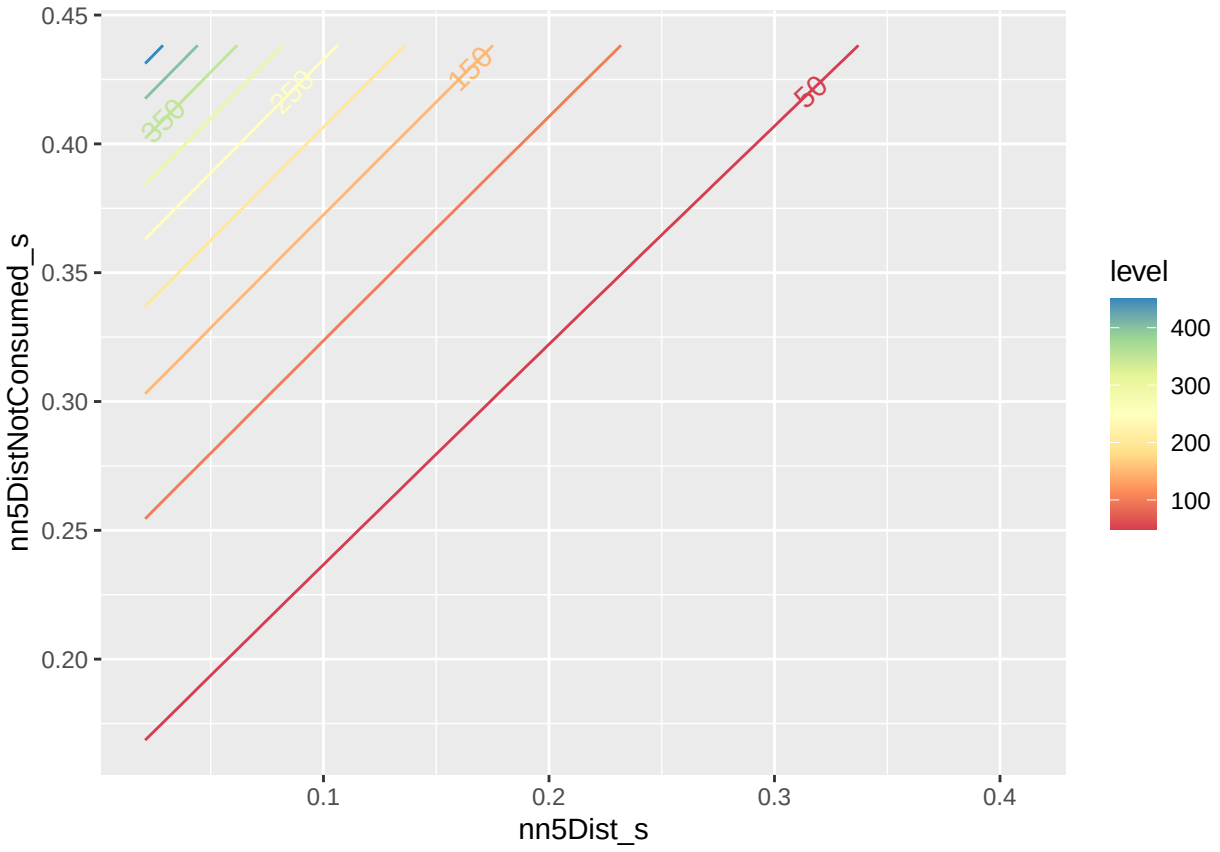

```
aster_AIC(mod)
```

```
## [1] -220864.2
```

```
mod_2d <- aster(resp ~ -1 + varb + fit:nn5Dist_s +Deer:nn5Dist_s + Pollination:nn5DistNotConsumed_s,
                pred, fam,varb,id,root,data=redata_trimmed)
fitness_landscape_2d(mod_2d)
```




```
fitness_landscape_2d(model11)
```



Modelling tail behavior with higher order

```
aster_AIC(model7)
```

```
## [1] -250317.1
```

Quadratic and cubic

```
model_comp <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s +
                      fit:I(nn5Dist_s^2) + Deer:I(nn5Dist_s^2) + Pollination:I(nn5Dist_s^2),
                pred, fam,varb,id,root,data=redata)

#summary(model_comp)
aster_AIC(model_comp)
```

```
## [1] -250561.4
```

```
#summary(model_comp, info.tol = 1e-9)
```

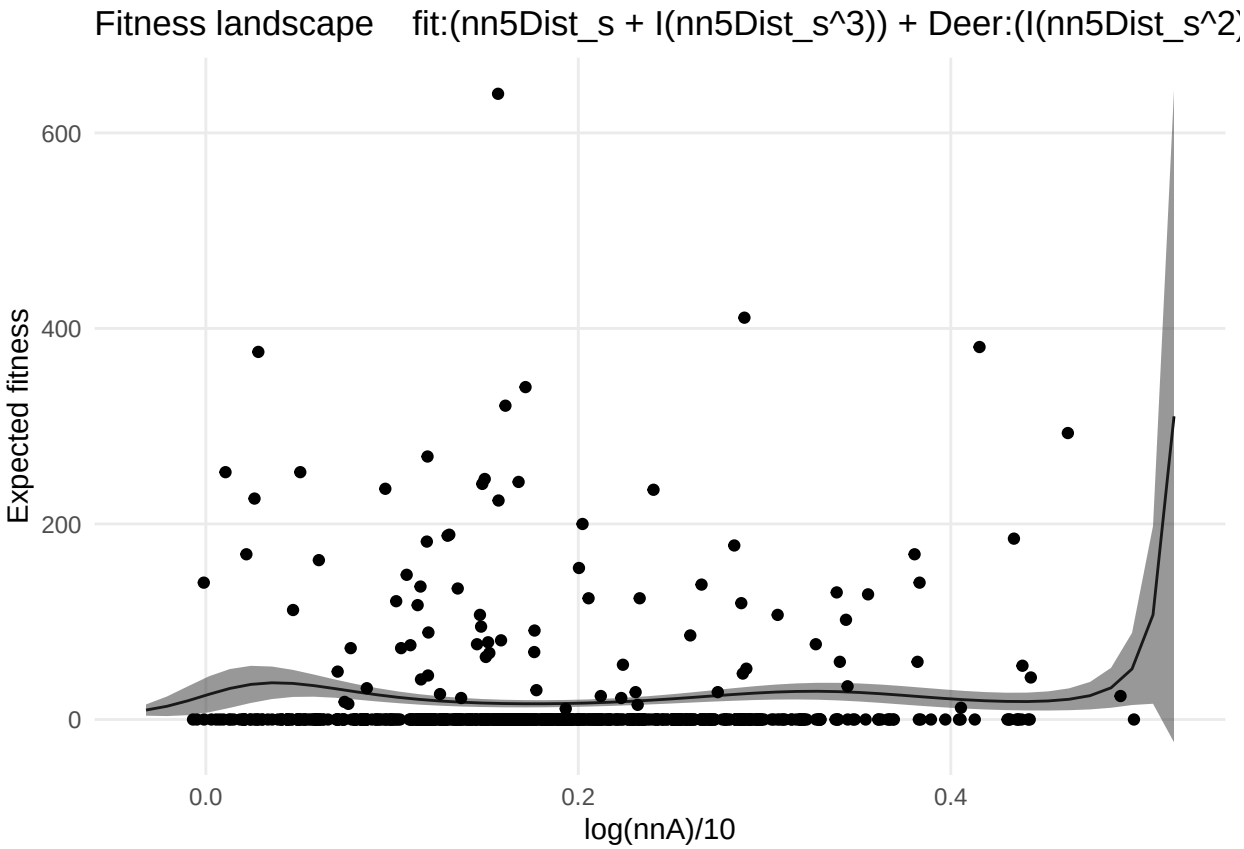
```
model_cubic1 <- aster(resp ~ -1 + varb + fit:(nn5Dist_s+I(nn5Dist_s^3)) + Deer:(I(nn5Dist_s^2)) +
                      Pollination:(nn5Dist_s+I(nn5Dist_s^3) + I(nn5Dist_s^2)) ,
                pred, fam,varb,id,root,data=redata)

#summary(model_comp)
aster_AIC(model_cubic1)
```

```
## [1] -250729.7
```

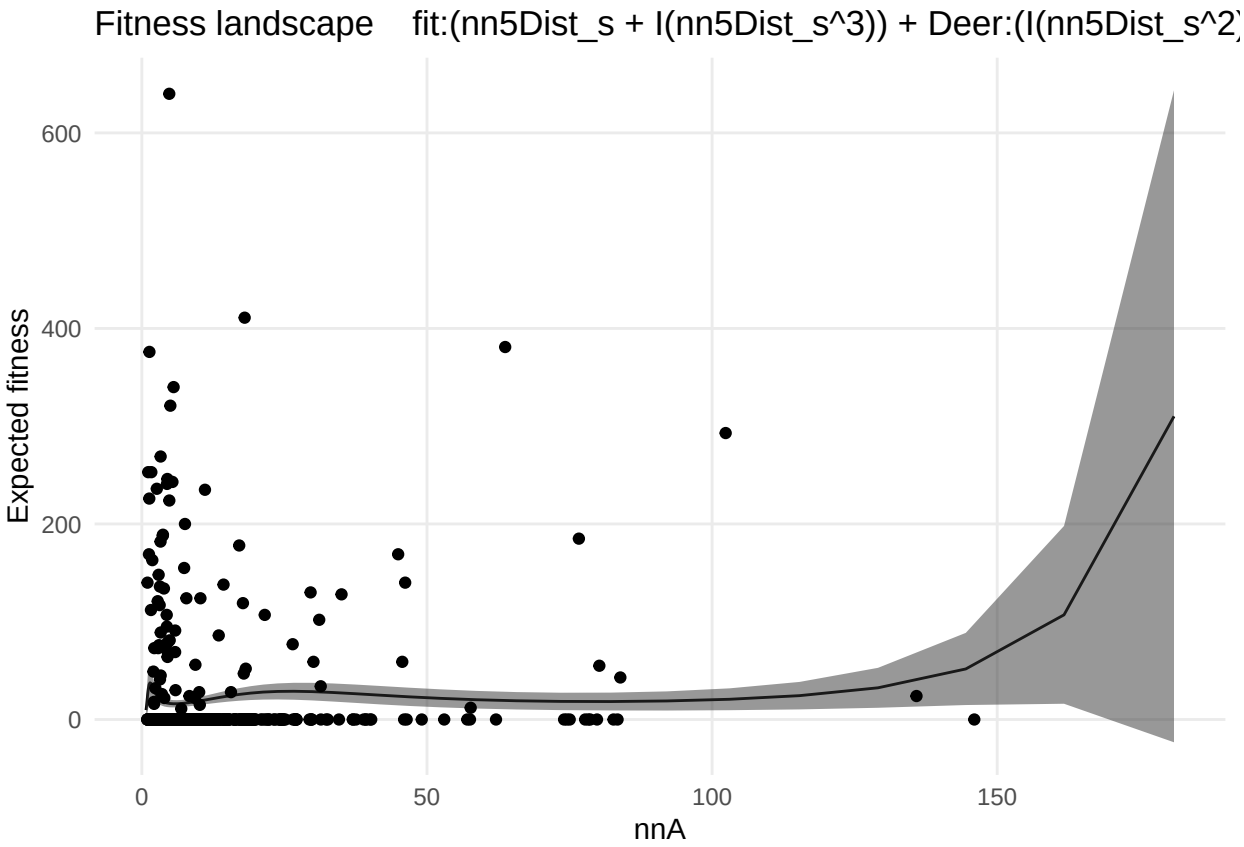
```
summary(model_cubic1, info.tol=1e-8)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + fit:(nn5Dist_s + I(nn5Dist_s^3)) +
##     Deer:(I(nn5Dist_s^2)) + Pollination:(nn5Dist_s + I(nn5Dist_s^3) +
##     I(nn5Dist_s^2)), pred = pred, fam = fam, varvar = varb, idvar = id,
##     root = root, data = redata)
##
##                               Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt                -1.46914    0.17093   -8.595  < 2e-16 ***
## varbembryoCt                  0.18165    0.02737    6.637 3.21e-11 ***
## varbflCt                     -0.24784    0.06063   -4.087 4.36e-05 ***
## varbflCtNotConsumed          -0.15652    0.11913   -1.314   0.1889
## varbflCtUndamaged             1.36641    0.10904   12.531  < 2e-16 ***
## varbisHarvested            -321.76566    1.85444 -173.511  < 2e-16 ***
## varbovuleCt                   5.68510    0.01613  352.505  < 2e-16 ***
## fit:nn5Dist_s                -6.57746    0.24231  -27.145  < 2e-16 ***
## fit:I(nn5Dist_s^3)           14.93002    0.85447   17.473  < 2e-16 ***
## Deer:I(nn5Dist_s^2)           1.90888    1.07027    1.784   0.0745 .
## nn5Dist_s:Pollination         2.37932    0.10939   21.751  < 2e-16 ***
## I(nn5Dist_s^3):Pollination   -0.97868    0.17944   -5.454 4.92e-08 ***
## I(nn5Dist_s^2):Pollination   -2.57625    0.19248  -13.384  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

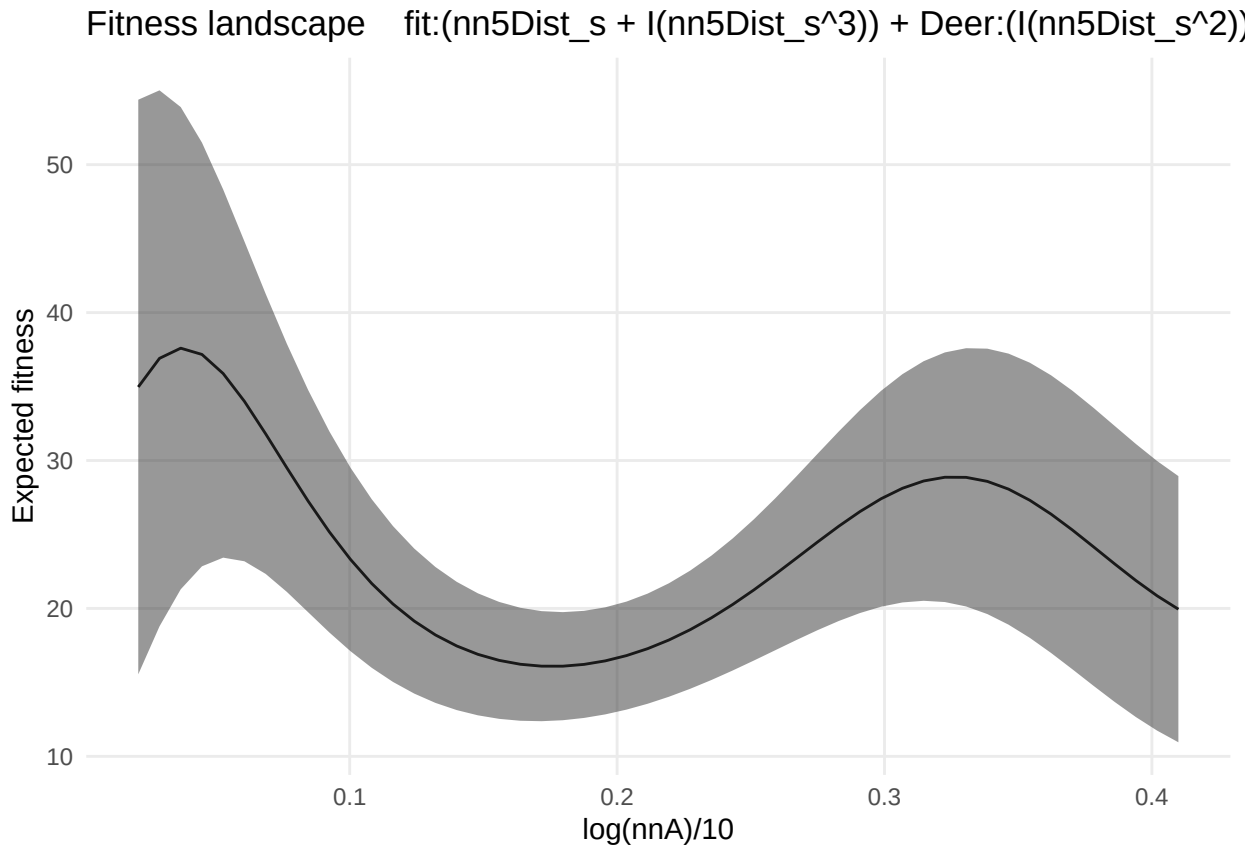
```
fitness_landscape(model_cubic1, observation = TRUE)
```



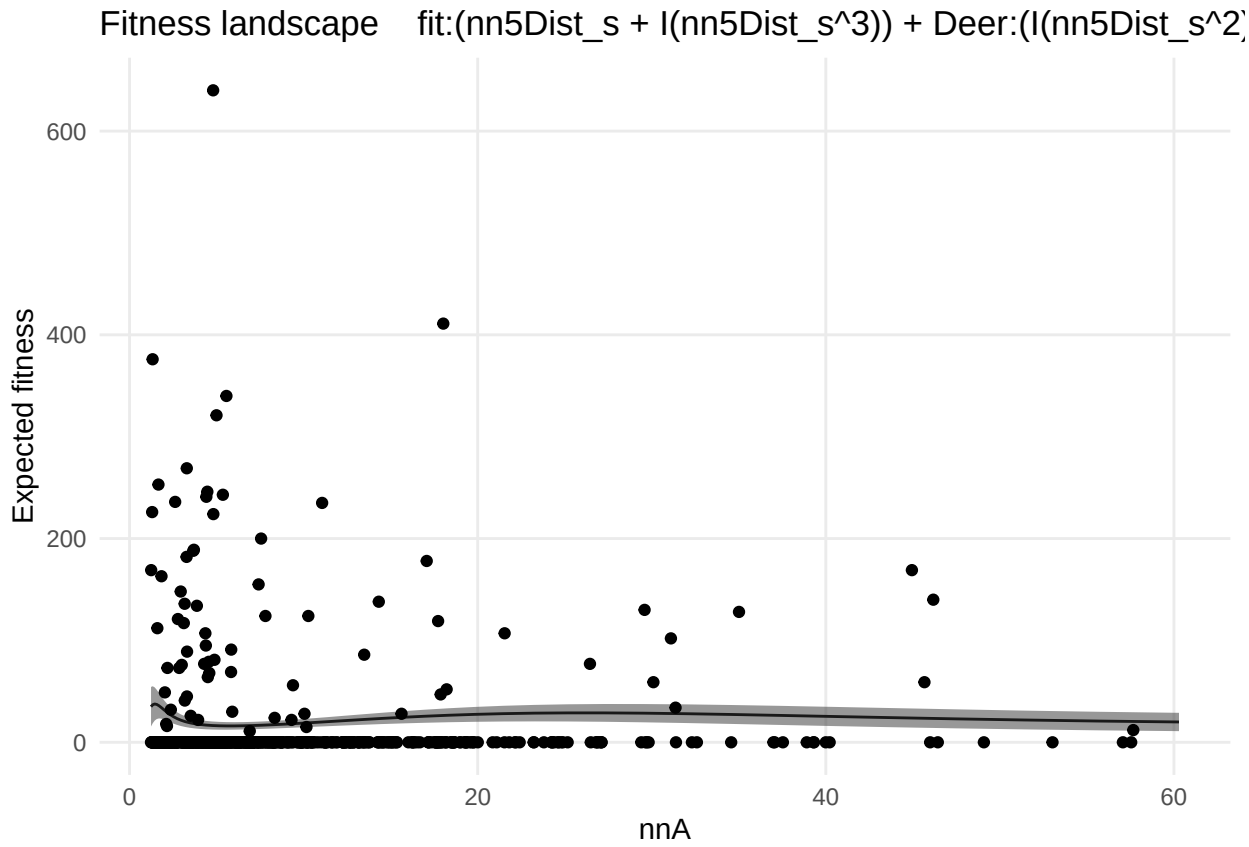

