

Beta diversity: unconstrained ordination (NMDS)

Stijn Schreven

23 April 2022

Contents

Introduction	1
Load packages	2
Input files	2
1. Prepare data	2
1.1. Transform and subsets	2
1.2. Distance matrices	2
1.3. Plot presets	3
2. Ordinations	3
2.1. All diets	3
2.2. Chicken feed	6
2.3. Camelina	10
2.4. Chicken manure	14
3. Plots	17
3.1. All diets	17
3.2. Chicken feed	19
3.3. Camelina	20
3.4. Chicken manure	21
4. Export plots	22

Introduction

We used NMDS because weighed UniFrac distances are semi-metric and would give negative eigenvalues and complex/imaginary axes in PCoA (which can be corrected but these corrections have their own drawbacks).

NMDS is rank-based and robust for non-metric distances. We used Shepard stress plots to assess the fit of the NMDS solution to the weighted UniFrac distances. All analyses were done at genus level, on relative abundance data, with weighted UniFrac distances.

N.B.: In contrast to `7__betadiversity__nmads.Rmd`, the code in this R-markdown file uses all data for the NMDS, *i.e.* including samples of day 0, larval density 0, and both substrates and larvae. This is done for clarity in the manuscript submitted to *Applied and Environmental Microbiology* (Figure 3). The input for the NMDS does not need to be a balanced dataset; for the dbRDA the dataset should be balanced.

Load packages

```
library(phyloseq)
library(microbiome)
library(microbiomeutilities)
library(vegan)
library(plyr)
library(sciplot)
library(ggplot2)
library(ggrepel)
library(ggpubr)
library(viridis)
```

Input files

```
pstot <- readRDS("../phyobjects/ps1.exp.rds")
```

1. Prepare data

1.1. Transform and subsets

```
# transform data to genus level, relative abundance
pstot.g <- aggregate_taxa(pstot, "Genus")
pstot.g.r <- microbiome::transform(pstot.g, "compositional")

# per diet separately
CF <- subset_samples(pstot.g.r, Diet == "CF")
CF <- prune_taxa(taxa_sums(otu_table(CF)) > 0, CF)
CS <- subset_samples(pstot.g.r, Diet == "CS")
CS <- prune_taxa(taxa_sums(otu_table(CS)) > 0, CS)
CM <- subset_samples(pstot.g.r, Diet == "CM")
CM <- prune_taxa(taxa_sums(otu_table(CM)) > 0, CM)
```

1.2. Distance matrices

```
wuf.dm <- distance(pstot.g.r, "wunifrac")
wuf.cf0 <- distance(CF, "wunifrac")
wuf.cs0 <- distance(CS, "wunifrac")
wuf.cm0 <- distance(CM, "wunifrac")
```

1.3. Plot presets

```
theme_nmds <- theme_classic() +
  theme(panel.grid = element_blank(),
        panel.spacing = unit(.5, "lines"),
        panel.border = element_rect(color = "black", fill = NA, size = .5),
        strip.background = element_blank(),
        text = element_text(size = 20))

labs_nmds <- as_labeller(c("0" = "0 larvae\nper container",
                           "50" = "50 larvae\nper container",
                           "100" = "100 larvae\nper container",
                           "200" = "200 larvae\nper container"))
```

