

1 This R script reads in the data, fits the models and visualises the data as presented in Ikoyi
2 et al. (2023), 'Plant diversity enhanced nematode-based soil quality indices and changed soil
3 nematode community structure in intensively-managed agricultural grasslands' in European
4 Journal of Soil Biology, 118, [103542](https://doi.org/10.1016/j.eurosoil.2023.103542).

5 The response variables comprise abundances of taxonomic groups of soil nematodes,
6 nematode-based soil quality indices, as well as measured properties of the vegetation and
7 soil from the plots on which nematodes were sampled (see associated ReadMe file).

8 The general approach is to analyse the multiple nematode indices using the DImodels
9 package. The DImodels package allows selection of the best-fitting of several alternative
10 models (STR, ID, AV, FG, and FULL) (see Moral et al. 2023), based on different
11 assumptions about the net effect of plant interspecific interactions on the nematode
12 community.

13 For visualization, we use the ternary diagrams to display the results of the best DI model
14 predictions for each of the responses. Please see lines 671-776 of this script.

15

16 Reference

17 Moral, R.A., Vishwakarma, R., Connolly, J., Byrne, L., Hurley, C., Finn, J.A. and Brophy, C.,
18 2022. Going beyond richness: Modelling the BEF relationship using species identity,
19 evenness, richness and species interactions via the DImodels R package. Methods in
20 Ecology and Evolution. <https://doi.org/10.1111/2041-210X.14158>

21

22 #####

23 R Script

24 #####

25 #load required packages

26 library(readxl)

27 library(ggplot2)

28 library(DImodels)

29 library(tidyverse)

30 library(Ternary)

31 getwd()

32 setwd("[insert location of working directory]")

33 #import data

34 NemInd <- read.csv("[Insert location of data file]")

35 #Read in the data from the location stored on the computer

36 View(NemInd)

37 NemInd\$OmvPred <- NemInd\$Omv+NemInd\$Pred

```
38
39 #analysis of Maturity Index, variable 'MI'
40
41 #2. Generate a scatterplot of the response vs richness
42 NemInd$plantSR<-as.factor(NemInd$plantSR)
43 plot(NemInd$plantSR, NemInd$MI, ylab = "Nematode Maturity Index",
44      xlab = "richness")
45 #3. Fit the models
46 #STR
47 m99<- DI (y = "MI", prop = 7:12, DImodel = "STR", data = NemInd)
48 summary(m99)
49
50 #ID
51
52 m100<- DI (y = "MI", prop = 7:12, DImodel = "ID", data = NemInd)
53 summary(m100)
54
55 #AV
56 m101<- DI (y = "MI", prop = 7:12, DImodel = "AV", data = NemInd)
57 summary(m101)
58
59 #FG
60 m102<- DI (y = "MI", prop = 7:12, DImodel = "FG",
61           FG = c("FG1", "FG1", "FG2", "FG2", "FG3", "FG3"), data = NemInd)
62 summary(m102)
63
64 #FULL
65 m103<- DI (y = "MI", prop = 7:12, DImodel = "FULL", data = NemInd)
66 summary(m103)
67
68 #MODEL EVALUATION
69 anova(m99, m100, m101, m102, m103, test="F")
```

```

70
71 auto100 <- autoDI(y = "MI", prop = 7:12,
72                 FG = c("FG1", "FG1", "FG2", "FG2", "FG3", "FG3"), data = NemInd, selection =
73 "Ftest")
74 summary(auto100)
75 # Function to create ternary diagram
76 #get_ternary <- function(model, axis_text_size = 6, axis_label_size = 6, height = 100, width
77 = 87)
78 #get_ternary(model = m101)
79
80 predictions <- predict(m102)
81 predict(m102)
82
83 ##analysis of bacterial-feeding nematodes, variable 'PF'BF
84 #2. Generate a scatterplot of the response vs richness
85 plot(NemInd$plantSR, NemInd$BF, ylab = "Abundance of Bacterial Feeders",
86      xlab = "richness")
87
88 #3. Fit the models
89
90 #ID
91
92 m24<- DI (y = "BF", prop = 7:12, DImodel = "ID", data = NemInd)
93 summary(m24)
94
95 #AV
96 m25<- DI (y = "BF", prop = 7:12, DImodel = "AV", data = NemInd)
97 summary(m25)
98
99 #FG
100 m26<- DI (y = "BF", prop = 7:12, DImodel = "FG",
101          FG = c("FG1", "FG1", "FG2", "FG2", "FG3", "FG3"), data = NemInd)
102 summary(m26)