```
fitness_landscape(model_cubic1, observation = TRUE, scale_back = TRUE)
```




```
fitness_landscape(model_cubic1, lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data
```




```
fitness_landscape(model_cubic1,lower = quantile(log(data$nn5Dist)/10, 0.025), upper = quantile(log(data$
```



Piecewise linear

```
model_comp <- aster(resp ~ -1 + varb + fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s +
                      Ltail:nn5Dist_s + Rtail:nn5Dist_s,
                pred, fam,varb,id,root,data=wide2long(data_copy))

#summary(model_comp)
aster_AIC(model_comp)
```

```
## [1] -250362.1
```

```
#summary(model_comp, info.tol=1e-8)
```

```
model_comp <- aster(resp ~ -1 + varb + middle:(fit:nn5Dist_s + Deer:nn5Dist_s + Pollination:nn5Dist_s) +
                      Ltail:(fit:nn5Dist_s) +
                      Rtail:(fit:nn5Dist_s  + Pollination:nn5Dist_s),
                pred, fam,varb,id,root,data=wide2long(data_copy))

#summary(model_comp)
aster_AIC(model_comp)
```

```
## [1] -250446.7
```

```
summary(model_comp, info.tol=1e-11)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + middle:(fit:nn5Dist_s +
##     Deer:nn5Dist_s + Pollination:nn5Dist_s) + Ltail:(fit:nn5Dist_s) +
```

```
##     Rtail:(fit:nn5Dist_s + Pollination:nn5Dist_s), pred = pred,
##     fam = fam, varvar = varb, idvar = id, root = root, data = wide2long(data_copy))
##
##                                Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt                -1.246e+00  1.704e-01   -7.311 2.65e-13 ***
## varbembryoCt                 -2.759e-01  1.505e-02  -18.333  < 2e-16 ***
## varbflCt                     -2.478e-01  6.063e-02   -4.087 4.36e-05 ***
## varbflCtNotConsumed          -2.711e-01  1.461e-01   -1.856   0.0634 .
## varbflCtUndamaged             1.366e+00  1.090e-01   12.531  < 2e-16 ***
## varbisHarvested              -3.208e+02  1.853e+00 -173.148  < 2e-16 ***
## varbovuleCt                   5.912e+00  8.658e-03  682.835  < 2e-16 ***
## middle:fit:nn5Dist_s         -2.196e+00  6.734e-02  -32.604  < 2e-16 ***
## middle:nn5Dist_s:Deer         1.124e+00  5.349e-01    2.102   0.0356 *
## middle:nn5Dist_s:Pollination  5.745e-01  1.802e-02   31.882  < 2e-16 ***
## fit:nn5Dist_s:Ltail          -2.002e-01  1.043e-01   -1.920   0.0548 .
## fit:nn5Dist_s:Rtail          -1.070e+00  1.242e-01   -8.612  < 2e-16 ***
## nn5Dist_s:Pollination:Rtail   2.944e-01  3.262e-02    9.025  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Piecewise quadratic/cubic

```
model_comp <- aster(resp ~ -1 + varb  +
                      Rtail:(fit:(I(nn5Dist_s^2)) + Deer:(nn5Dist_s) +Pollination:( I(nn5Dist_s^2))) +
                      fit:(nn5Dist_s+I(nn5Dist_s^2) ) + Deer:(I(nn5Dist_s^3)) +
                      Pollination:(nn5Dist_s + I(nn5Dist_s^2)+ I(nn5Dist_s^3))+
                      Ltail:(fit:(nn5Dist_s) + Pollination:(nn5Dist_s + I(nn5Dist_s^2))),
                pred, fam,varb,id,root,data=wide2long(data_copy))

#summary(model_comp)
aster_AIC(model_comp)
```

```
## [1] -250902
```

```
summary(model_comp, info.tol=1e-12)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + Rtail:(fit:(I(nn5Dist_s^2)) +
##     Deer:(nn5Dist_s) + Pollination:(I(nn5Dist_s^2))) + fit:(nn5Dist_s +
##     I(nn5Dist_s^2)) + Deer:(I(nn5Dist_s^3)) + Pollination:(nn5Dist_s +
##     I(nn5Dist_s^2) + I(nn5Dist_s^3)) + Ltail:(fit:(nn5Dist_s) +
##     Pollination:(nn5Dist_s + I(nn5Dist_s^2))), pred = pred, fam = fam,
##     varvar = varb, idvar = id, root = root, data = wide2long(data_copy))
##
##                                   Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt                     -1.46619    0.17116   -8.566  < 2e-16 ***
## varbembryoCt                       0.16445    0.03234    5.085 3.68e-07 ***
## varbflCt                          -0.24784    0.06063   -4.087 4.36e-05 ***
## varbflCtNotConsumed               -0.16087    0.11544   -1.394  0.16346
## varbflCtUndamaged                  1.36641    0.10904   12.531  < 2e-16 ***
## varbisHarvested                 -321.75614    1.85486 -173.466  < 2e-16 ***
## varbovuleCt                        5.69465    0.01807  315.077  < 2e-16 ***
## fit:nn5Dist_s                     -6.58973    0.37454  -17.594  < 2e-16 ***
## fit:I(nn5Dist_s^2)                 3.52573    0.69283    5.089 3.60e-07 ***
```

```
## Deer:I(nn5Dist_s^3)                 8.62520    3.58922    2.403  0.01626 *
## nn5Dist_s:Pollination               2.32938    0.14591   15.965  < 2e-16 ***
## I(nn5Dist_s^2):Pollination         -3.02116    0.35679   -8.468  < 2e-16 ***
## Pollination:I(nn5Dist_s^3)          1.57453    0.26162    6.018 1.76e-09 ***
## Rtail:fit:I(nn5Dist_s^2)            5.49322    0.39937   13.755  < 2e-16 ***
## Rtail:Deer:nn5Dist_s               -1.62396    0.95243   -1.705  0.08818 .
## Rtail:I(nn5Dist_s^2):Pollination   -1.13139    0.08749  -12.932  < 2e-16 ***
## fit:nn5Dist_s:Ltail                90.00042   14.05536    6.403 1.52e-10 ***
## nn5Dist_s:Pollination:Ltail       -30.75399    4.61850   -6.659 2.76e-11 ***
## I(nn5Dist_s^2):Pollination:Ltail -271.39186   94.95114   -2.858  0.00426 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Third order model on trimmed data

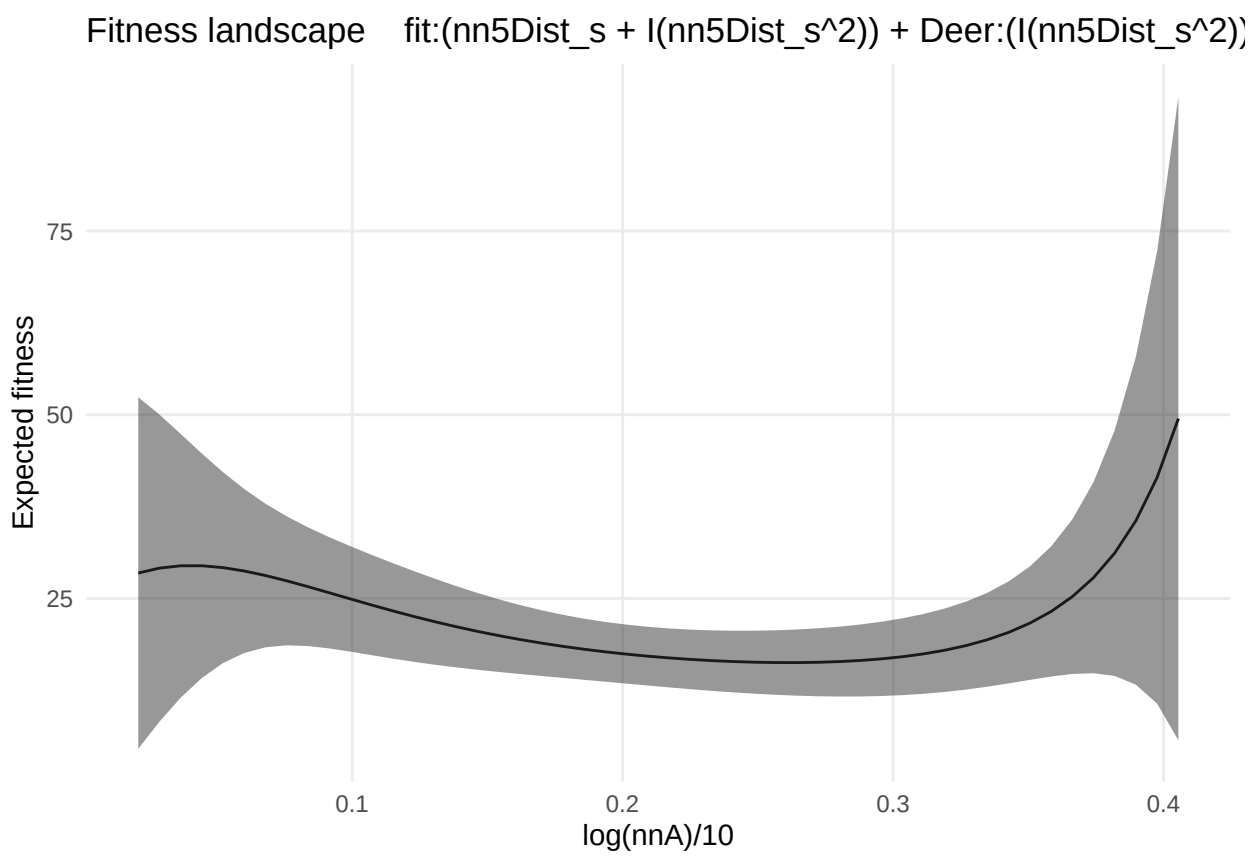
```
model_cubic2 <- aster(resp ~ -1 + varb + fit:(nn5Dist_s+I(nn5Dist_s^2)) +
                      Deer:(I(nn5Dist_s^2)) +
                      Pollination:(nn5Dist_s+I(nn5Dist_s^2) + I(nn5Dist_s^3)) ,
                pred, fam,varb,id,root,data=redata_trimmed)