2. Ordinations

2.1. All diets

```
set.seed(200)
nmds.tot <- metaMDS(comm = wuf.dm, autotransform = F,
                    k = 2, try = 100, trymax = 200)

## Run 0 stress 0.1316834
## Run 1 stress 0.2163065
## Run 2 stress 0.1397161
## Run 3 stress 0.1286777
## ... New best solution
## ... Procrustes: rmse 0.0076921 max resid 0.1064141
## Run 4 stress 0.22013
## Run 5 stress 0.1337652
## Run 6 stress 0.1295432
## Run 7 stress 0.1397168
## Run 8 stress 0.1356613
## Run 9 stress 0.133061
## Run 10 stress 0.1348032
## Run 11 stress 0.1295429
## Run 12 stress 0.207428
## Run 13 stress 0.1508278
## Run 14 stress 0.2150813
## Run 15 stress 0.1455672
## Run 16 stress 0.1992187
## Run 17 stress 0.2077905
```

```

## Run 18 stress 0.1320922
## Run 19 stress 0.1320933
## Run 20 stress 0.2182341
## Run 21 stress 0.1286836
## ... Procrustes: rmse 0.000599764  max resid 0.00938632
## ... Similar to previous best
## Run 22 stress 0.1310014
## Run 23 stress 0.1331767
## Run 24 stress 0.1642493
## Run 25 stress 0.1321405
## Run 26 stress 0.4178767
## Run 27 stress 0.2076075
## Run 28 stress 0.1342047
## Run 29 stress 0.1376313
## Run 30 stress 0.2215897
## Run 31 stress 0.1311767
## Run 32 stress 0.1353046
## Run 33 stress 0.1446653
## Run 34 stress 0.1321294
## Run 35 stress 0.1295429
## Run 36 stress 0.1332029
## Run 37 stress 0.1352751
## Run 38 stress 0.2136956
## Run 39 stress 0.1286779
## ... Procrustes: rmse 7.517783e-05  max resid 0.0007080376
## ... Similar to previous best
## Run 40 stress 0.2261654
## Run 41 stress 0.1335034
## Run 42 stress 0.1306405
## Run 43 stress 0.1286833
## ... Procrustes: rmse 0.0006051086  max resid 0.009484363
## ... Similar to previous best
## Run 44 stress 0.2265479
## Run 45 stress 0.1351849
## Run 46 stress 0.1575956
## Run 47 stress 0.1447118
## Run 48 stress 0.1791714
## Run 49 stress 0.147003
## Run 50 stress 0.1443988
## Run 51 stress 0.1386055
## Run 52 stress 0.1334537
## Run 53 stress 0.2183003
## Run 54 stress 0.1309617
## Run 55 stress 0.1337648
## Run 56 stress 0.1652433
## Run 57 stress 0.1330688
## Run 58 stress 0.1328479
## Run 59 stress 0.1334205
## Run 60 stress 0.1360101
## Run 61 stress 0.2136349
## Run 62 stress 0.1295432
## Run 63 stress 0.1356273
## Run 64 stress 0.1606468
## Run 65 stress 0.1362817

```

```

## Run 66 stress 0.1334121
## Run 67 stress 0.1363124
## Run 68 stress 0.2102316
## Run 69 stress 0.1286786
## ... Procrustes: rmse 0.000607353  max resid 0.009545193
## ... Similar to previous best
## Run 70 stress 0.131175
## Run 71 stress 0.1352753
## Run 72 stress 0.1311766
## Run 73 stress 0.133061
## Run 74 stress 0.1286842
## ... Procrustes: rmse 0.0005720992  max resid 0.009089687
## ... Similar to previous best
## Run 75 stress 0.2031861
## Run 76 stress 0.2201049
## Run 77 stress 0.1286781
## ... Procrustes: rmse 0.0001089905  max resid 0.001363725
## ... Similar to previous best
## Run 78 stress 0.1687933
## Run 79 stress 0.2200081
## Run 80 stress 0.1287097
## ... Procrustes: rmse 0.001056286  max resid 0.0143328
## Run 81 stress 0.1316587
## Run 82 stress 0.1286786
## ... Procrustes: rmse 0.0005368265  max resid 0.008437238
## ... Similar to previous best
## Run 83 stress 0.1399967
## Run 84 stress 0.1341978
## Run 85 stress 0.4179729
## Run 86 stress 0.2081215
## Run 87 stress 0.129543
## Run 88 stress 0.1364131
## Run 89 stress 0.2149657
## Run 90 stress 0.1356189
## Run 91 stress 0.1365063
## Run 92 stress 0.2055351
## Run 93 stress 0.1351791
## Run 94 stress 0.1313618
## Run 95 stress 0.1345443
## Run 96 stress 0.1856204
## Run 97 stress 0.1632412
## Run 98 stress 0.1306398
## Run 99 stress 0.2143027
## Run 100 stress 0.1286782
## ... Procrustes: rmse 0.0005007309  max resid 0.007990398
## ... Similar to previous best
## *** Solution reached