```

```

103
104 #FULL
105 m27<- DI (y = "BF", prop = 7:12, DImodel = "FULL", data = NemInd)
106 summary(m27)
107 #MODEL EVALUATION
108 anova(m24, m25, m26, m27, test="F")
109 #AV
110 m28<- DI (y = "BF", prop = 7:12, DImodel = "STR", data = NemInd)
111 summary(m28)
112
113 conBF <- contrasts_DI(object = m28,
114                       contrast = list('p1vall' = c(1)))
115 summary(conBF)
116
117 auto21 <- autoDI(y = "BF", prop = 7:12,
118                 FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, selection =
119 "Ftest")
120 summary(auto21)
121 #get_ternary(model = auto21)
122
123
124 #analysis of plant-feeding nematodes, variable 'PF'
125 #2. Generate a scatterplot of the response vs richness
126 plot(NemInd$plantSR, NemInd$PF, ylab = "Abundance of Plant Feeders",
127       xlab = "richness")
128 hist(NemInd$PF)
129
130 NemInd$logPF <- log(NemInd$PF + 1)
131 hist(NemInd$logPF)
132 #3. Fit the models
133
134 #ID
135

```

```

136 m104<- DI (y = "logPF", prop = 7:12, DImodel = "ID", data = NemInd)
137 summary(m104)
138
139 #AV
140 m105<- DI (y = "logPF", prop = 7:12, DImodel = "AV", data = NemInd)
141 summary(m105)
142
143 #FG
144 m106<- DI (y = "logPF", prop = 7:12, DImodel = "FG",
145           FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd)
146 summary(m106)
147
148 #FULL
149 m107<- DI (y = "logPF", prop = 7:12, DImodel = "FULL", data = NemInd)
150 summary(m107)
151 #STR
152 m205<- DI (y = "logPF", prop = 7:12, DImodel = "STR", data = NemInd)
153 #MODEL EVALUATION
154 anova(m205, m104, m105, m106, m107, test="F")
155 summary(m205)
156
157 #get_ternary(model = m104)
158 con <- contrasts_DI(object = m205,
159                   contrast = list('p1vall' = c(1)))
160 summary(con)
161
162 auto101 <- autoDI(y = "logPF", prop = 7:12,
163                 FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, selection =
164 "Ftest")
165 summary(auto101)
166 #get_ternary(model = auto101)
167
168 ##analysis of c-p35 index, variable 'CP35'

```

```

169 #2. Generate a scatterplot of the response vs richness
170 plot(NemInd$plantSR, NemInd$CP35, ylab = "Abundance of sensitive taxa",
171       xlab = "richness")
172
173 #3. Fit the models
174
175 #ID
176
177 m108<- DI (y = "CP35", prop = 7:12, DImodel = "ID", data = NemInd)
178 summary(m108)
179
180 #AV
181 m109<- DI (y = "CP35", prop = 7:12, DImodel = "AV", data = NemInd)
182 summary(m109)
183
184 #FG
185 m110<- DI (y = "CP35", prop = 7:12, DImodel = "FG",
186           FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd)
187 summary(m110)
188
189 #FULL
190 m111<- DI (y = "CP35", prop = 7:12, DImodel = "FULL", data = NemInd)
191 summary(m111)
192 #MODEL EVALUATION
193 anova(m108, m109, m110, m111, test="F")
194
195 auto102 <- autoDI(y = "CP35", prop = 7:12,
196                 FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, selection =
197 "Ftest")
198 m109_redo <- DI (y = "CP35", prop = 7:12, DImodel = "AV",
199                 FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd)
200 summary(m109_redo)
201 m109_theta <- update_DI(object = m109_redo, estimate_theta = TRUE)

```

```
202 summary(m109_theta)
203 summary(auto102)
204 get_ternary(model = m109_theta)
205
206 plot(NemInd$plantSR, NemInd$CP35, ylab = "Abundance of c-p 3-5 nematodes (sensitive
207 taxa)",
208       xlab = "richness")
209 boxplot(CP35~plantSR, data = NemInd)
210
211 #logCP35
212 NemInd$logCP35 <- log(NemInd$CP35 + 1)
213 #2. Generate a scatterplot of the response vs richness
214 plot(NemInd$plantSR, NemInd$logCP35, ylab = "log Abundance of sensitive taxa",
215       xlab = "richness")
216
217 #3. Fit the models
218
219 #ID
220
221 m208<- DI (y = "logCP35", prop = 7:12, DImodel = "ID", data = NemInd)
222 summary(m208)
223
224 #AV
225 m209<- DI (y = "logCP35", prop = 7:12, DImodel = "AV", data = NemInd)
226 summary(m209)
227
228 #FG
229 m210<- DI (y = "logCP35", prop = 7:12, DImodel = "FG",
230           FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd)
231
232 summary(m210)
233
234 #FULL
```

```

235 m211<- DI (y = "logCP35", prop = 7:12, DImodel = "FULL", data = NemInd)
236 summary(m211)
237 #MODEL EVALUATION
238 anova(m208, m209, m210, m211, test="F")
239
240 auto202 <- autoDI(y = "logCP35", prop = 7:12,
241                 FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, selection =
242 "Ftest")
243 m209_redo <- DI (y = "logCP35", prop = 7:12, DImodel = "AV",
244                 FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd)
245 summary(m209_redo)
246
247 m209_theta <- update_DI(object = m209_redo, estimate_theta = TRUE)
248 summary(m209_theta)
249 summary(auto202)
250 #get_ternary(model = m209)
251
252 plot(NemInd$plantSR, NemInd$CP35, ylab = "Abundance of sensitive taxa",
253      xlab = "richness")
254 hist(NemInd$logCP35)
255
256 #analysis of nematode enrichment index, variable 'EI'
257 #2. Generate a scatterplot of the response vs richness
258 plot(NemInd$plantSR, NemInd$EI, ylab = "Nematode enrichment index",
259      xlab = "richness")
260
261 #3. Fit the models
262
263 #ID
264
265 m112<- DI (y = "EI", prop = 7:12, DImodel = "ID", data = NemInd)
266 summary(m112)
267