#summary(model_comp)
aster_AIC(model_cubic2)
```

```
## [1] -221215.5
```

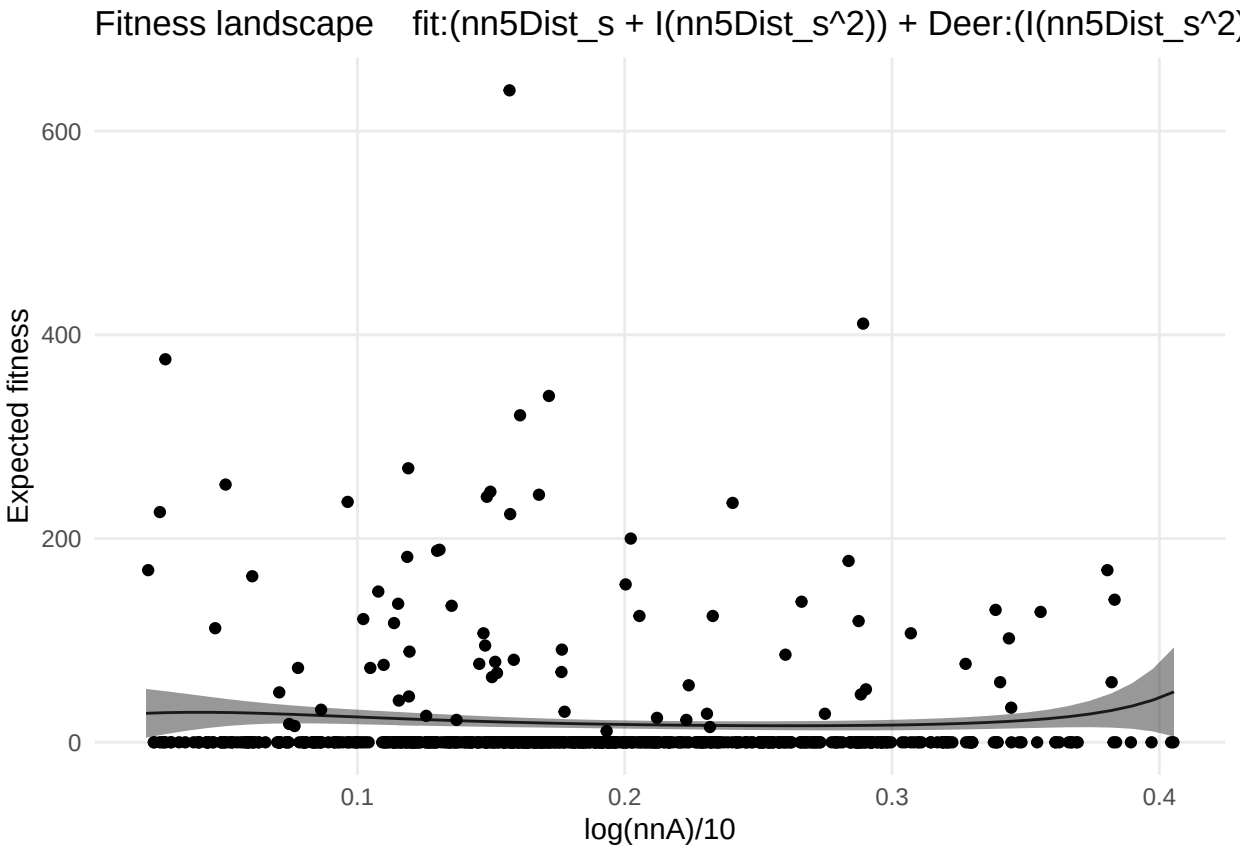
```
summary(model_cubic2, info.tol=1e-8)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + fit:(nn5Dist_s + I(nn5Dist_s^2)) +
##     Deer:(I(nn5Dist_s^2)) + Pollination:(nn5Dist_s + I(nn5Dist_s^2) +
##     I(nn5Dist_s^3)), pred = pred, fam = fam, varvar = varb, idvar = id,
##     root = root, data = redata_trimmed)
##
##                               Estimate Std. Error  z value Pr(>|z|)
## varbcapsuleCt                -1.58818    0.18083   -8.783  < 2e-16 ***
## varbembryoCt                  0.27456    0.04624    5.937 2.90e-09 ***
## varbflCt                     -0.22088    0.06143   -3.596 0.000324 ***
## varbflCtNotConsumed          -0.19715    0.12551   -1.571 0.116233
## varbflCtUndamaged             1.33670    0.11091   12.052  < 2e-16 ***
## varbisHarvested            -318.25092    1.94989 -163.215  < 2e-16 ***
## varbovuleCt                   5.62300    0.02657  211.621  < 2e-16 ***
## fit:nn5Dist_s                -8.31687    0.73023  -11.389  < 2e-16 ***
## fit:I(nn5Dist_s^2)            7.20557    1.65154    4.363 1.28e-05 ***
## I(nn5Dist_s^2):Deer           3.03581    1.39050    2.183 0.029017 *
## nn5Dist_s:Pollination         3.03005    0.29448   10.290  < 2e-16 ***
## I(nn5Dist_s^2):Pollination   -5.19911    0.93836   -5.541 3.01e-08 ***
## Pollination:I(nn5Dist_s^3)    3.41624    0.81139    4.210 2.55e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

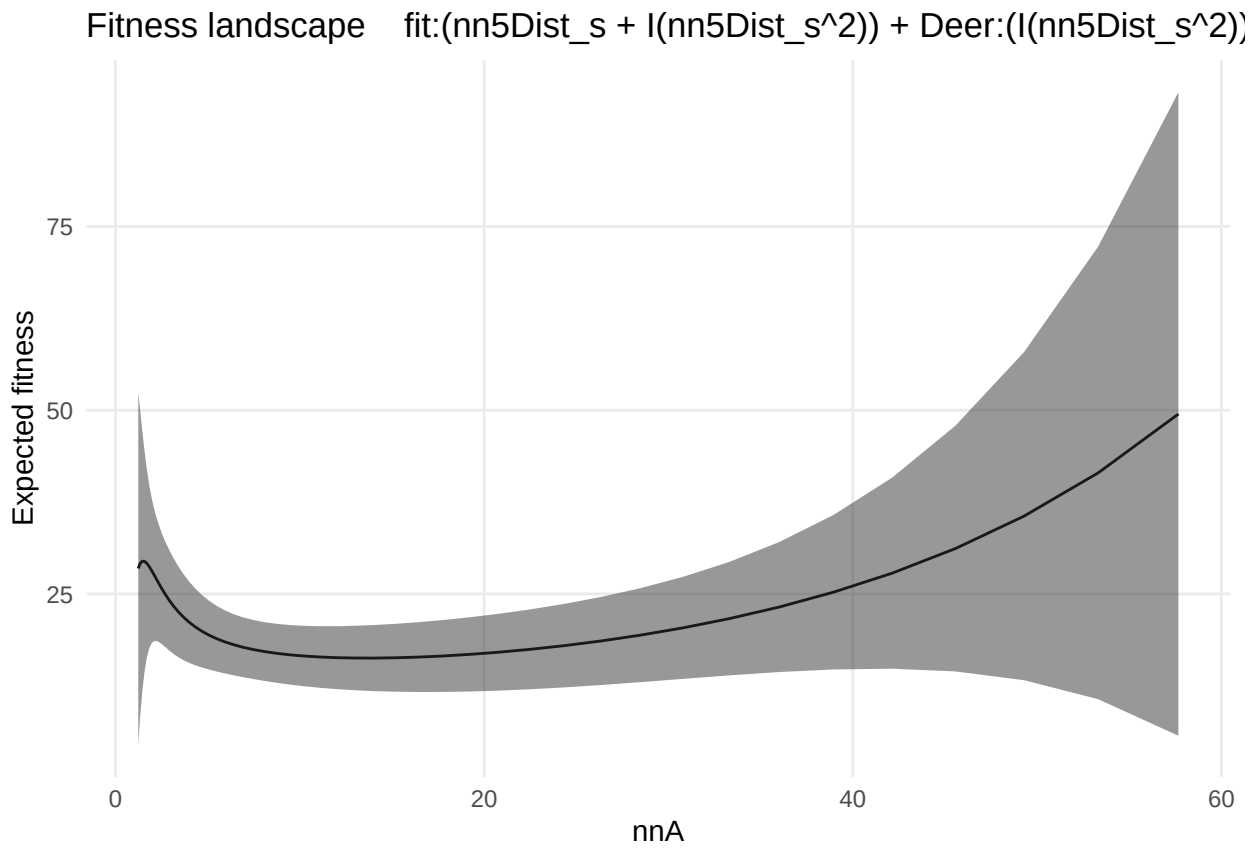
```
fitness_landscape(model_cubic2)
```



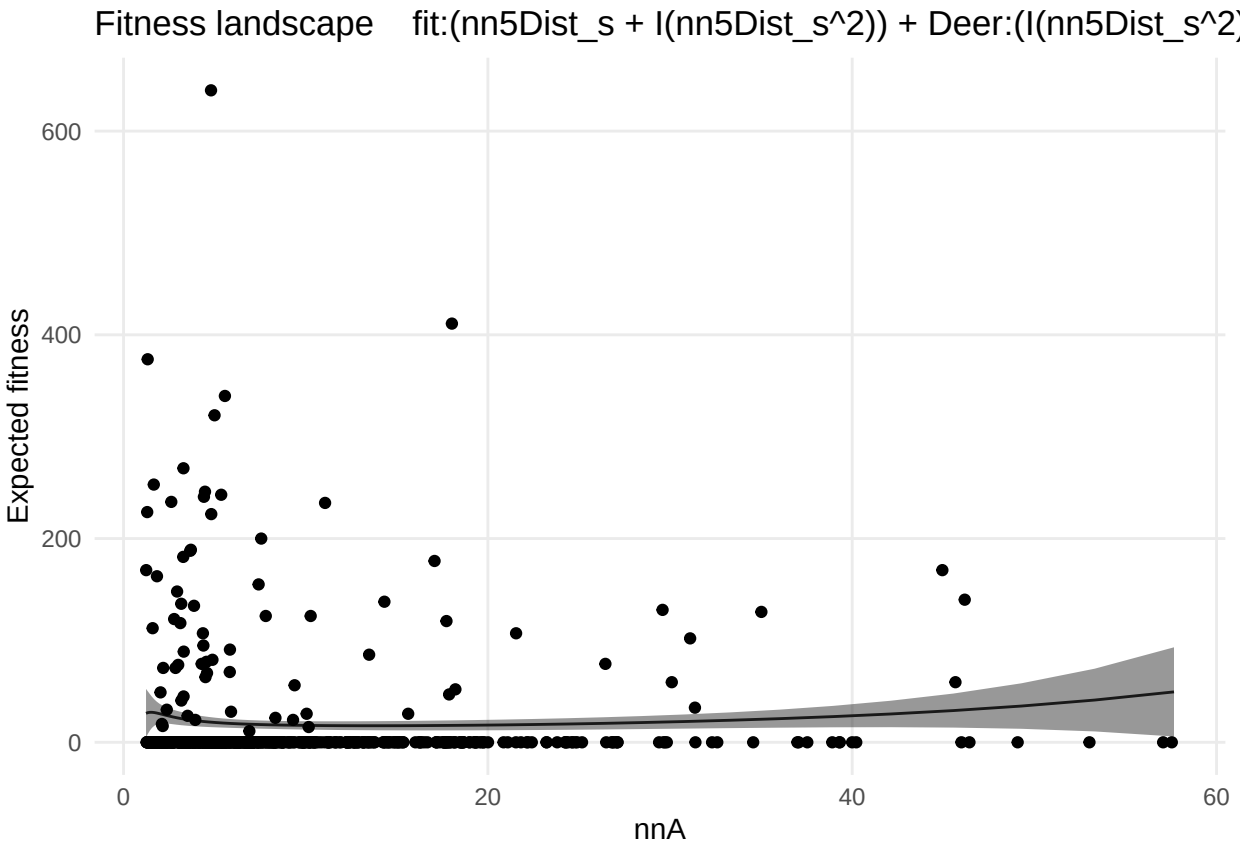

```
fitness_landscape(model_cubic2, observation = TRUE)
```




```
fitness_landscape(model_cubic2, scale_back = TRUE)
```




```
fitness_landscape(model_cubic2, observation = TRUE, scale_back = TRUE)
```



Conditional landscapes

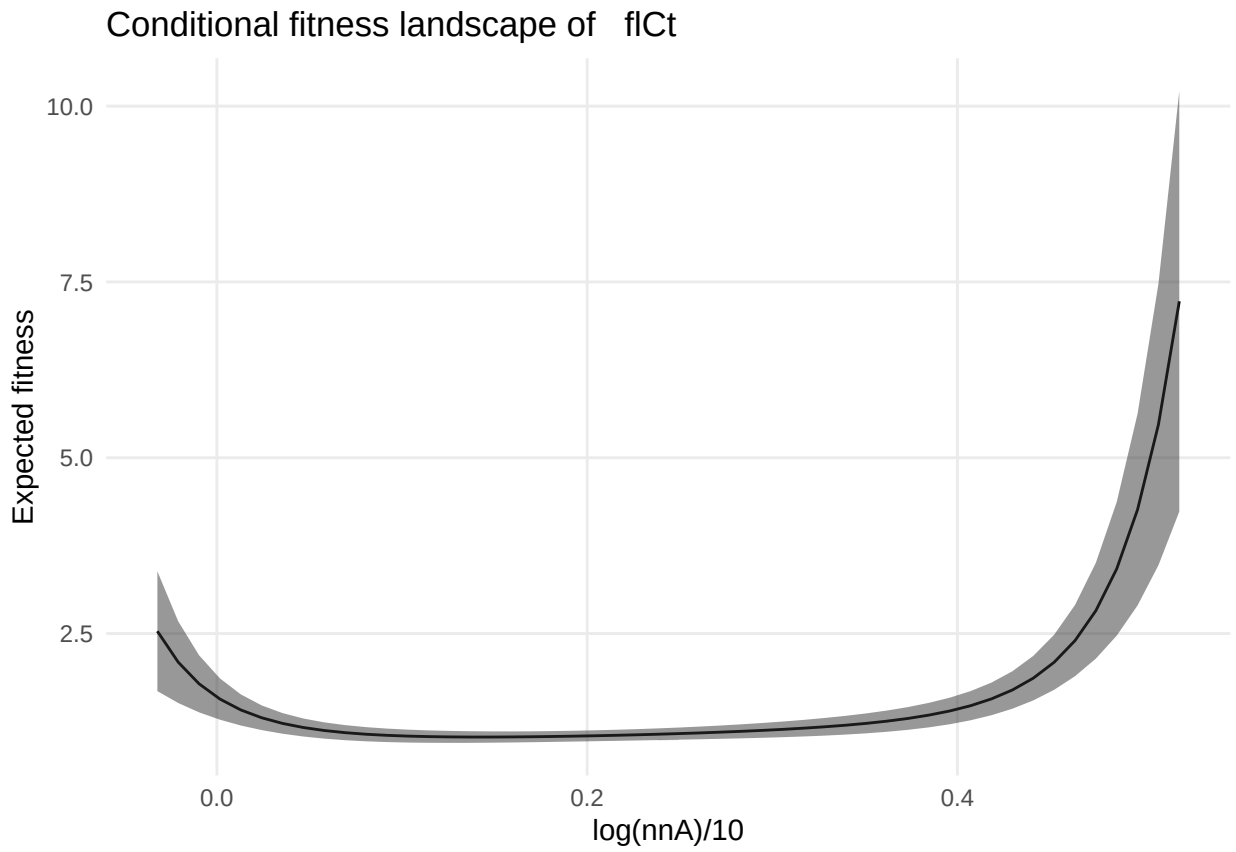
According to Jared and Prof. Wagenius, the U-shape landscape is somehow against biological intuitive. Let's see conditonal landscape of each node.