```

```

nmds.tot # convergent solution, stress = 0.129

```

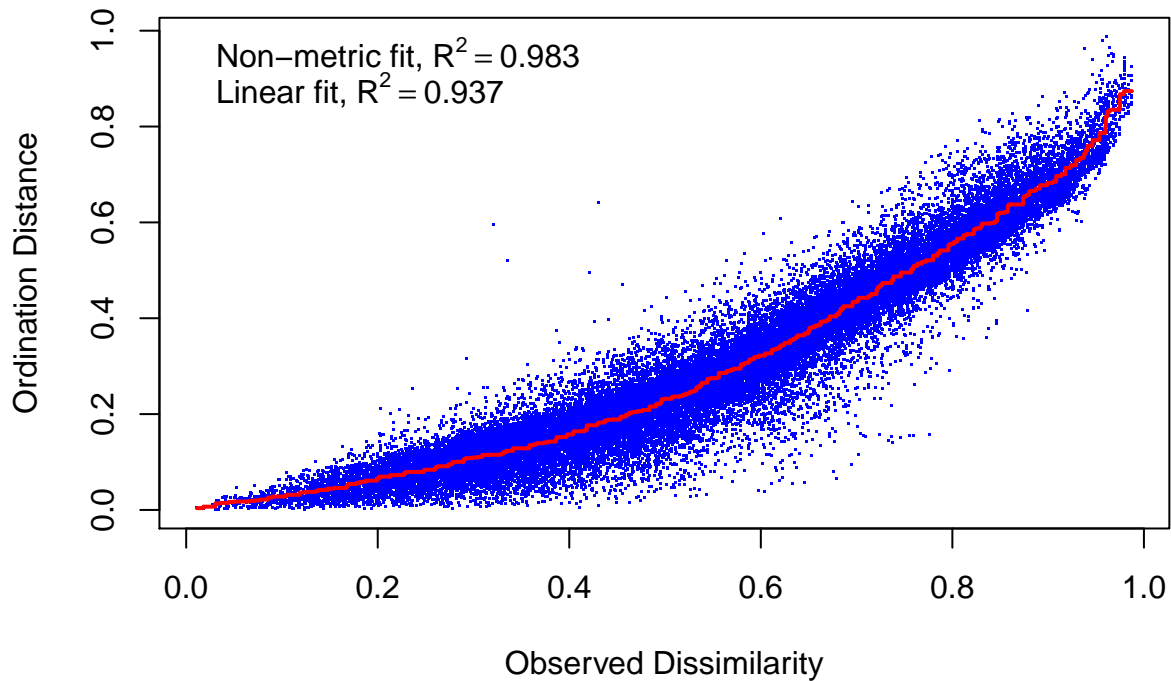
```

##
## Call:
## metaMDS(comm = wuf.dm, k = 2, try = 100, trymax = 200, autotransform = F)
##

```

```
## global Multidimensional Scaling using monoMDS
##
## Data:      wuf.dm
## Distance:  user supplied
##
## Dimensions: 2
## Stress:    0.1286777
## Stress type 1, weak ties
## Two convergent solutions found after 100 tries
## Scaling:  centring, PC rotation
## Species:  scores missing
```

```
# Shepard stress plot
stressplot(nmds.tot)
```