```



```

268 #AV
269 m113<- DI (y = "EI", prop = 7:12, DImodel = "AV", data = NemInd)
270 summary(m113)
271
272 #FG
273 m114<- DI (y = "EI", prop = 7:12, DImodel = "FG",
274           FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd)
275 summary(m114)
276
277 #FULL
278 m115<- DI (y = "EI", prop = 7:12, DImodel = "FULL", data = NemInd)
279 summary(m115)
280 #MODEL EVALUATION
281 anova(m112, m113, m114, m115, test="F")
282
283 auto103 <- autoDI(y = "EI", prop = 7:12,
284                 FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, selection =
285 "Ftest")
286 m119_redo <- DI (y = "EI", prop = 7:12, DImodel = "FULL",
287                FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, )
288 summary(m109_redo)
289 m109_theta <- update_DI(object = m109_redo, estimate_theta = TRUE)
290 summary(m109_theta)
291 summary(auto102)
292 #get_ternary(model = m113)
293
294
295 #analysis of nematode structure index, variable 'SI'
296 #2. Generate a scatterplot of the response vs richness
297 plot(NemInd$plantSR, NemInd$SI, ylab = "Nematode structure index",
298      xlab = "richness")
299
300 #3. Fit the models

```

```
301
302 #ID
303
304 m116<- DI (y = "SI", prop = 7:12, DImodel = "ID", data = NemInd, estimate_theta = T)
305 summary(m116)
306
307 #AV
308 m117<- DI (y = "SI", prop = 7:12, DImodel = "AV", data = NemInd)
309 summary(m117)
310
311 #FG
312 m118<- DI (y = "SI", prop = 7:12, DImodel = "FG",
313           FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, estimate_theta = T)
314 summary(m118)
315
316 #FULL
317 m119<- DI (y = "SI", prop = 7:12, DImodel = "FULL", data = NemInd)
318 summary(m119)
319 #MODEL EVALUATION
320 anova(m116, m117, m118, m119, test="F")
321
322 auto104 <- autoDI(y = "SI", prop = 7:12,
323                 FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, selection =
324 "Ftest")
325 m119_redo <- DI (y = "SI", prop = 7:12, DImodel = "FULL",
326                 FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, estimate_theta =
327 T)
328 summary(m119_redo)
329
330 m117_redo <- DI (y = "SI", prop = 7:12, DImodel = "AV",
331                 FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, estimate_theta =
332 T)
333 summary(m117_redo)
```

```
334 anova(m119_redo,m117_redo)
335 m109_theta <- update_DI(object = m109_redo, estimate_theta = TRUE)
336 summary(m109_theta)
337 summary(auto102)
338 #get_ternary(model = m117)
339
340
341
342 #analysis of nematode channel index, variable 'CI'
343 #2. Generate a scatterplot of the response vs richness
344 plot(NemInd$plantSR, NemInd$CI, ylab = "Nematode Channel index",
345       xlab = "richness")
346 hist(NemInd$CI)
347 NemInd$logCI <- log(NemInd$CI + 1)
348 hist(NemInd$logCI)
349 NemInd$propCI <- NemInd$CI/100
350 hist(NemInd$propCI)
351 NemInd$asinCI <- asin(sqrt(NemInd$propCI))
352 hist(NemInd$asinCI)
353 #3. Fit the models
354
355 #ID
356
357 m120<- DI (y = "logCI", prop = 7:12, DImodel = "ID", data = NemInd)
358 summary(m120)
359
360 #AV
361 m121<- DI (y = "logCI", prop = 7:12, DImodel = "AV", data = NemInd)
362 summary(m121)
363
364 #FG
365 m122<- DI (y = "logCI", prop = 7:12, DImodel = "FG",
```

```
366         FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd)
367 summary(m122)
368
369 #FULL
370 m123<- DI (y = "logCI", prop = 7:12, DImodel = "FULL", data = NemInd)
371 summary(m123)
372 #STR
373 m124<- DI (y = "logCI", prop = 7:12, DImodel = "STR", data = NemInd)
374 summary(m124)
375 #MODEL EVALUATION