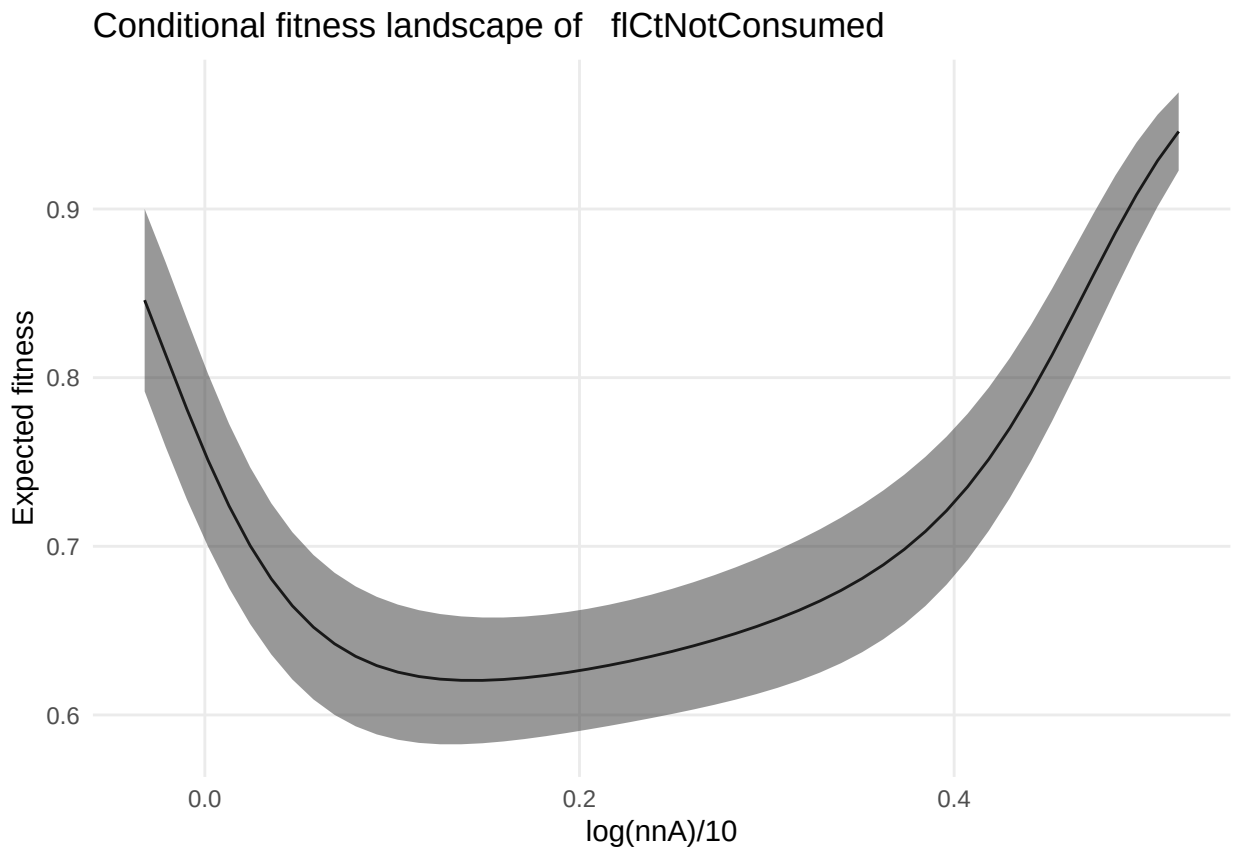
```
conditional_fitness_landscape <- function(model, covariate = 'nn5Dist_s', lower = NULL, upper = NULL,
                                observation=FALSE, scale_back = FALSE) {
  # Make fake individuals
  nInd <- 50
  lwr <- if (is.null(lower)) min(unique(model$data[,covariate])) else lower
  upr <- if (is.null(upper)) max(unique(model$data[,covariate])) else upper
  cand.nnA <- seq(from = lwr, to = upr, length = nInd)
  cand <- as.data.frame(cand.nnA)
  colnames(cand) <- covariate
  cand$root <- 1
  blah <- data[1:nInd, colnames(data) %in% vars]
  cand <- cbind(cand, blah)
  cand$id <- data[1:nInd, 'id']

  # Transform fake data into long format
  cand_long <- reshape(cand, varying = list(vars), direction="long", timevar="varb",
                  times = as.factor(vars), v.names="resp")

  cand_long <- data.frame(cand_long)
  cand_long$fit <- as.numeric(cand_long$varb == "embryoCt")
```

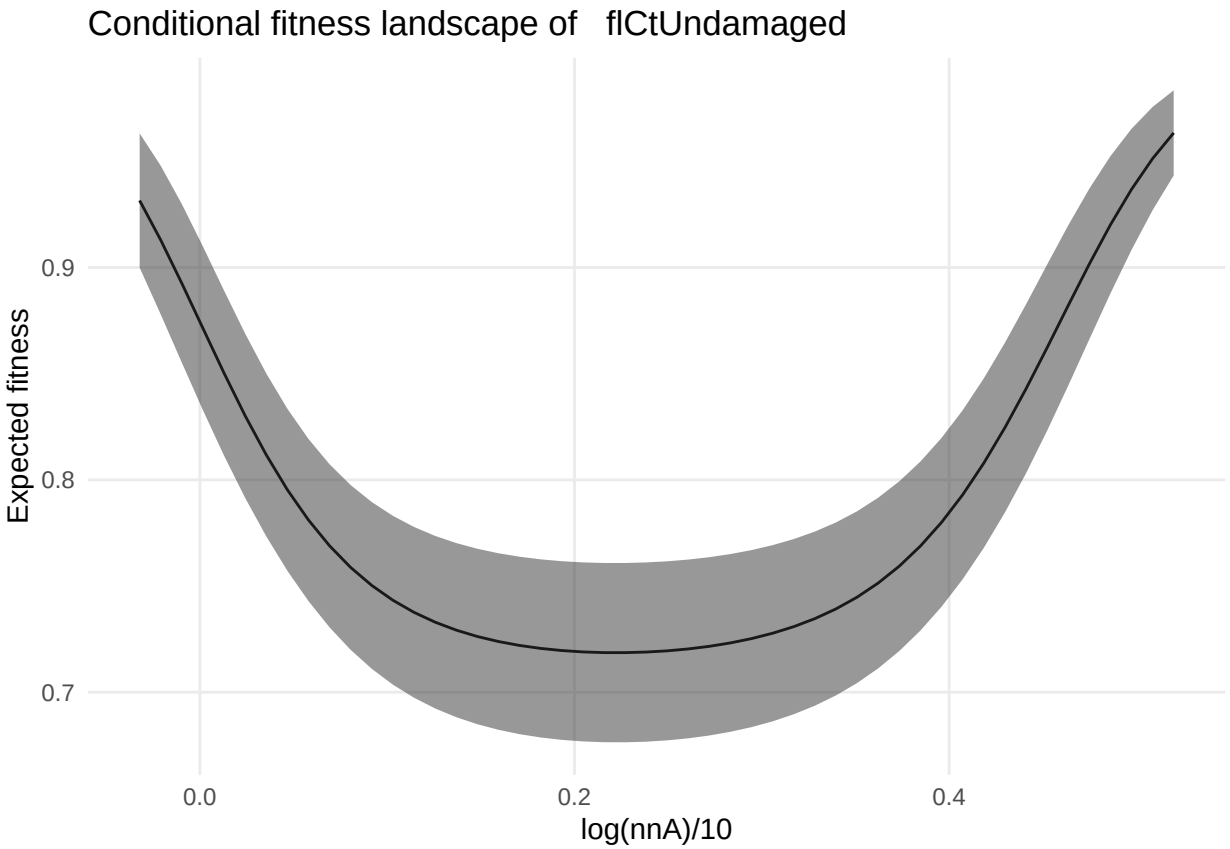
```r
cand_long$Nid <- as.numeric(gsub("[^0-9.-]", "", cand_long$id))

cand_long$Deer <- as.numeric(cand_long$varb=='flCtNotConsumed')
cand_long$Pollination <- as.numeric(is.element(cand_long$varb,
                              c("capsuleCt", "isHarvested", "ovuleCt", "embryoCt")))
cand_long$Rtail <- as.numeric(cand_long$nn5Dist_s > quantile(cand_long$nn5Dist_s, 0.975))
cand_long$Ltail <- as.numeric(cand_long$nn5Dist_s < quantile(cand_long$nn5Dist_s, 0.025))

# Get conditional mean value parameters
pred <- predict(model, cand_long, varvar=varb, idvar=id, root=root,
                    se.fit = TRUE, model.type='conditional',
                is.always.parameter = TRUE, info.tol=1e-8)

xi_parm <- pred$fit
xi_parm_se <- pred$se.fit

names(xi_parm) <- paste0(cand_long$id,'.',cand_long$varb)
names(xi_parm_se) <- paste0(cand_long$id,'.',cand_long$varb)

conditional_exp <- matrix(xi_parm, nrow=50)
rownames(conditional_exp) <- cand$id
colnames(conditional_exp) <- paste0(vars,'.exp')

conditional_exp_se <- matrix(xi_parm_se, nrow=50)
rownames(conditional_exp_se) <- cand$id
colnames(conditional_exp_se) <- paste0(vars,'.exp_se')


cand_block <- cbind(cand, conditional_exp, conditional_exp_se)

xlabel = if (covariate == 'nn5Dist_s') 'log(nnA)/10' else 'log(nnB)/10'

#cand_block <- cand %>% mutate(xi_parm[1:50], xi_parm, lower = xi_parm[1:50] - 2 * xi_parm_se[1:50],
if (scale_back == TRUE) {
  cand_block[, as.character(covariate)] <- exp(10*cand_block[,as.character(covariate)])
  xlabel = if (covariate == 'nn5Dist_s') 'nnA' else 'nnB'
}
#cand_block <- cand_block[1:40,]
for (node in vars) {
  plt <- ggplot(data = cand_block) + geom_line(mapping = aes(x = cand_block[,as.character(covariate)]
  if (observation == TRUE) {
    obs <- test %>% filter((!!sym(covariate)) > lwr & (!!sym(covariate)) < upr)
    if (scale_back == TRUE) {
      plt <- plt + geom_point(data=obs, mapping = aes(x =exp(10*obs[,as.character(covariate)]),
                                                      y = obs[,'fecundity']))
    } else {
      plt <- plt + geom_point(data=obs, mapping = aes(x =obs[,as.character(covariate)], y = obs[,'fecu
    }
  }
  plt <- plt + labs(x=xlabel, y="Expected fitness", title = paste0("Conditional fitness landscape of
    #annotate(geom='text', x=c(0, 0.1, 0.3, 0.5), y=750, label=c('nnA', '2.718282', '20.085537', '148
    theme_minimal() + scale_x_continuous(minor_breaks = NULL) +scale_y_continuous(minor_breaks = NULL
```

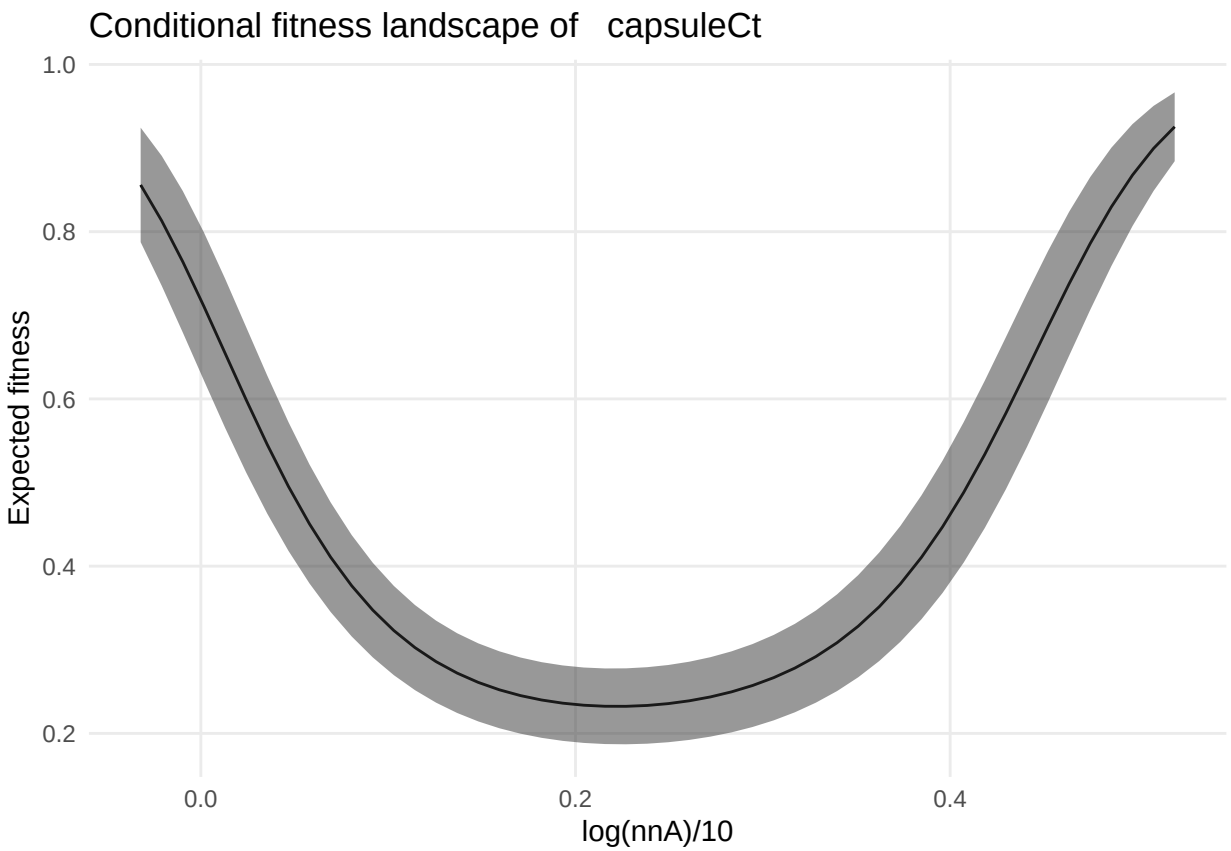
```
    print(plt)
  }


}
```

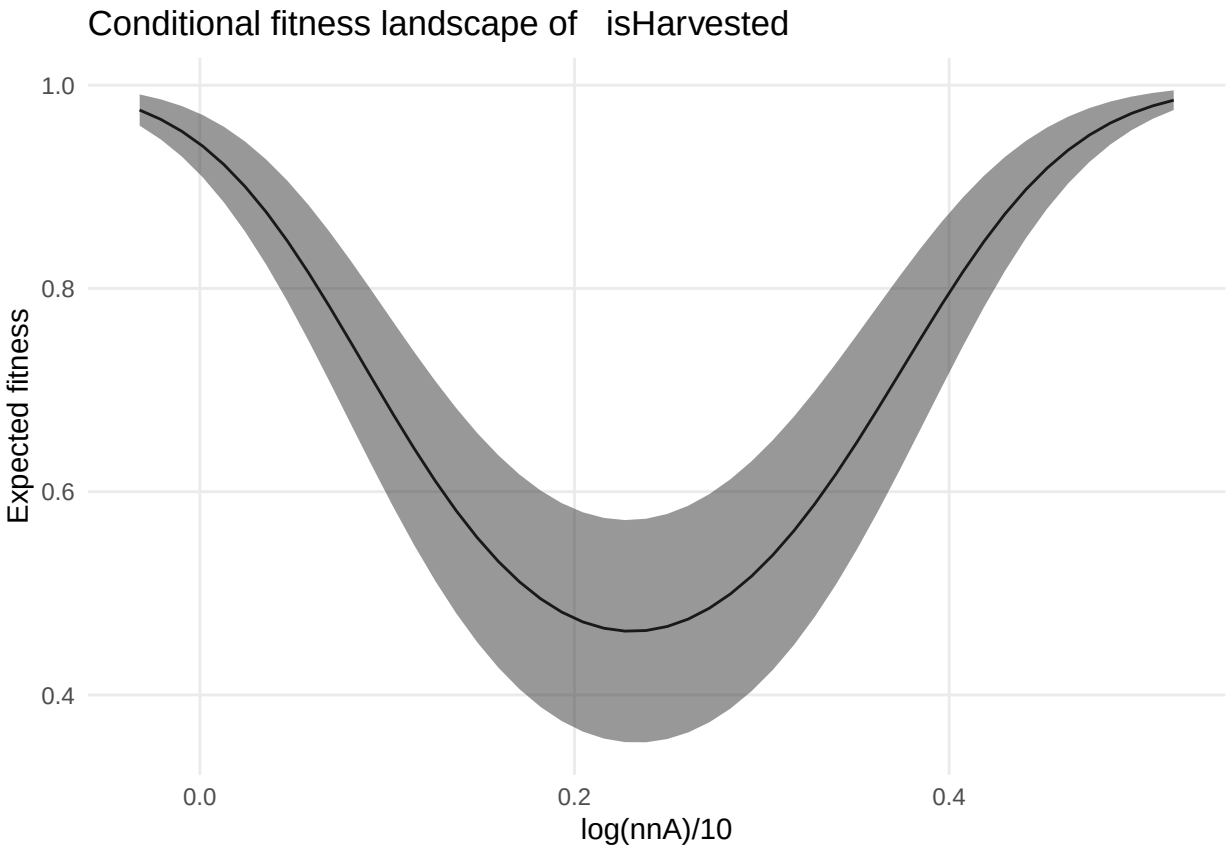
```
conditional_fitness_landscape(model7)
```