2.2. Chicken feed

```
set.seed(200)
nmds.cf0 <- metaMDS(comm = wuf.cf0, autotransform = F,
                    k = 2, try = 100, trymax = 200)
```

```
## Run 0 stress 0.0900147
## Run 1 stress 0.1115758
```

```

## Run 2 stress 0.1065697
## Run 3 stress 0.1277292
## Run 4 stress 0.1358196
## Run 5 stress 0.1304382
## Run 6 stress 0.1095859
## Run 7 stress 0.1141773
## Run 8 stress 0.1285829
## Run 9 stress 0.1240983
## Run 10 stress 0.1297031
## Run 11 stress 0.09455317
## Run 12 stress 0.1261765
## Run 13 stress 0.1138541
## Run 14 stress 0.0899934
## ... New best solution
## ... Procrustes: rmse 0.002689983 max resid 0.0199149
## Run 15 stress 0.1301101
## Run 16 stress 0.08999348
## ... Procrustes: rmse 1.816807e-05 max resid 8.704491e-05
## ... Similar to previous best
## Run 17 stress 0.1323283
## Run 18 stress 0.1194562
## Run 19 stress 0.1296164
## Run 20 stress 0.1312852
## Run 21 stress 0.1116682
## Run 22 stress 0.1004764
## Run 23 stress 0.1394093
## Run 24 stress 0.0900146
## ... Procrustes: rmse 0.002649303 max resid 0.01987411
## Run 25 stress 0.1145644
## Run 26 stress 0.08999344
## ... Procrustes: rmse 2.957298e-05 max resid 0.000134406
## ... Similar to previous best
## Run 27 stress 0.1365552
## Run 28 stress 0.1369195
## Run 29 stress 0.1390169
## Run 30 stress 0.0900148
## ... Procrustes: rmse 0.002692891 max resid 0.01993231
## Run 31 stress 0.1358533
## Run 32 stress 0.1038932
## Run 33 stress 0.1045377
## Run 34 stress 0.09002301
## ... Procrustes: rmse 0.001300742 max resid 0.00713403
## ... Similar to previous best
## Run 35 stress 0.1341714
## Run 36 stress 0.1131354
## Run 37 stress 0.1361035
## Run 38 stress 0.0983174
## Run 39 stress 0.09455419
## Run 40 stress 0.1167758
## Run 41 stress 0.1173976
## Run 42 stress 0.1312109
## Run 43 stress 0.1162399
## Run 44 stress 0.1332007
## Run 45 stress 0.1186391

```

```

## Run 46 stress 0.1362496
## Run 47 stress 0.1303314
## Run 48 stress 0.09452312
## Run 49 stress 0.1356959
## Run 50 stress 0.1211159
## Run 51 stress 0.1326433
## Run 52 stress 0.127024
## Run 53 stress 0.1077586
## Run 54 stress 0.1049889
## Run 55 stress 0.1157281
## Run 56 stress 0.09002275
## ... Procrustes: rmse 0.001263617 max resid 0.007032531
## ... Similar to previous best
## Run 57 stress 0.1147498
## Run 58 stress 0.1004763
## Run 59 stress 0.09461771
## Run 60 stress 0.09001458
## ... Procrustes: rmse 0.002646265 max resid 0.0198657
## Run 61 stress 0.1374503
## Run 62 stress 0.1199688
## Run 63 stress 0.1291434
## Run 64 stress 0.1265864
## Run 65 stress 0.09002822
## ... Procrustes: rmse 0.001831445 max resid 0.009016502
## ... Similar to previous best
## Run 66 stress 0.1199395
## Run 67 stress 0.1216991
## Run 68 stress 0.09001469
## ... Procrustes: rmse 0.002688561 max resid 0.01992224
## Run 69 stress 0.1281243
## Run 70 stress 0.09002834
## ... Procrustes: rmse 0.001854145 max resid 0.009081236
## ... Similar to previous best
## Run 71 stress 0.1212999
## Run 72 stress 0.09001459
## ... Procrustes: rmse 0.002647338 max resid 0.01986858
## Run 73 stress 0.1187321
## Run 74 stress 0.1382914
## Run 75 stress 0.1378728
## Run 76 stress 0.1349464
## Run 77 stress 0.1141775
## Run 78 stress 0.125102
## Run 79 stress 0.1328545
## Run 80 stress 0.09461784
## Run 81 stress 0.1356936
## Run 82 stress 0.08999339
## ... New best solution
## ... Procrustes: rmse 9.686323e-06 max resid 4.650523e-05
## ... Similar to previous best
## Run 83 stress 0.1275397
## Run 84 stress 0.1004764
## Run 85 stress 0.08999338
## ... New best solution
## ... Procrustes: rmse 2.095096e-05 max resid 0.0001146027

