

1 Appendix A. R-script file for null model analysis.

```
2 # This program tests for heterogeneity in linear trend lines
3 of species abundances.
4
5 #####
6 # GLOBAL VARIABLES
7
8 ObservedMatrix <- as.matrix(read.csv("Grossman Streamfish Data.csv", header=TRUE, row.names=1))
9 # input has species as rows, columns as times, with labels in the first row and column
10
11
12 Time <- c(11, 21, 34, 35, 45, 47, 57, 59, 71, 81, 82, 93, 117, 129, 141, 153)
13 # these are months since start of 1962 in Grossman streamfish datafile
14
15 NRep <- 1000
16 # this is the number of random replicates to use in the null model analysis
17
18 SimVar <- numeric(NRep)
19 # this is the vector of simulated variances in growth rates
20
21 ObsVar <- 0
22 # this is the single observed variance in growth rates for comparison with null
23 #####
24 # FUNCTION SourcePool
25 # This function takes an input matrix and a calculated number of missing species
26 # It returns a probability vector for creating null matrices
27 # For the missing species we assume their frequency = 0.5*rarest observed species
28
29 SourcePool <- function(FMatrix, FMissingSpecies) {
30
31     SourcePool <- rowSums(FMatrix)
32
33     SourcePool <- SourcePool/sum(SourcePool)
34     SourcePool <- c(SourcePool, rep(0.5*min(SourcePool), FMissingSpecies))
35
36 SourcePool <- SourcePool/sum(SourcePool)
37 }
38 #####
39 #
40 #
```

```

41 #
42 #
43 #####
44 # FUNCTION RandomSampler
45 # This function takes as input the SourcePool vector and the original data matrix
46 # It returns a single null matrix created by sampling individuals one at a time
47
48 RandomSampler <- function(FSourcePool, FMatrix) {
49
50   TimeProbs <- colSums(FMatrix)/sum(FMatrix)
51   PoissonN <- numeric(ncol(FMatrix))
52
53   for (j in 1:sum(FMatrix)) {
54     i <-sample(seq(1:length(PoissonN)),1,replace = TRUE,TimeProbs)
55     PoissonN[i] <- PoissonN[i] + 1
56   }
57
58
59   SampleMatrix <- matrix(0,length(FSourcePool),ncol(FMatrix))
60
61   for (j in 1:ncol(FMatrix)) {
62     for (n in 1:PoissonN[j]) {
63       i <- sample(seq(1:length(FSourcePool)),1,replace = TRUE,FSourcePool)
64       SampleMatrix[i,j] <- SampleMatrix[i,j] + 1
65     }
66   }
67   RandomSampler <- SampleMatrix
68 }
69 #####
70 #
71 #
72 #
73 #
74 #####
75 # FUNCTION GrowthVar
76 # This function takes as input a data matrix and a vector of times
77 # It fits a linear model to temporal trends for each species and
78 # calculates the variance among species in the measured slopes
79 # The calculation is only made for species that occurred at least once in the matrix
80 # This function can be modified to fit other kinds of temporal models
81
82 GrowthVar <- function(FMatrix, FTime) {
83

```

```

84     Growth <- NULL #creates empty vector of unspecified length
85
86     for (i in 1:nrow(FMatrix)) {
87
88         if (sum(FMatrix[i,]) > 0)
89     {
90         model <- lm(FMatrix[i,]~FTime) #fits model for species that are present
91         Growth[i] <- model$coefficients[2] #stores regression slope parameter
92     }
93 }
94 }
95 }
96 }
97
98 var(Growth, use = "complete") #species with 0 generate missing values that are ignored
99 }
100 #####
101 #
102 #
103 #
104 #
105 #####
106 # FUNCTION Chao2
107 # This function takes as input the observed matrix
108 # It calculated the bias corrected version of Chao2 (Equation 4 in EstimateS manual)
109 # Note that is this function is needed only once to get missing species for the simulation
110 # Results are rounded to the nearest whole integer
111
112 Chao2 <- function(FMatrix) {
113 Occurrences <- NULL
114 for (i in 1:nrow(FMatrix)) {
115 Occurrences[i] <- sum(FMatrix[i,] > 0)
116 }
117 }
118 Uniques <- sum(Occurrences == 1)
119 Duplicates <- sum(Occurrences == 2)
120 m <- ncol(FMatrix) # m = number of samples
121
122
123 Chao2 <- ((m - 1)/m)* ((Uniques*(Uniques - 1))/(2*(Duplicates + 1))
124 Chao2 <- round(Chao2)
125
126 }

```

```

127 #####
128 #
129 #
130 #
131 #
132 #####
133 # PROGRAM Basic Simulation Loop
134
135 MissingSpecies <- Chao2(ObservedMatrix)
136 SourcePoolVector <- SourcePool(ObservedMatrix,MissingSpecies)
137 ObsVar <-GrowthVar(ObservedMatrix,Time)
138
139 for (i in 1:NRep) {
140 RandomMatrix <- RandomSampler(SourcePoolVector,ObservedMatrix)
141 SimVar[i] <-GrowthVar(RandomMatrix,Time)
142 }
143 ConfidenceInterval <- quantile(SimVar,c(0.025,0.975))
144 ConfidenceInterval
145 ObsVar
146 SES <- (ObsVar - mean(SimVar))/sqrt(var(SimVar))
147 SES
148 sum(ObsVar > SimVar)/NRep
149 summary(SimVar)
150 #####

```