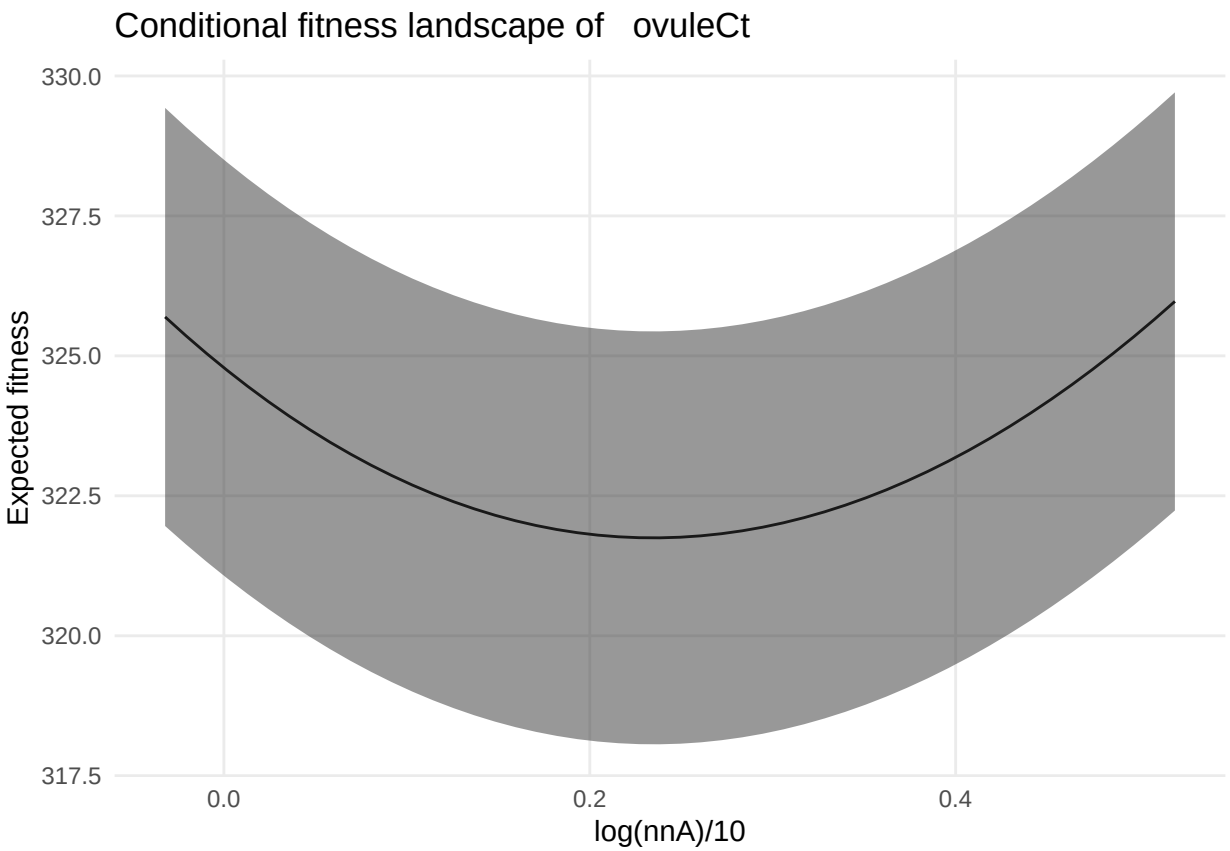


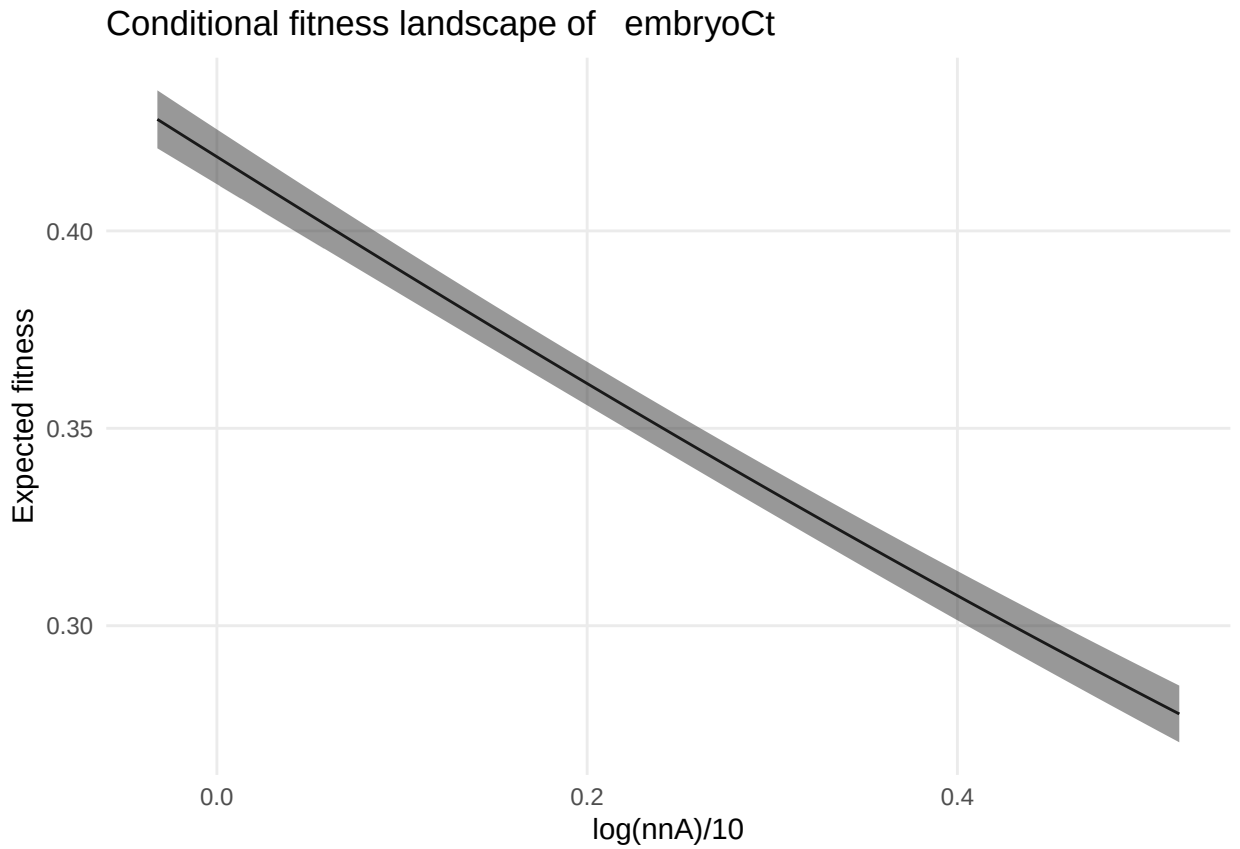

Conditional fitness landscape of flCtNotConsumed
Expected fitness
0.9
0.8
0.7
0.6
0.0
0.2
0.4
log(nnA)/10


Conditional fitness landscape of flCtUndamaged
Expected fitness
0.9
0.8
0.7
0.0
0.2
0.4
log(nnA)/10

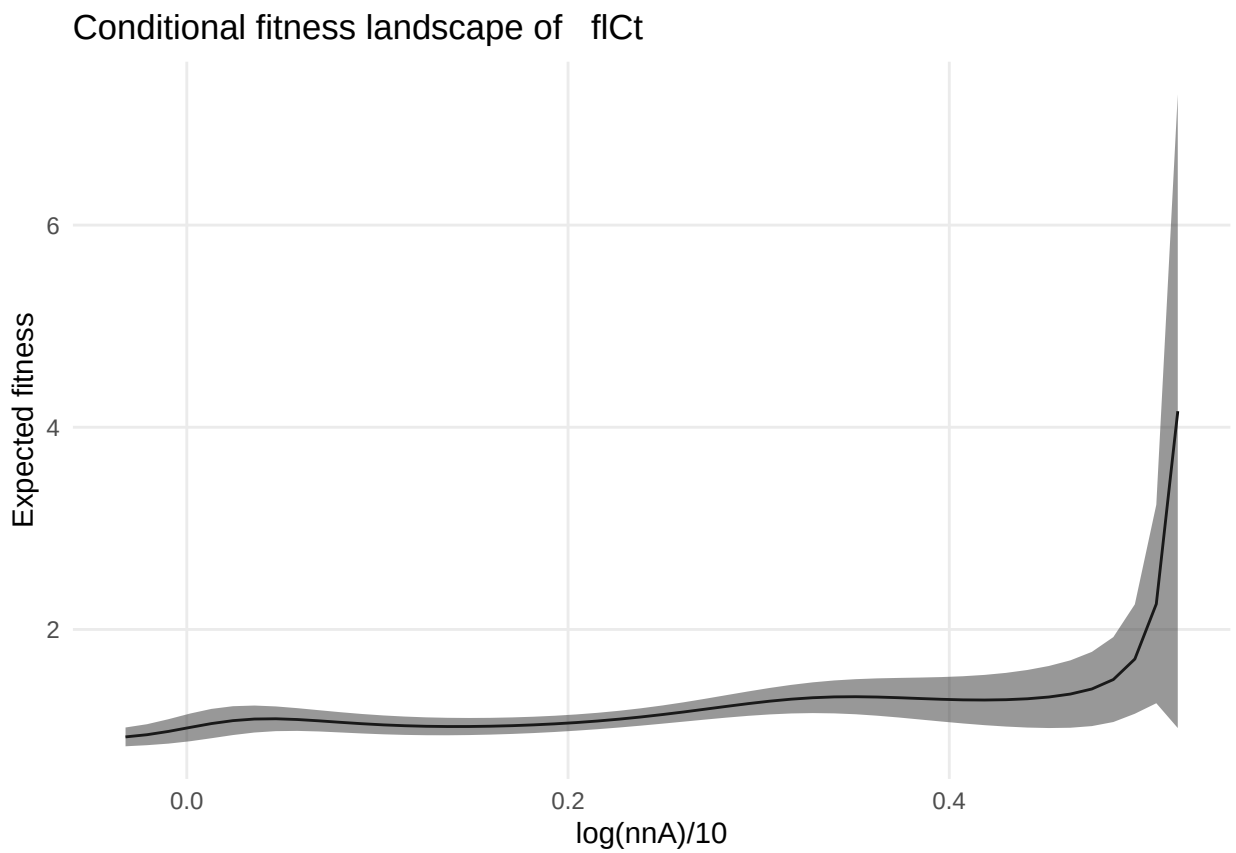

Conditional fitness landscape of capsuleCt
Expected fitness
1.0
0.8
0.6
0.4
0.2
0.0
0.2
0.4
log(nnA)/10

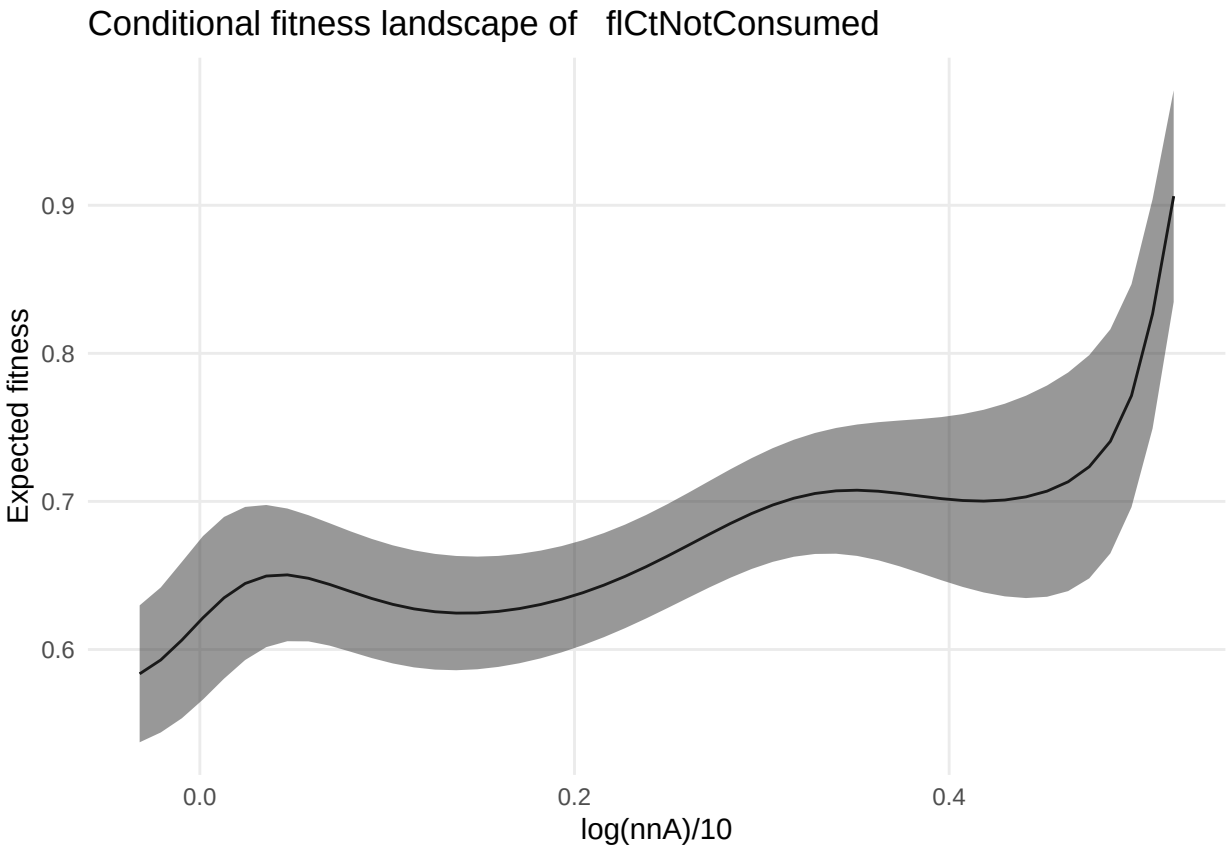

Conditional fitness landscape of isHarvested
1.0
0.8
0.6
0.4
Expected fitness
0.0
0.2
0.4
log(nnA)/10