```

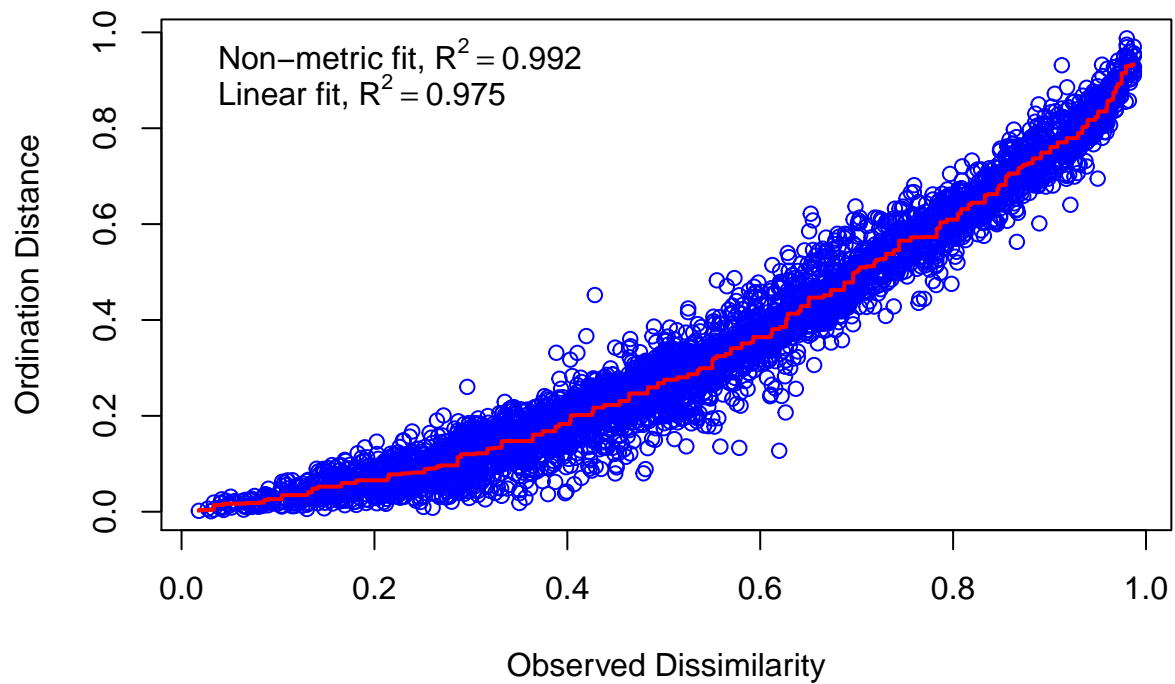


```
## ... Similar to previous best
## Run 86 stress 0.1286641
## Run 87 stress 0.125655
## Run 88 stress 0.0900147
## ... Procrustes: rmse 0.002694868  max resid 0.01994453
## Run 89 stress 0.1335838
## Run 90 stress 0.1082542
## Run 91 stress 0.1195011
## Run 92 stress 0.1220158
## Run 93 stress 0.1180669
## Run 94 stress 0.132569
## Run 95 stress 0.1080056
## Run 96 stress 0.1276151
## Run 97 stress 0.4121561
## Run 98 stress 0.1394866
## Run 99 stress 0.1082032
## Run 100 stress 0.411981
## *** Solution reached
```

```
nm.ds.cf0 # convergent solution, stress = 0.090
```

```
##
## Call:
## metaMDS(comm = wuf.cf0, k = 2, try = 100, trymax = 200, autotransform = F)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      wuf.cf0
## Distance: user supplied
##
## Dimensions: 2
## Stress:      0.08999338
## Stress type 1, weak ties
## Two convergent solutions found after 100 tries
## Scaling: centring, PC rotation
## Species: scores missing
```

```
# Shepard stress plot
stressplot(nm.ds.cf0)
```