376 anova(m124,m120, m121, m122, m123, test="F")
377
378 auto105 <- autoDI(y = "asinCI", prop = 7:12,
379                 FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, selection =
380 "Ftest")
381
382
383
384 #More on predictions
385 library(lattice)
386 library(grid)
387 #install.packages("daewr")
388 library(daewr)
389 library(gdata)
390 #install.packages("gdata")
391 library(mixexp)
392 #install.packages("mixexp")
393 library(ggplot2)
394
395
396 ##### Plotting #####
397
398 #Set color palette (not necessary, but allows to set the green as highest value)
```

```

399 revterrain.colors <- function (n, alpha = 1)
400 {
401   if ((n <- as.integer(n[1L])) > 0) {
402     k <- n%%2
403     h <- rev(c(4/12, 2/12, 0/12))
404     s <- rev(c(1, 1, 0))
405     v <- rev(c(0.65, 0.9, 0.95))
406     c(hsv(h = seq.int(h[1L], h[2L], length.out = k), s = seq.int(s[1L],
407                               s[2L], length.out = k), v = seq.int(v[1L], v[2L],
408                               length.out = k), alpha =
409     alpha), hsv(h = seq.int(h[2L],
410                               h[3L], length.out = n - k + 1)[-1L], s = seq.int(s[2L],
411                               s[3L], length.out = n - k + 1)[-1L], v = seq.int(v[2L],
412                               v[3L], length.out = n - k + 1)[-1L], alpha = alpha))
413   }
414   else character()
415 }
416
417 #Contrasts
418
419
420
421
422
423 con1 <- contrasts_DI(object = m113,
424                     contrast = list('p1vp2 Mono' = c(1, -1, 0, 0, 0, 0, 0),
425                                     'p3vp4 Mono' = c(0, 0, 1, -1, 0, 0, 0),
426                                     'p5vp6 Mono' = c(0, 0, 0, 0, 1, -1, 0),
427                                     'p1vp3 Mono' = c(1, 0, -1, 0, 0, 0, 0),
428                                     'p1vp4 Mono' = c(1, 0, 0, -1, 0, 0, 0),
429                                     'p1vp5 Mono' = c(1, 0, 0, 0, -1, 0, 0),
430                                     'p1vp6 Mono' = c(1, 0, 0, 0, 0, -1, 0),
431                                     'p2vp3 Mono' = c(0, 1, -1, 0, 0, 0, 0),
432                                     'p2vp4 Mono' = c(0, 1, 0, 0, -1, 0, 0),

```

```

433         'p2vp5 Mono' = c(0, 1, 0, 0, -1, 0, 0),
434         'p2vp6 Mono' = c(0, 1, 0, 0, 0, -1, 0),
435         'p3vp5 Mono' = c(0, 0, 1, 0, -1, 0, 0),
436         'p3vp6 Mono' = c(0, 0, 1, 0, 0, -1, 0),
437         'p4vp5 Mono' = c(0, 0, 0, 1, -1, 0, 0),
438         'p4vp6 Mono' = c(0, 0, 0, 0, 0, -1, 0)))
439 summary(con1)
440
441 con11 <- contrasts_DI(object = m113,
442         contrast = list('GvL FG' = c(0.5, 0.5, -0.5, -0.5, 0, 0, 0),
443         'GvH FG' = c(0.5, 0.5, 0, 0, -0.5, -0.5, 0),
444         'LvH FG' = c(0, 0, 0.5, 0.5, -0.5, -0.5, 0)))
445 summary(con11)
446 conMI <- contrasts_DI(object = m113,
447         contrast = list('p1vmss' = c(0.833333,-0.16667,-0.16667,-0.16667,-0.16667,-
448 0.16667, -0.41667),
449         'p2vmss' = c(-0.16667,0.833333,-0.16667,-0.16667,-0.16667,-
450 0.16667, -0.41667),
451         'p3vmss' = c(-0.16667,-0.16667,0.833333,-0.16667,-0.16667,-
452 0.16667, -0.41667),
453         'p4vmss' = c(-0.16667,-0.16667,-0.16667, 0.833333,-0.16667,-
454 0.16667, -0.41667),
455         'p5vmss' = c(-0.16667,-0.16667,-0.16667,-0.16667,0.833333,-
456 0.16667, -0.41667),
457         'p6vmss' = c(-0.16667,-0.16667,-0.16667,-0.16667,-
458 0.16667,0.833333, -0.41667)))
459 summary(conMI)
460 predict(m101)
461
462
463
464 #####
465 hist(NemInd$CI)
466 NemInd$logCI<-log(NemInd$CI + 1)
467 hist(NemInd$logCI)