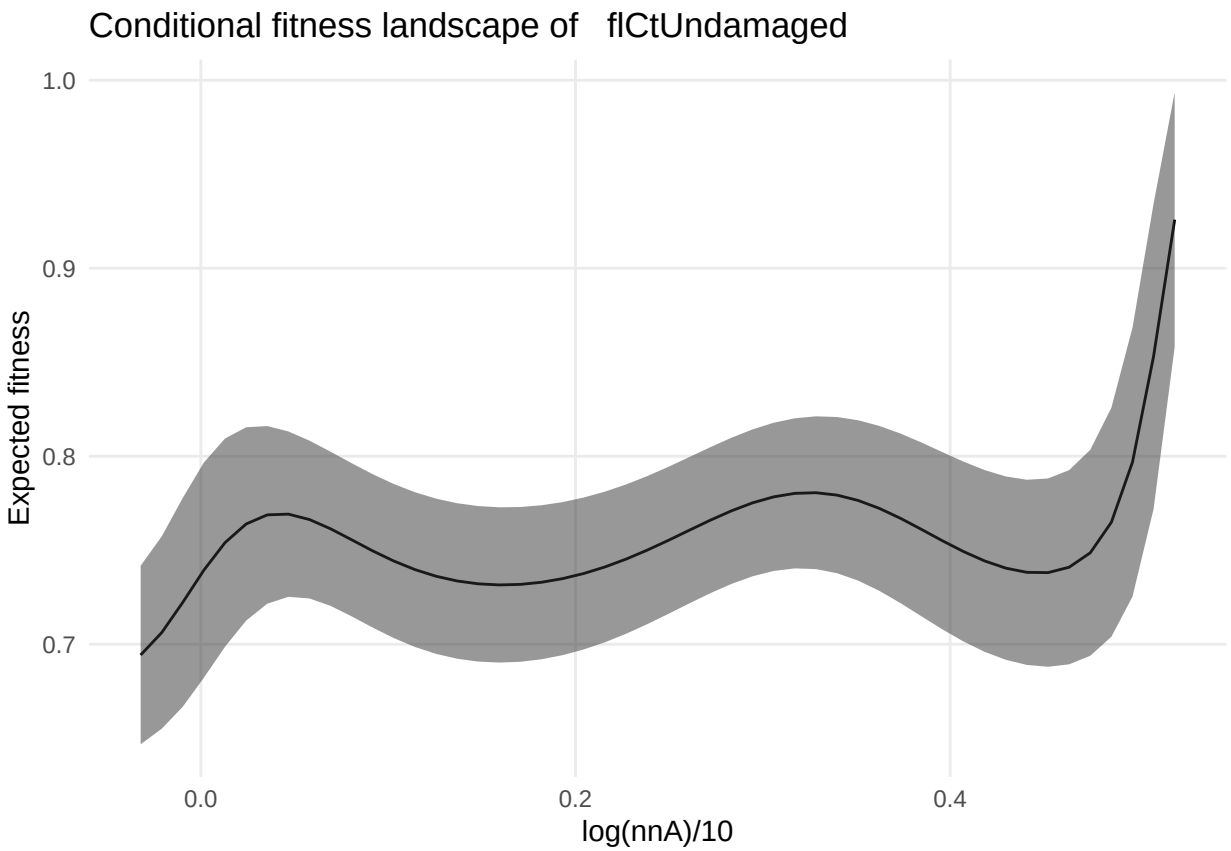

Conditional fitness landscape of ovuleCt
330.0
327.5
325.0
322.5
320.0
317.5
Expected fitness
0.0
0.2
0.4
log(nnA)/10

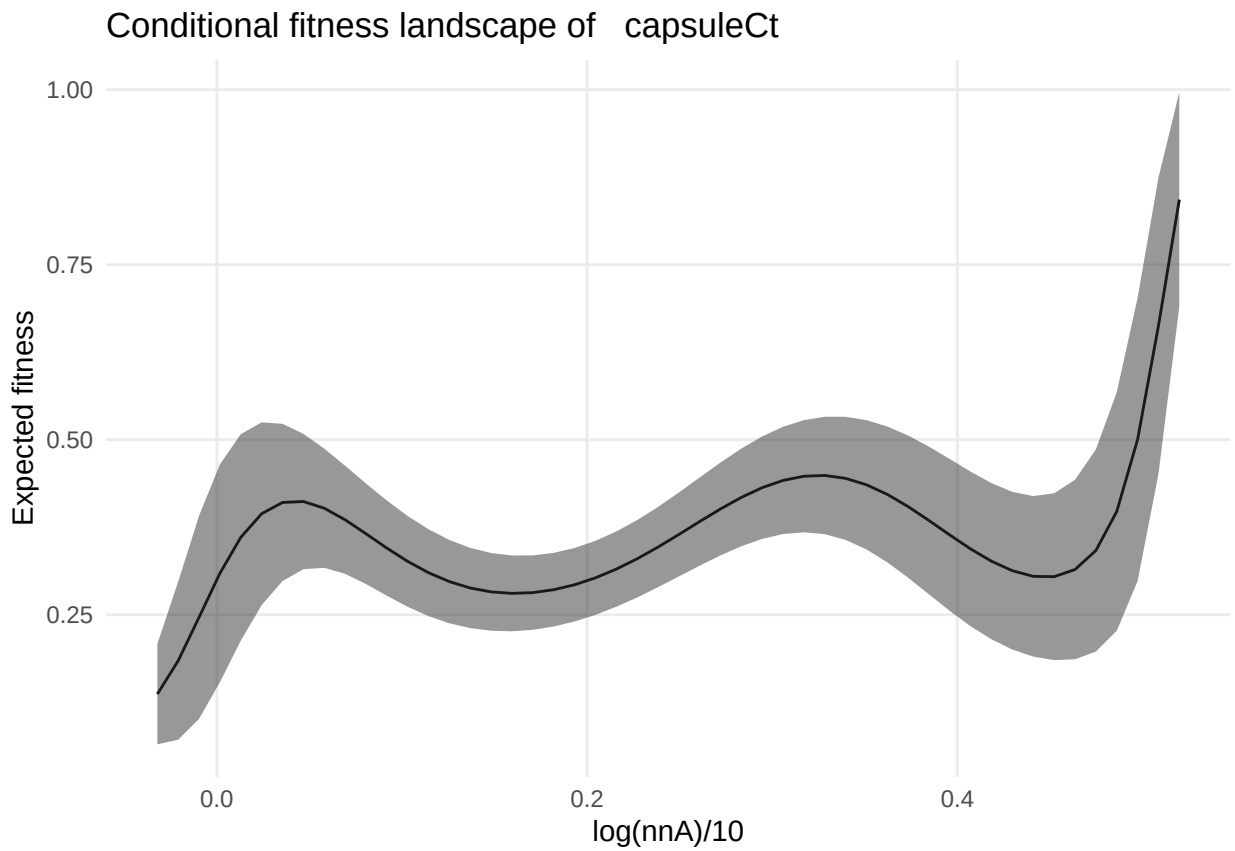


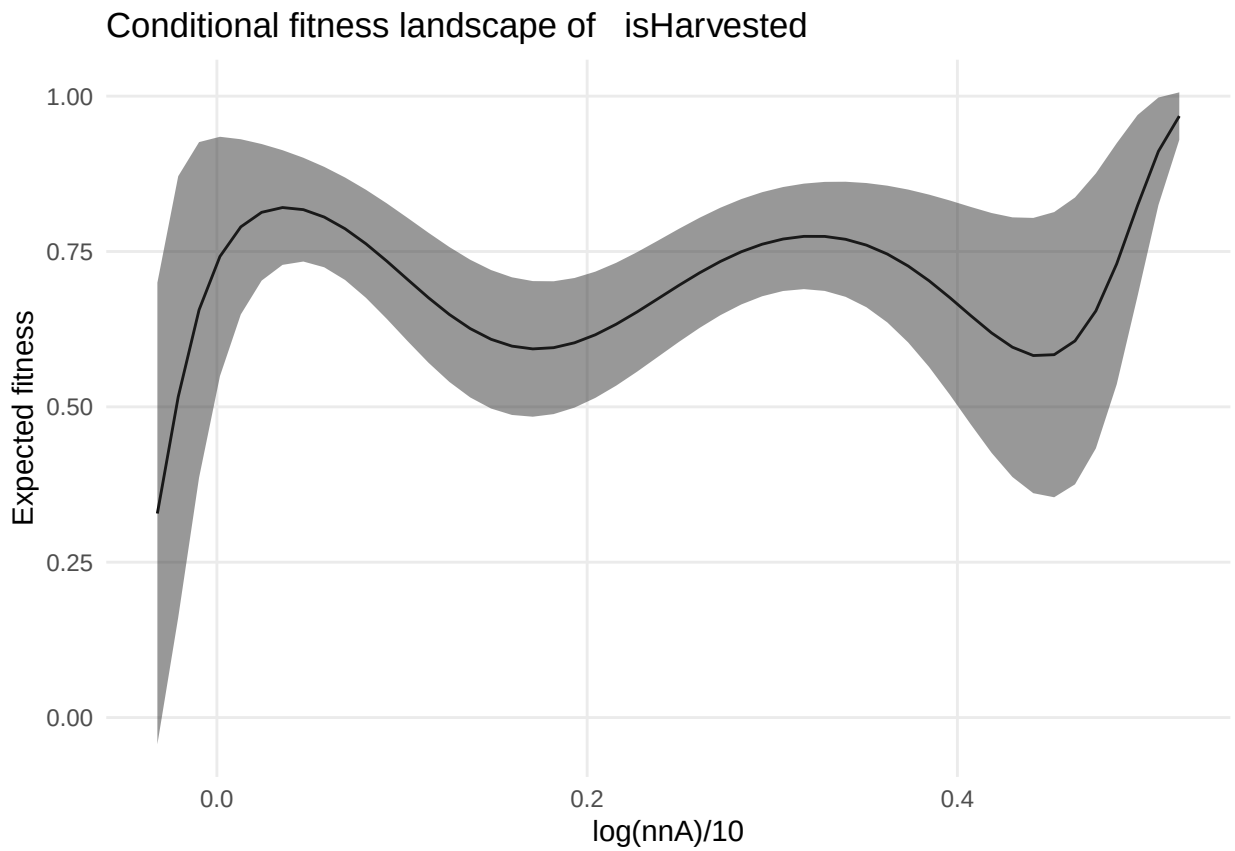

```
conditional_fitness_landscape(model_cubic1)
```

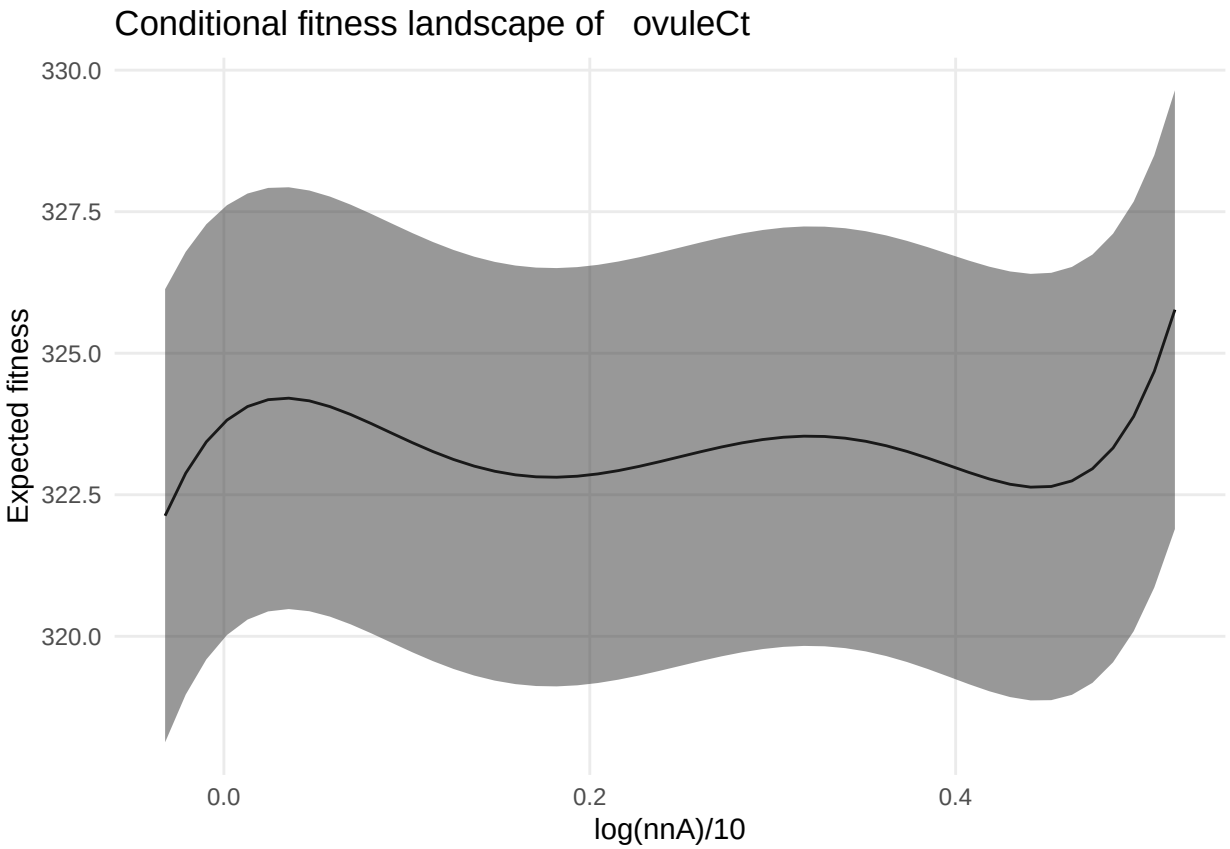

Conditional fitness landscape of flCt
Expected fitness
6
4
2
0.0
0.2
0.4
log(nnA)/10