2.3. Camelina

```
set.seed(200)
nmms.cs0 <- metaMDS(comm = wuf.cs0, autotransform = F,
                    k = 2, try = 100, trymax = 200)

## Run 0 stress 0.1314098
## Run 1 stress 0.1569723
## Run 2 stress 0.1398768
## Run 3 stress 0.1249017
## ... New best solution
## ... Procrustes: rmse 0.03272818  max resid 0.2889195
## Run 4 stress 0.1362114
## Run 5 stress 0.1463317
## Run 6 stress 0.1424764
## Run 7 stress 0.1356223
## Run 8 stress 0.1463317
## Run 9 stress 0.17663
## Run 10 stress 0.1679679
## Run 11 stress 0.1458275
## Run 12 stress 0.1315574
## Run 13 stress 0.1252093
## ... Procrustes: rmse 0.01096793  max resid 0.09867119
```

```

## Run 14 stress 0.130616
## Run 15 stress 0.1811071
## Run 16 stress 0.139883
## Run 17 stress 0.1249178
## ... Procrustes: rmse 0.001739357  max resid 0.01325786
## Run 18 stress 0.1463107
## Run 19 stress 0.1355232
## Run 20 stress 0.1248576
## ... New best solution
## ... Procrustes: rmse 0.00103429  max resid 0.007103321
## ... Similar to previous best
## Run 21 stress 0.1684253
## Run 22 stress 0.1341262
## Run 23 stress 0.1527062
## Run 24 stress 0.1248576
## ... New best solution
## ... Procrustes: rmse 6.749681e-06  max resid 3.164506e-05
## ... Similar to previous best
## Run 25 stress 0.1305928
## Run 26 stress 0.165851
## Run 27 stress 0.1475085
## Run 28 stress 0.1251707
## ... Procrustes: rmse 0.01096725  max resid 0.09866382
## Run 29 stress 0.1520256
## Run 30 stress 0.1341514
## Run 31 stress 0.1314275
## Run 32 stress 0.1618491
## Run 33 stress 0.1424377
## Run 34 stress 0.1362363
## Run 35 stress 0.1315514
## Run 36 stress 0.1251707
## ... Procrustes: rmse 0.01096713  max resid 0.09866148
## Run 37 stress 0.1512861
## Run 38 stress 0.1443204
## Run 39 stress 0.1349369
## Run 40 stress 0.1458266
## Run 41 stress 0.130162
## Run 42 stress 0.1349049
## Run 43 stress 0.130162
## Run 44 stress 0.1398875
## Run 45 stress 0.1398496
## Run 46 stress 0.1472078
## Run 47 stress 0.1646223
## Run 48 stress 0.1398046
## Run 49 stress 0.1398829
## Run 50 stress 0.1666579
## Run 51 stress 0.1569507
## Run 52 stress 0.1305928
## Run 53 stress 0.1676928
## Run 54 stress 0.1612215
## Run 55 stress 0.1305583
## Run 56 stress 0.1400497
## Run 57 stress 0.1340413
## Run 58 stress 0.1251707