```

```
468
469 #3. Fit the models
470
471 #ID
472 #STR
473 m1<- DI (y = "logCI", prop = 7:12, DImodel = "STR", data = NemInd)
474 summary(m1)
475
476 #ID
477 m2<- DI (y = "logCI", prop = 7:12, DImodel = "ID", data = NemInd)
478 summary(m2)
479
480 m21<- DI (y = "CI", prop = 7:12, DImodel = "ID", data = NemInd)
481 summary(m21)
482 predict(m21)
483 #AV
484 m3<- DI (y = "logCI", prop = 7:12, DImodel = "AV", data = NemInd)
485 summary(m3)
486
487 #FG
488 m4<- DI (y = "logCI", prop = 7:12, DImodel = "FG",
489         FG = c("FG1", "FG1", "FG2", "FG2", "FG2", "FG2"), data = NemInd)
490 summary(m4)
491
492 #FULL
493 m5<- DI (y = "logCI", prop = 7:12, DImodel = "FULL", data = NemInd)
494 summary(m5)
495 #MODEL EVALUATION
496 anova(m1, m2, m3, m4, m5, test="F")
497
498 autoCI <- autoDI(y = "logCI", prop = 7:12,
499                FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, selection =
500 "Ftest")
```

```

501
502 #get_ternary(m2)
503 #contrasts###
504 con1CI <- contrasts_DI(object = m2,
505     contrast = list('p1vp2 Mono' = c(1, -1, 0, 0, 0, 0),
506     'p3vp4 Mono' = c(0, 0, 1, -1, 0, 0),
507     'p5vp6 Mono' = c(0, 0, 0, 0, 1, -1),
508     'p1vp3 Mono' = c(1, 0, -1, 0, 0, 0),
509     'p1vp4 Mono' = c(1, 0, 0, -1, 0, 0),
510     'p1vp5 Mono' = c(1, 0, 0, 0, -1, 0),
511     'p1vp6 Mono' = c(1, 0, 0, 0, 0, -1),
512     'p2vp3 Mono' = c(0, 1, -1, 0, 0, 0),
513     'p2vp4 Mono' = c(0, 1, 0, 0, -1, 0),
514     'p2vp5 Mono' = c(0, 1, 0, 0, -1, 0),
515     'p2vp6 Mono' = c(0, 1, 0, 0, 0, -1),
516     'p3vp5 Mono' = c(0, 0, 1, 0, -1, 0),
517     'p3vp6 Mono' = c(0, 0, 1, 0, 0, -1),
518     'p4vp5 Mono' = c(0, 0, 0, 1, -1, 0),
519     'p4vp6 Mono' = c(0, 0, 0, 0, 0, -1)))
520 summary(con1CI)
521
522 con11CI <- contrasts_DI(object = m2,
523     contrast = list('GvL FG' = c(0.5, 0.5, -0.5, -0.5, 0, 0),
524     'GvH FG' = c(0.5, 0.5, 0, 0, -0.5, -0.5),
525     'LvH FG' = c(0, 0, 0.5, 0.5, -0.5, -0.5)))
526 summary(con11CI)
527 conCI <- contrasts_DI(object = m2,
528     contrast = list('p1vmss' = c(0.83333,-0.16667,-0.16667,-0.16667,-0.16667,-
529     0.16667),
530     'p2vmss' = c(-0.16667,0.83333,-0.16667,-0.16667,-0.16667,-
531     0.16667),
532     'p3vmss' = c(-0.16667,-0.16667,0.83333,-0.16667,-0.16667,-
533     0.16667),

```



```

534           'p4vmss' = c(-0.16667,-0.16667,-0.16667, 0.83333,-0.16667,-
535 0.16667),
536           'p5vmss' = c(-0.16667,-0.16667,-0.16667,-0.16667,0.83333,-
537 0.16667),
538           'p6vmss' = c(-0.16667,-0.16667,-0.16667,-0.16667,-
539 0.16667,0.83333)))
540
541 summary(conCI)
542 predict(m2)
543 ###Analysis of fungal-feeders, variable 'FF'
544 hist(NemInd$FF)
545 NemInd$logFF<-log(NemInd$FF + 1)
546 hist(NemInd$logFF)
547 #2. Generate a scatterplot of the response vs richness
548 plot(NemInd$plantSR, NemInd$logFF, ylab = "Fungivores",
549      xlab = "richness")
550
551 #3. Fit the models
552 #STR
553 m701<- DI (y = "logFF", prop = 7:12, DImodel = "STR", data = NemInd)
554 summary(m701)
555
556 #ID
557
558 m702<- DI (y = "logFF", prop = 7:12, DImodel = "ID", data = NemInd)
559 summary(m702)
560
561 #AV
562 m703<- DI (y = "logFF", prop = 7:12, DImodel = "AV", data = NemInd)
563 summary(m703)
564
565 #FG
566 m704<- DI (y = "logFF", prop = 7:12, DImodel = "FG",