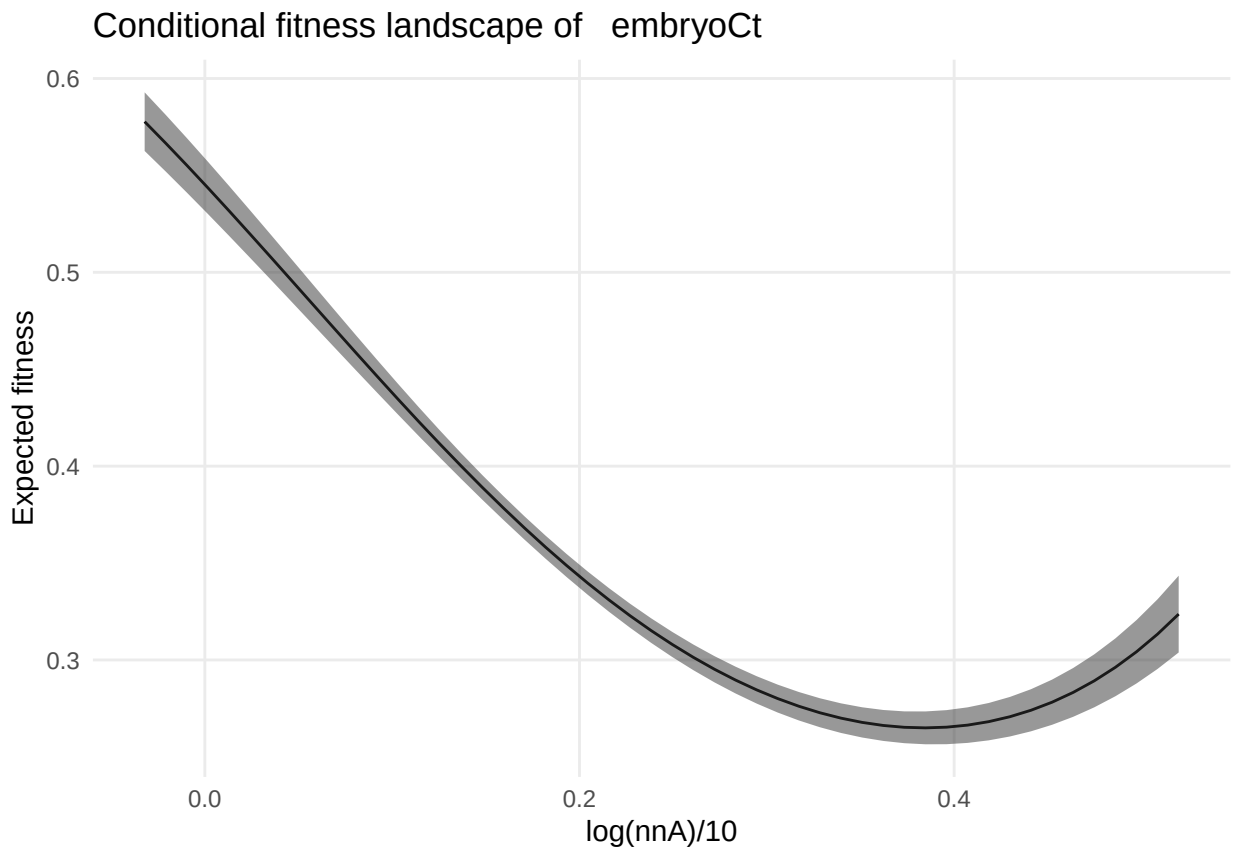

Conditional fitness landscape of flCtNotConsumed
Expected fitness
0.9
0.8
0.7
0.6
0.0
0.2
0.4
log(nnA)/10


Conditional fitness landscape of flCtUndamaged
1.0
0.9
0.8
0.7
Expected fitness
0.0
0.2
0.4
log(nnA)/10

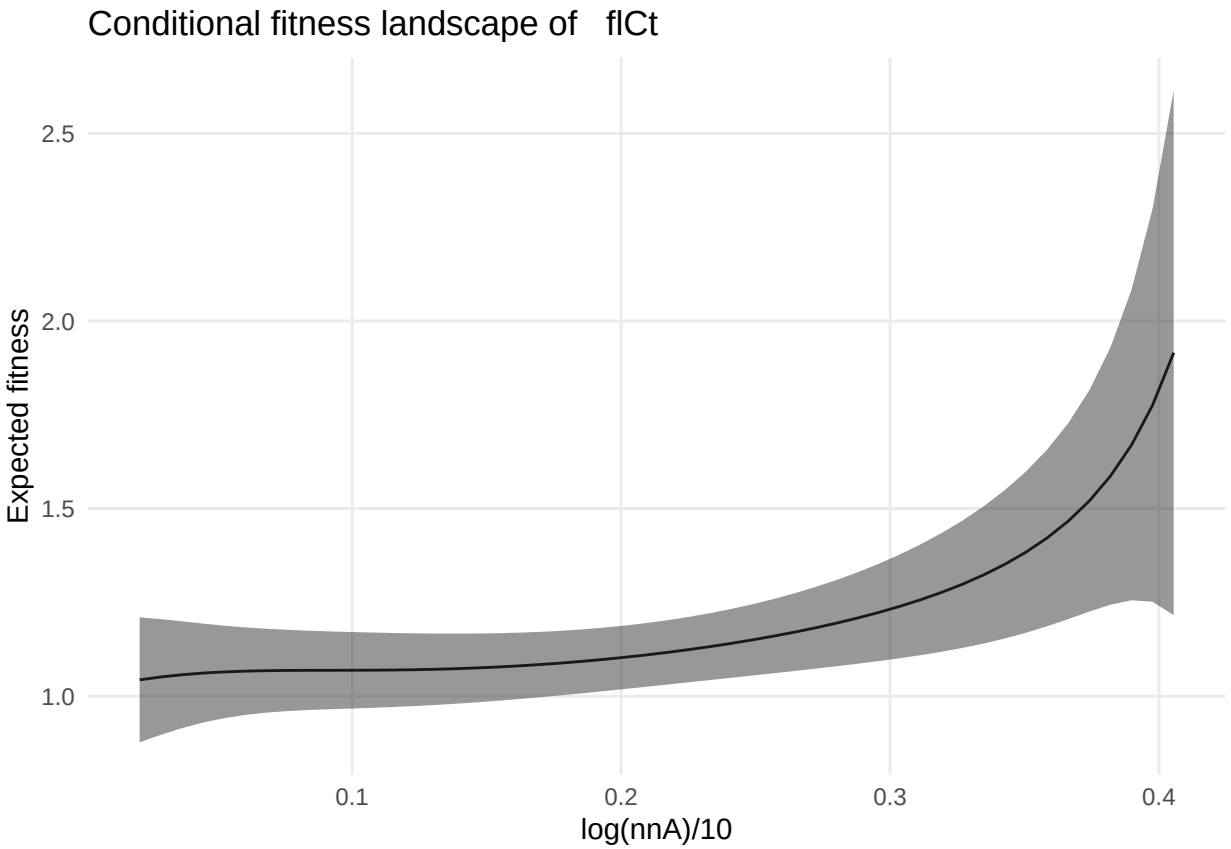

Conditional fitness landscape of capsuleCt
1.00
0.75
0.50
0.25
Expected fitness
0.0
0.2
0.4
log(nnA)/10

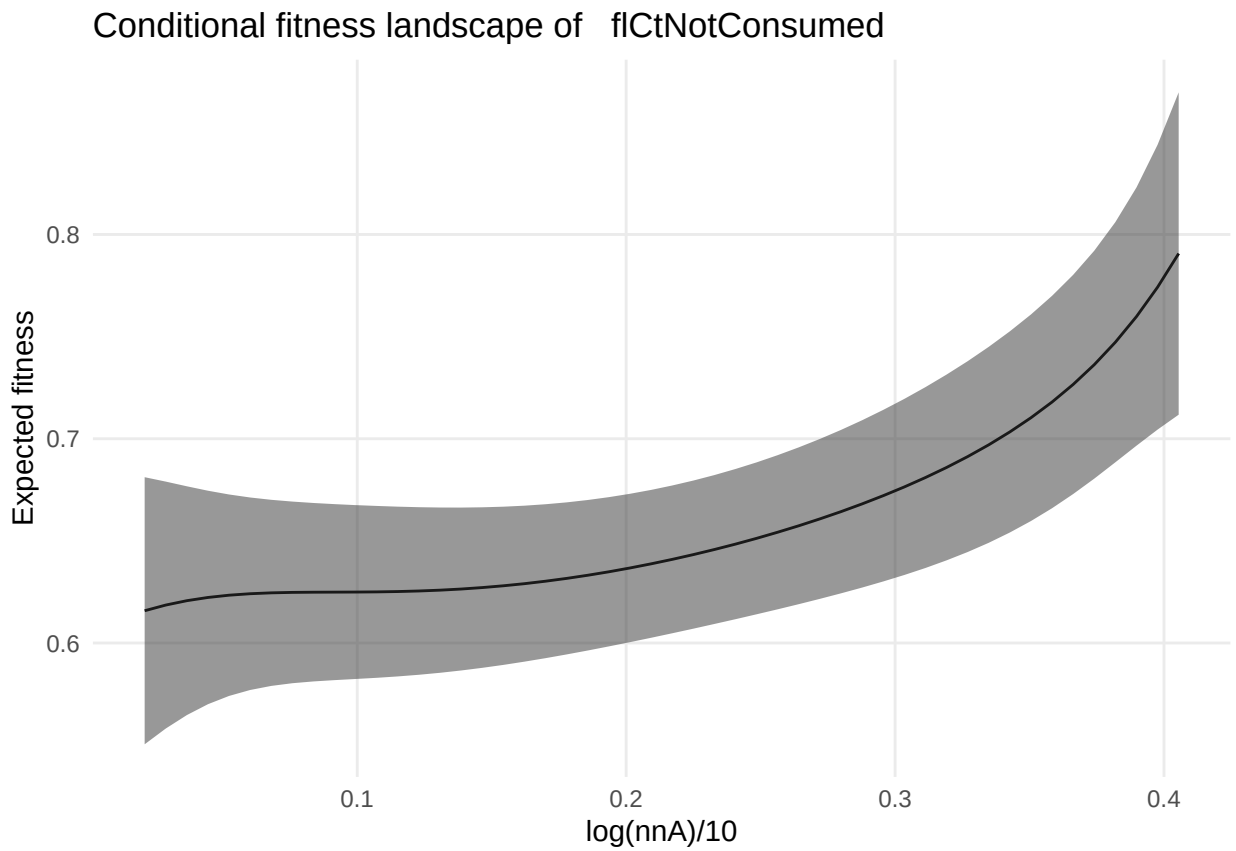

Conditional fitness landscape of isHarvested
1.00
0.75
0.50
0.25
0.00
Expected fitness
0.0
0.2
0.4
log(nnA)/10

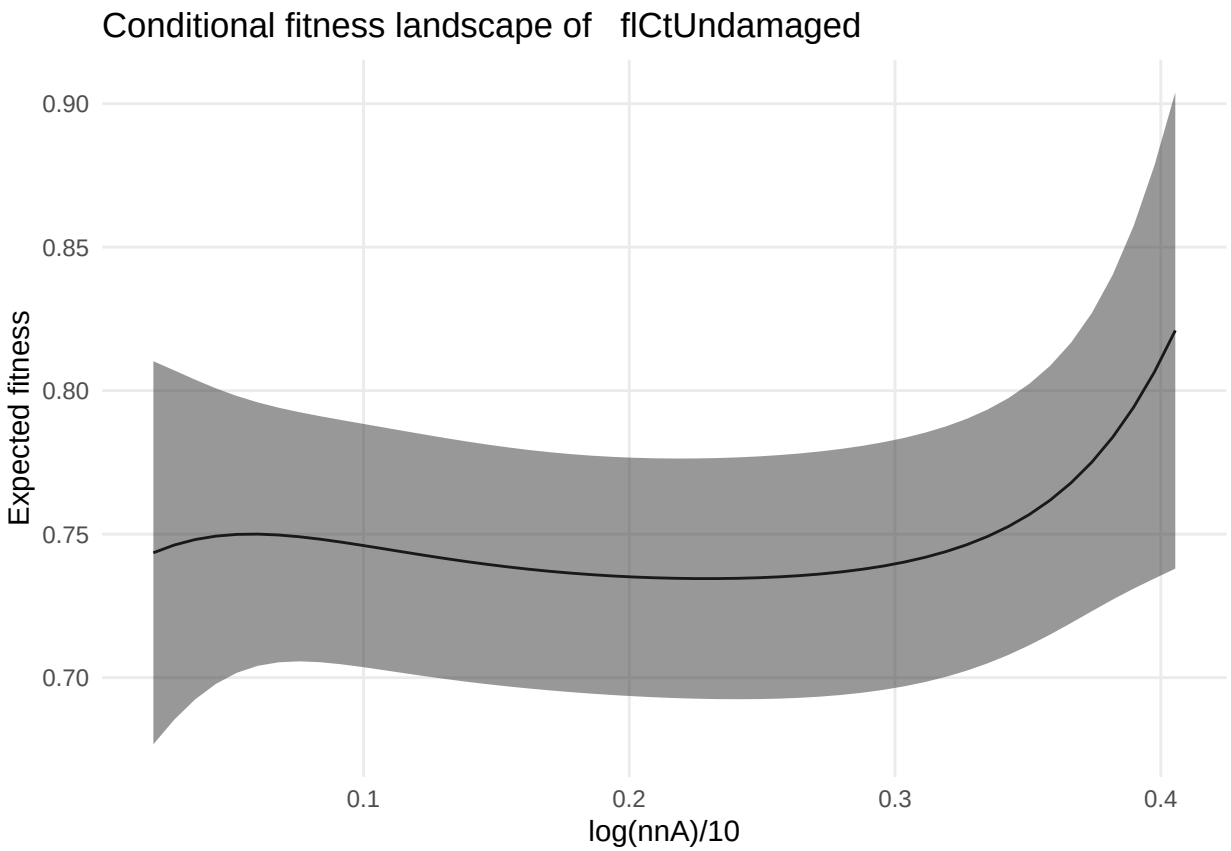

Conditional fitness landscape of ovuleCt
330.0
327.5
325.0
322.5
320.0
Expected fitness
0.0
0.2
0.4
log(nnA)/10