```

```

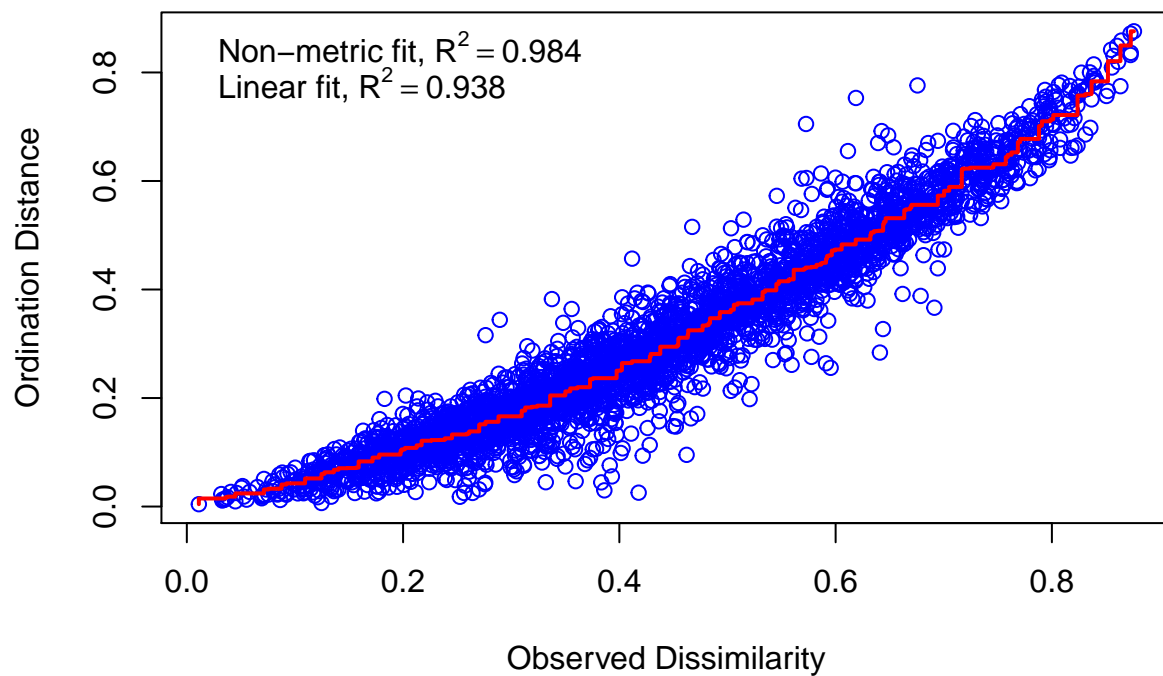
## ... Procrustes: rmse 0.01096699  max resid 0.09866169
## Run 59 stress 0.1923804
## Run 60 stress 0.168006
## Run 61 stress 0.1526947
## Run 62 stress 0.1885866
## Run 63 stress 0.1248576
## ... Procrustes: rmse 3.476941e-06  max resid 1.662543e-05
## ... Similar to previous best
## Run 64 stress 0.1306513
## Run 65 stress 0.1248576
## ... Procrustes: rmse 8.350872e-06  max resid 4.140651e-05
## ... Similar to previous best
## Run 66 stress 0.1677263
## Run 67 stress 0.1382832
## Run 68 stress 0.1397402
## Run 69 stress 0.1375324
## Run 70 stress 0.1429809
## Run 71 stress 0.1432578
## Run 72 stress 0.1473007
## Run 73 stress 0.1633743
## Run 74 stress 0.130162
## Run 75 stress 0.1314113
## Run 76 stress 0.1355667
## Run 77 stress 0.1443204
## Run 78 stress 0.1400497
## Run 79 stress 0.1424768
## Run 80 stress 0.1398785
## Run 81 stress 0.1809786
## Run 82 stress 0.1715283
## Run 83 stress 0.1398786
## Run 84 stress 0.1616122
## Run 85 stress 0.1249017
## ... Procrustes: rmse 0.001035521  max resid 0.00710532
## ... Similar to previous best
## Run 86 stress 0.1398867
## Run 87 stress 0.1307662
## Run 88 stress 0.130162
## Run 89 stress 0.1437908
## Run 90 stress 0.1354569
## Run 91 stress 0.1563267
## Run 92 stress 0.1382832
## Run 93 stress 0.1648675
## Run 94 stress 0.1248576
## ... Procrustes: rmse 9.771061e-06  max resid 5.073136e-05
## ... Similar to previous best
## Run 95 stress 0.1611749
## Run 96 stress 0.1467448
## Run 97 stress 0.1460387
## Run 98 stress 0.1251707
## ... Procrustes: rmse 0.01096717  max resid 0.09866205
## Run 99 stress 0.142547
## Run 100 stress 0.1427314
## *** Solution reached