```

```

567         FG = c("FG1", "FG1", "FG2", "FG2", "FG2", "FG2"), data = NemInd)
568 summary(m704)
569
570 #FULL
571 m705<- DI (y = "logFF", prop = 7:12, DImodel = "FULL", data = NemInd)
572 summary(m705)
573 #MODEL EVALUATION
574 anova(m701, m702, m703, m704, m705, test="F")
575
576 auto700 <- autoDI(y = "logFF", prop = 7:12,
577         FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = NemInd, selection =
578 "Ftest")
579 summary(auto700)
580
581 #get_ternary(model = m703)
582 predict(m702)
583
584 #contrasts###
585 con1FF <- contrasts_DI(object = m702,
586         contrast = list('p1vp2 Mono' = c(1, -1, 0, 0, 0, 0),
587         'p3vp4 Mono' = c(0, 0, 1, -1, 0, 0),
588         'p5vp6 Mono' = c(0, 0, 0, 0, 1, -1),
589         'p1vp3 Mono' = c(1, 0, -1, 0, 0, 0),
590         'p1vp4 Mono' = c(1, 0, 0, -1, 0, 0),
591         'p1vp5 Mono' = c(1, 0, 0, 0, -1, 0),
592         'p1vp6 Mono' = c(1, 0, 0, 0, 0, -1),
593         'p2vp3 Mono' = c(0, 1, -1, 0, 0, 0),
594         'p2vp4 Mono' = c(0, 1, 0, 0, -1, 0),
595         'p2vp5 Mono' = c(0, 1, 0, 0, -1, 0),
596         'p2vp6 Mono' = c(0, 1, 0, 0, 0, -1),
597         'p3vp5 Mono' = c(0, 0, 1, 0, -1, 0),
598         'p3vp6 Mono' = c(0, 0, 1, 0, 0, -1),
599         'p4vp5 Mono' = c(0, 0, 0, 1, -1, 0),

```

```

600             'p4vp6 Mono' = c(0, 0, 0, 1, 0, -1)))
601 summary(con1FF)
602
603 con11FF <- contrasts_DI(object = m702,
604             contrast = list('GvL FG' = c(0.5, 0.5, -0.5, -0.5, 0, 0),
605                             'GvH FG' = c(0.5, 0.5, 0, 0, -0.5, -0.5),
606                             'LvH FG' = c(0, 0, 0.5, 0.5, -0.5, -0.5)))
607 summary(con11FF)
608 conFF <- contrasts_DI(object = m702,
609             contrast = list('p1vmss' = c(0.833333,-0.16667,-0.16667,-0.16667,-0.16667,-
610 0.16667),
611                             'p2vmss' = c(-0.16667,0.833333,-0.16667,-0.16667,-0.16667,-
612 0.16667),
613                             'p3vmss' = c(-0.16667,-0.16667,0.833333,-0.16667,-0.16667,-
614 0.16667),
615                             'p4vmss' = c(-0.16667,-0.16667,-0.16667, 0.833333,-0.16667,-
616 0.16667),
617                             'p5vmss' = c(-0.16667,-0.16667,-0.16667,-0.16667,0.833333,-
618 0.16667),
619                             'p6vmss' = c(-0.16667,-0.16667,-0.16667,-0.16667,-
620 0.16667,0.833333)))
621
622 summary(conFF)
623 #get_ternary(m702)
624 ###Omnivores+Predators
625 NemInd$OmvPred<-NemInd$Omv + NemInd$Pred
626 View(NemInd)
627 hist(NemInd$OmvPred)
628 NemInd$logOmvPred<-log(NemInd$OmvPred + 1)
629 hist(NemInd$logOmvPred)
630 #2. Generate a scatterplot of the response vs richness
631 plot(NemInd$plantSR, NemInd$logOmvPred, ylab = "Omnivores + Predators",
632       xlab = "richness")
633
634 #3. Fit the models