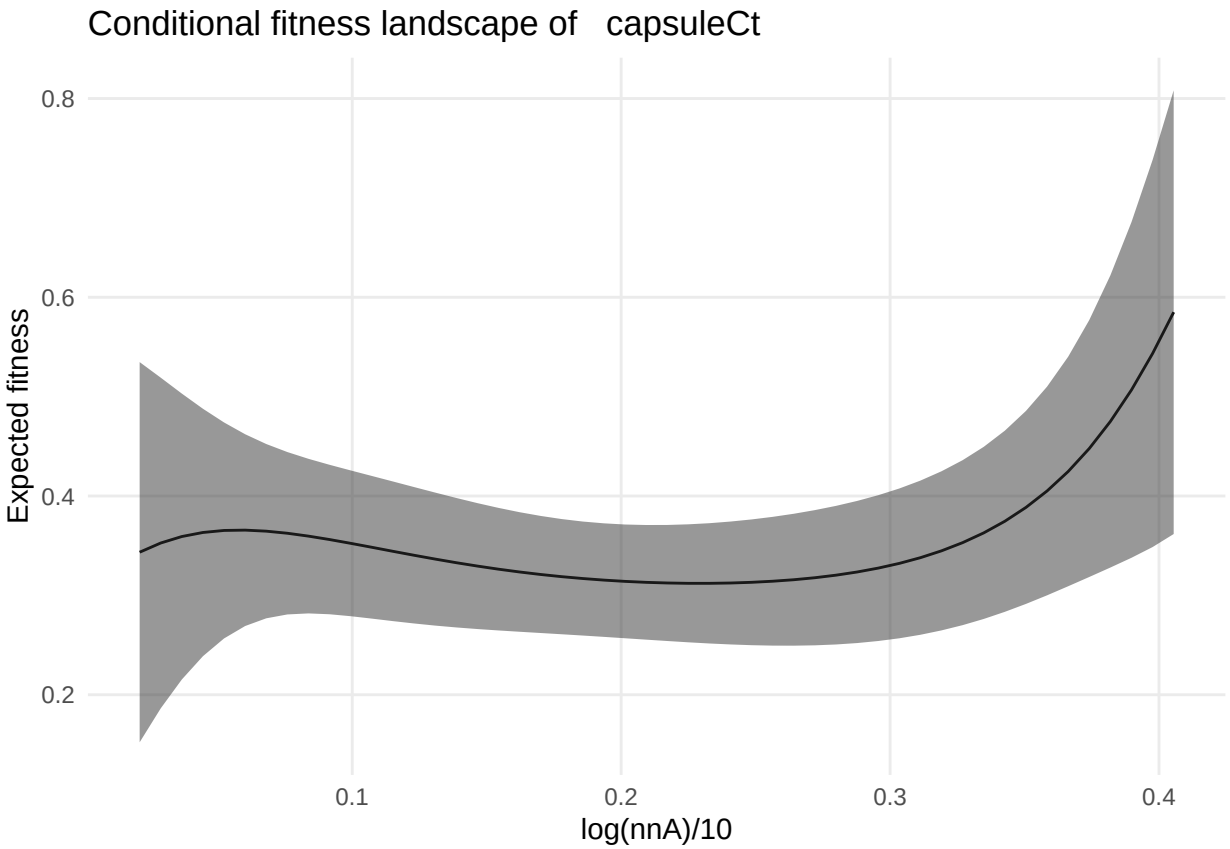


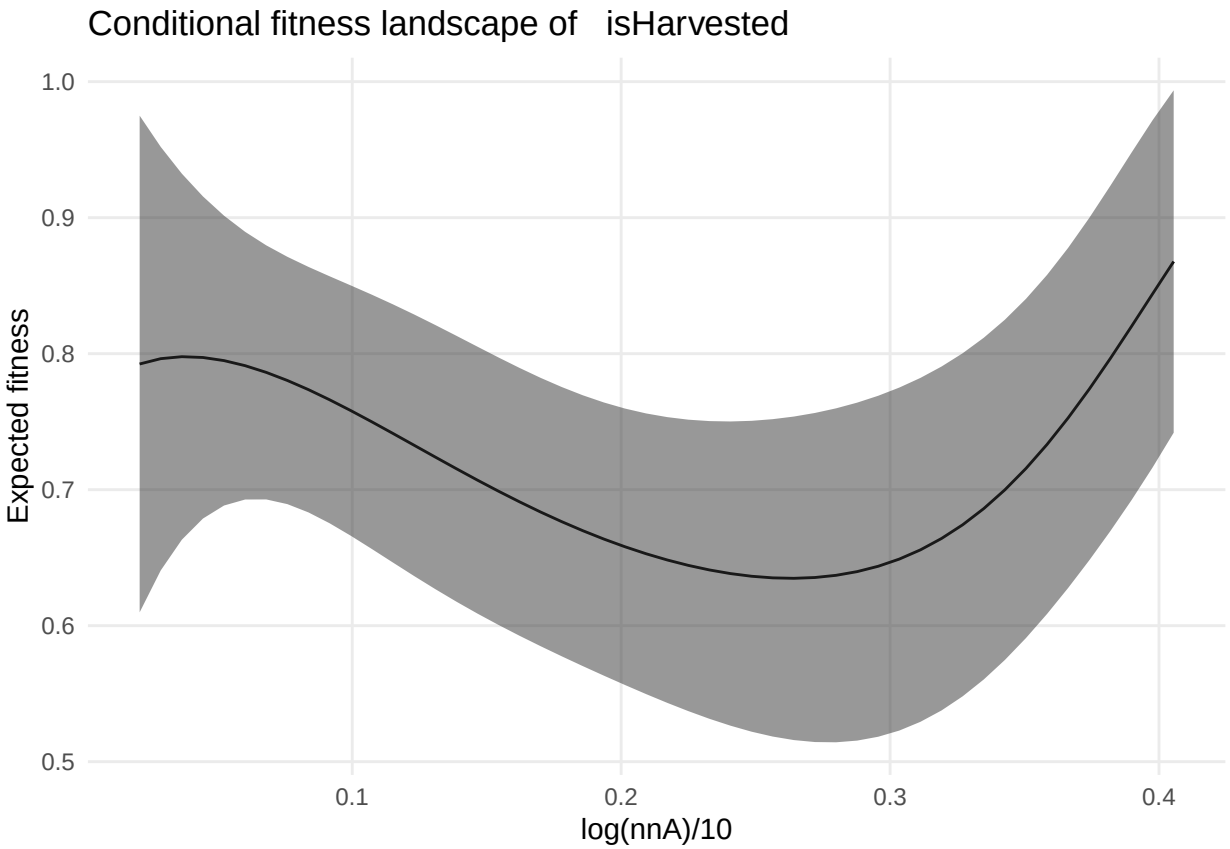

```
conditional_fitness_landscape(model_cubic2)
```

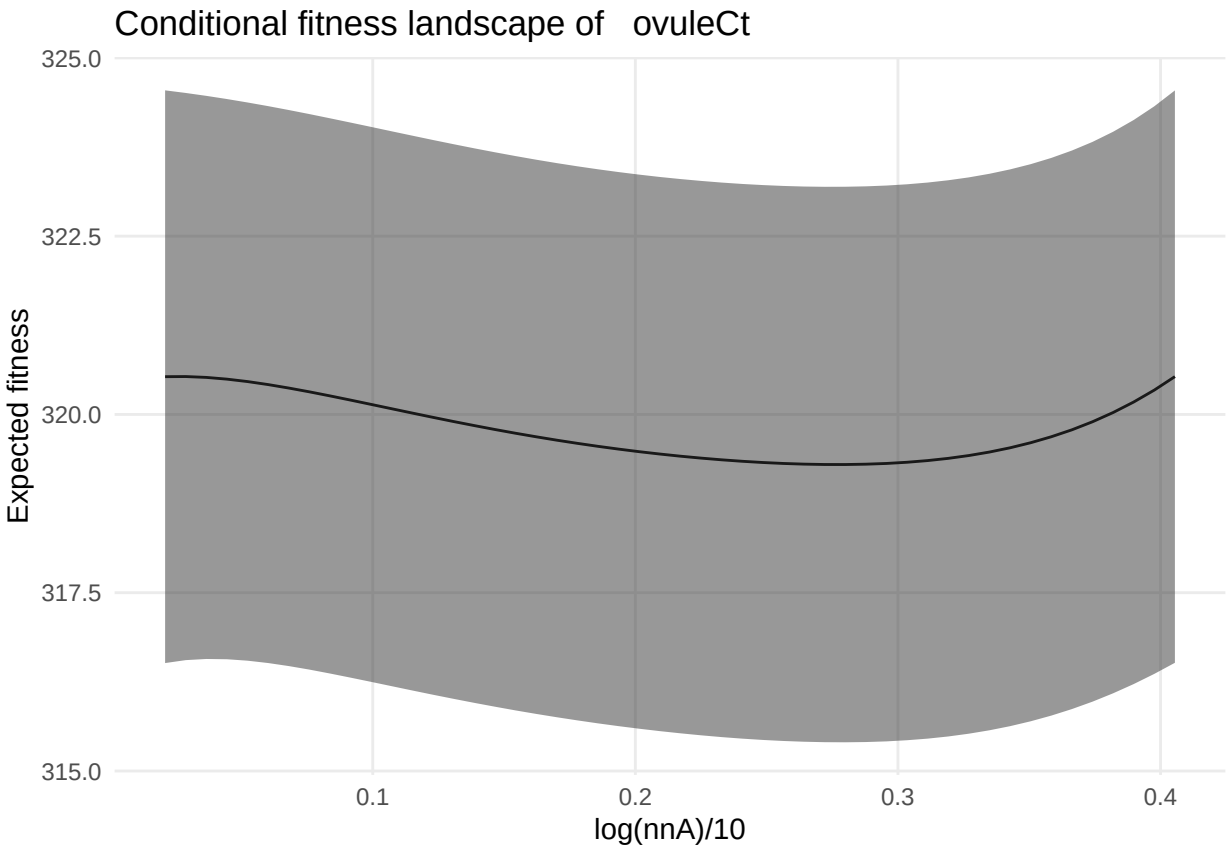

Conditional fitness landscape of flCt
Expected fitness
2.5
2.0
1.5
1.0
0.1
0.2
0.3
0.4
log(nnA)/10

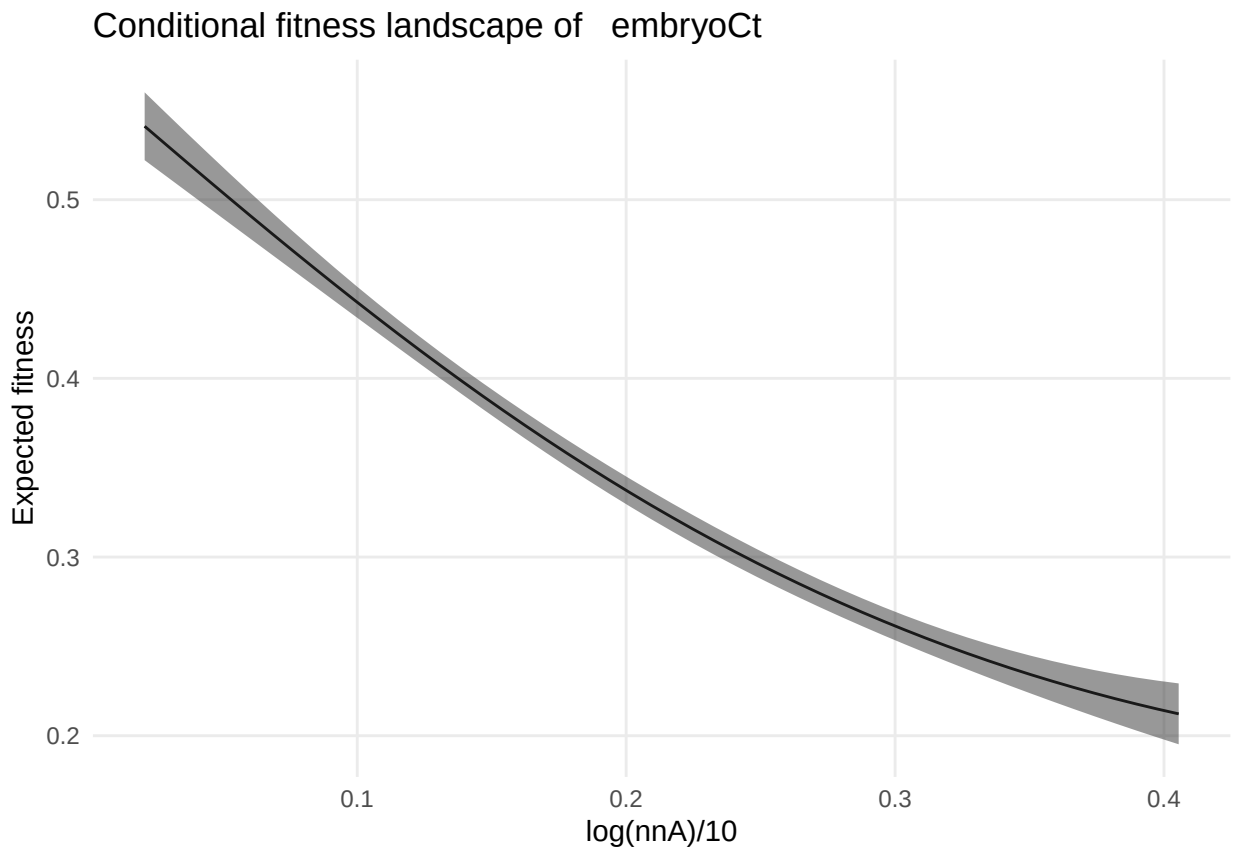

Conditional fitness landscape of flCtNotConsumed
Expected fitness
0.8
0.7
0.6
0.1
0.2
0.3
0.4
log(nnA)/10


Conditional fitness landscape of flCtUndamaged
Expected fitness
0.90
0.85
0.80
0.75
0.70
0.1
0.2
0.3
0.4
log(nnA)/10


Conditional fitness landscape of capsuleCt
Expected fitness
0.8
0.6
0.4
0.2
0.1
0.2
0.3
0.4
log(nnA)/10


Conditional fitness landscape of isHarvested
Expected fitness
1.0
0.9
0.8
0.7
0.6
0.5
0.1
0.2
0.3
0.4
log(nnA)/10


Conditional fitness landscape of ovuleCt
325.0
322.5
320.0
317.5
315.0
Expected fitness
0.1
0.2
0.3
0.4
log(nnA)/10



Zero-inflated model

As a comparison, a zero-inflated poisson model is fitted.

```
library(pscl)
```

```
## Warning: package 'pscl' was built under R version 4.2.3
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
mod_zeroinfl <- zeroinfl(embryoCt ~ nn5Dist_s, dist = 'poisson', data = test)
summary(mod_zeroinfl)
```

```
##
## Call:
## zeroinfl(formula = embryoCt ~ nn5Dist_s, data = test, dist = "poisson")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -0.4645 -0.3771 -0.3591 -0.3385 14.3296
##
## Count model coefficients (poisson with log link):
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.03239    0.01835 274.232   <2e-16 ***
## nn5Dist_s   -0.73043    0.08278  -8.824   <2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.2829     0.2627   8.689   <2e-16 ***
## nn5Dist_s    -1.4639     1.1897  -1.230    0.219
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 1
## Log-likelihood: -3792 on 4 Df
```

```
landscape_zeroinfl <- function(model, covariate = 'nn5Dist_s', lower = NULL, upper = NULL,
                               observation=FALSE, scale_back = FALSE) {
  # Make fake individuals
  nInd <- 50
  lwr <- if (is.null(lower)) min(unique(model$model[,covariate])) else lower
  upr <- if (is.null(upper)) max(unique(model$model[,covariate])) else upper
  cand.nnA <- seq(from = lwr, to = upr, length = nInd)
  cand <- as.data.frame(cand.nnA)
  colnames(cand) <- covariate
  cand$root <- 1
  blah <- data[1:nInd, colnames(data) %in% vars]
  cand <- cbind(cand, blah)
  cand$id <- data[1:nInd, 'id']

  #print(head(cand))


  # Get conditional mean value parameters
  pred <- predict(model, newdata=cand, type='response')
  #print(pred)

  pred_prob <- predict(model, newdata=cand, type='prob')
  #print(dim(pred_prob))

  xlabel = if (covariate == 'nn5Dist_s') 'log(nnA)/10' else 'log(nnB)/10'
  if (scale_back == TRUE) {
    cand[, as.character(covariate)] <- exp(10*cand[,as.character(covariate)])
    xlabel = if (covariate == 'nn5Dist_s') 'nnA' else 'nnB'
  }
  #cand_block <- cand_block[1:40,]
  plt <- ggplot(data = cand) + geom_line(mapping = aes(x = cand[,as.character(covariate)], y = pred))
  if (observation == TRUE) {
    obs <- test %>% filter((!!sym(covariate)) > lwr & (!!sym(covariate)) < upr)
    if (scale_back == TRUE) {
      plt <- plt + geom_point(data=obs, mapping = aes(x =exp(10*obs[,as.character(covariate)]),
                                                      y = obs[,'fecundity']))
    } else {
      plt <- plt + geom_point(data=obs, mapping = aes(x =obs[,as.character(covariate)], y = obs[,'fecun
    }
  }
```

```
  plt <- plt + labs(x=xlabel, y="Expected fitness", title = paste0("Fitness landscape    ", substr(forma
    #annotate(geom='text', x=c(0, 0.1, 0.3, 0.5), y=750, label=c('nnA', '2.718282', '20.085537', '148.4
    theme_minimal() + scale_x_continuous(minor_breaks = NULL) +scale_y_continuous(minor_breaks = NULL)

  print(plt)
  return(pred_prob)
}
```

```
zeroinfl_pred_prob <- landscape_zeroinfl(mod_zeroinfl)
```