```

```
635 #STR
636 m706<- DI (y = "logOmvPred", prop = 7:12, DImodel = "STR", data = NemInd)
637 summary(m706)
638
639 #ID
640
641 m707<- DI (y = "logOmvPred", prop = 7:12, DImodel = "ID", data = NemInd)
642 summary(m707)
643
644 #AV
645 m708<- DI (y = "logOmvPred", prop = 7:12, DImodel = "AV", data = NemInd)
646 summary(m708)
647
648 #FG
649 m709<- DI (y = "logOmvPred", prop = 7:12, DImodel = "FG",
650           FG = c("FG1", "FG1", "FG2", "FG2", "FG2", "FG2"), data = NemInd)
651 summary(m709)
652
653 #FULL
654 m710<- DI (y = "logOmvPred", prop = 7:12, DImodel = "FULL", data = NemInd)
655 summary(m710)
656 #MODEL EVALUATION
657 anova(m706, m707, m708, m709, m710, test="F")
658
659 auto701 <- autoDI(y = "logOmvPred", prop = 7:12,
660                 FG = c("FG1", "FG1", "FG2", "FG2", "FG3", "FG3"), data = NemInd, selection =
661 "Ftest")
662 summary(auto701)
663
664 #get_ternary(model = m708)
665 predict(m708)
666
667 #contrasts####
```

```

668 con1OmvPred <- contrasts_DI(object = m708,
669     contrast = list('p1vp2 Mono' = c(1, -1, 0, 0, 0, 0, 0),
670     'p3vp4 Mono' = c(0, 0, 1, -1, 0, 0, 0),
671     'p5vp6 Mono' = c(0, 0, 0, 0, 1, -1, 0),
672     'p1vp3 Mono' = c(1, 0, -1, 0, 0, 0, 0),
673     'p1vp4 Mono' = c(1, 0, 0, -1, 0, 0, 0),
674     'p1vp5 Mono' = c(1, 0, 0, 0, -1, 0, 0),
675     'p1vp6 Mono' = c(1, 0, 0, 0, 0, -1, 0),
676     'p2vp3 Mono' = c(0, 1, -1, 0, 0, 0, 0),
677     'p2vp4 Mono' = c(0, 1, 0, 0, -1, 0, 0),
678     'p2vp5 Mono' = c(0, 1, 0, 0, -1, 0, 0),
679     'p2vp6 Mono' = c(0, 1, 0, 0, 0, -1, 0),
680     'p3vp5 Mono' = c(0, 0, 1, 0, -1, 0, 0),
681     'p3vp6 Mono' = c(0, 0, 1, 0, 0, -1, 0),
682     'p4vp5 Mono' = c(0, 0, 0, 1, -1, 0, 0),
683     'p4vp6 Mono' = c(0, 0, 0, 0, 0, -1, 0)))
684 summary(con1OmvPred)
685
686 con11OmvPred <- contrasts_DI(object = m708,
687     contrast = list('GvL FG' = c(0.5, 0.5, -0.5, -0.5, 0, 0, 0),
688     'GvH FG' = c(0.5, 0.5, 0, 0, -0.5, -0.5, 0),
689     'LvH FG' = c(0, 0, 0.5, 0.5, -0.5, -0.5, 0)))
690 summary(con11OmvPred)
691 conOmvPred <- contrasts_DI(object = m708,
692     contrast = list('p1vmss' = c(0.83333,-0.16667,-0.16667,-0.16667,-0.16667,-
693     0.16667, -0.41667),
694     'p2vmss' = c(-0.16667,0.83333,-0.16667,-0.16667,-0.16667,-
695     0.16667, -0.41667),
696     'p3vmss' = c(-0.16667,-0.16667,0.83333,-0.16667,-0.16667,-
697     0.16667, -0.41667),
698     'p4vmss' = c(-0.16667,-0.16667,-0.16667, 0.83333,-0.16667,-
699     0.16667, -0.41667),
700     'p5vmss' = c(-0.16667,-0.16667,-0.16667,-0.16667,0.83333,-
701     0.16667, -0.41667),

```

```

702           'p6vmss' = c(-0.16667,-0.16667,-0.16667,-0.16667,-
703 0.16667,0.83333, -0.41667)))
704 summary(conOmvPred)
705 predict(m708)
706
707
708 #####Ternary plots#####
709 library(Ternary)
710 library(GA)
711 library(ggplot2)
712 library(lattice)
713 #Check that you have the selected models for the response of interest. For example,
714 m101<- DI (y = "MI", prop = 7:12, DImodel = "AV", data = NemInd)
715 summary(m101)
716
717 m113<- DI (y = "EI", prop = 7:12, DImodel = "AV", data = NemInd)
718 summary(m113)
719
720 m117<- DI (y = "SI", prop = 7:12, DImodel = "AV", data = NemInd)
721 summary(m117)
722 #define dimensions
723 height=100
724 width=85
725 # Create mapping of 3d simplex space in 2d
726 trian <- expand.grid(base=seq(0,1,l=height*2), high=seq(0,sin(pi/3),l=width*2))
727 trian <- subset(trian, (base*sin(pi/3)*2)>high)
728 trian <- subset(trian, ((1-base)*sin(pi/3)*2)>high)
729
730 # Map x and y coordinates to species proportions
731 trian$g2 <- trian$high*2/sqrt(3)
732 trian$g3 <- trian$base-trian$high/sqrt(3)
733 trian$g1 <- 1-trian$g3-trian$g2
734

```

```

735 #####SP1 TOP OF TRIANGLE, SP2 BOTTOM LEFT OF TRIANGLE, SP3 BOTTOM RIGHT
736 OF TRIANGLE
737 trian$p1 <- trian$g1*0
738 trian$p2 <- trian$g1*1
739 trian$p3 <- trian$g2*0
740 trian$p4 <- trian$g2*1
741 trian$p5 <- trian$g3*0
742 trian$p6 <- trian$g3*1
743
744 ##### Predict for points within the simplex
745 trian$yhat <- predict(m103, newdata = trian)
746
747 #####DIVIDE EACH PREDICTION BY THE MAX PREDICTION FOR GRAPHICAL
748 COMPARISONS ON A SCALE OF 0 TO 1
749 #trian$yhatT=trian$yhat/max(trian$yhat)
750 summary(trian$yhat)
751 #summary(trian$yhatT)
752 #Better to keep on the original scale
753
754 #####SETUP THE BASIC CONTOUR DIAGRAM
755 grade.trellis <- function(from=0.2, to=0.8, step=0.2, col=1, lty=2, lwd=0.5){
756   x1 <- seq(from, to, step)
757   x2 <- x1/2
758   y2 <- x1*sqrt(3)/2
759   x3 <- (1-x1)*0.5+x1
760   y3 <- sqrt(3)/2-x1*sqrt(3)/2
761   panel.segments(x1, 0, x2, y2, col=col, lty=lty, lwd=lwd)
762   panel.text(x1, 0, label=x1, pos=1)
763   panel.segments(x1, 0, x3, y3, col=col, lty=lty, lwd=lwd)
764   panel.text(x2, y2, label=rev(x1), pos=2)
765   panel.segments(x2, y2, 1-x2, y2, col=col, lty=lty, lwd=lwd)
766   panel.text(x3, y3, label=rev(x1), pos=4)
767 }

```

```

768
769 #####COLOUR FOR THE TERNARY DIAGRAM#####
770 revterrain.colors <- function (n, alpha = 1)
771 {
772   if ((n <- as.integer(n[1L])) > 0) {
773     k <- n%%2
774     h <- rev(c(4/12, 2/12, 0/12))
775     s <- rev(c(1, 1, 0))
776     v <- rev(c(0.65, 0.9, 0.95))
777     c(hsv(h = seq.int(h[1L], h[2L], length.out = k), s = seq.int(s[1L],
778                               s[2L], length.out = k), v = seq.int(v[1L], v[2L],
779                               length.out = k), alpha =
780 alpha), hsv(h = seq.int(h[2L],
781 h[3L], length.out = n - k + 1)[-1L], s = seq.int(s[2L],
782 s[3L], length.out = n - k + 1)[-1L], v = seq.int(v[2L],
783 v[3L], length.out = n - k + 1)[-1L], alpha = alpha))
784   }
785   else character()
786 }
787 }
788 }
789 }
790
791 levelplot(yhat~base*high, trian, aspect="iso", xlim=c(-0.1,1.1),
792           ylim=c(-0.1,0.96), main = "B) Log CP 3-5",
793           xlab=NULL, ylab=NULL, contour=TRUE,
794           col.regions=revterrain.colors,
795           colorkey=list(space="bottom", axis.line=list(col=1),
796                         axis.text=list(col=1)),
797           at=seq(3.7,5.1,0.2),
798           #use different values if you want different number of contours, for instance,
799           "at=seq(from,to,spacing)"
800           par.settings=list(axis.line=list(col=NA), axis.text=list(col=NA)),
801           panel=function(..., at, contour=TRUE, labels=TRUE){

```



```
802     panel.levelplot(..., at=at, #contour=contour,
803         labels=labels,
804         lty=3, lwd=0.01, col=1)
805     })
806
807
808     #####PUT LABELS FOR EACH SPECIES ON THE VERTICES OF THE BASIC CONTOUR
809     DIAGRAM
810     trellis.focus("panel", 1, 1, highlight=FALSE)
811     panel.segments(c(0,0,0.5), c(0,0,sqrt(3)/2), c(1,1/2,1), c(0,sqrt(3)/2,0))
812     grade.trellis()
813     panel.text(0, 0, label="G", pos=2)
814     panel.text(1/2, sqrt(3)/2, label="L", pos=3)
815     panel.text(1, 0, label="H", pos=4)
816
```