```

```
nm.ds.cs0 # convergent solution, stress = 0.125
```

```
##  
## Call:  
## metaMDS(comm = wuf.cs0, k = 2, try = 100, trymax = 200, autotransform = F)  
##  
## global Multidimensional Scaling using monoMDS  
##  
## Data:      wuf.cs0  
## Distance: user supplied  
##  
## Dimensions: 2  
## Stress:    0.1248576  
## Stress type 1, weak ties  
## Two convergent solutions found after 100 tries  
## Scaling: centring, PC rotation  
## Species: scores missing
```

```
# Shepard stress plot  
stressplot(nm.ds.cs0)
```



2.4. Chicken manure

```
set.seed(200)
nm.ds.cm0 <- metaMDS(comm = wuf.cm0, autotransform = F,
                     k = 2, try = 100, trymax = 200)

## Run 0 stress 0.1359578
## Run 1 stress 0.1360547
## ... Procrustes: rmse 0.003831689  max resid 0.03694011
## Run 2 stress 0.1463734
## Run 3 stress 0.1382728
## Run 4 stress 0.1377937
## Run 5 stress 0.1767213
## Run 6 stress 0.1654412
## Run 7 stress 0.1697882
## Run 8 stress 0.1357036
## ... New best solution
## ... Procrustes: rmse 0.007182809  max resid 0.06090273
## Run 9 stress 0.1659473
## Run 10 stress 0.1625686
## Run 11 stress 0.1613195
## Run 12 stress 0.1360538
## ... Procrustes: rmse 0.007605181  max resid 0.06051073
## Run 13 stress 0.1828163
## Run 14 stress 0.1370219
## Run 15 stress 0.1670075
## Run 16 stress 0.1555838
## Run 17 stress 0.1751425
## Run 18 stress 0.1330838
## ... New best solution
## ... Procrustes: rmse 0.01052871  max resid 0.08843053
## Run 19 stress 0.1683484
## Run 20 stress 0.1510829
## Run 21 stress 0.1613012
## Run 22 stress 0.1606272
## Run 23 stress 0.1542343
## Run 24 stress 0.1603797
## Run 25 stress 0.1331037
## ... Procrustes: rmse 0.001217063  max resid 0.009356751
## ... Similar to previous best
## Run 26 stress 0.1613463
## Run 27 stress 0.140741
## Run 28 stress 0.1703991
## Run 29 stress 0.1523828
## Run 30 stress 0.1331045
## ... Procrustes: rmse 0.00151874  max resid 0.009729704
## ... Similar to previous best
## Run 31 stress 0.1356177
## Run 32 stress 0.1493827
## Run 33 stress 0.1434641
## Run 34 stress 0.1609576
## Run 35 stress 0.1669154
## Run 36 stress 0.1669709
```

```

## Run 37 stress 0.1563485
## Run 38 stress 0.1431574
## Run 39 stress 0.1339057
## Run 40 stress 0.1632812
## Run 41 stress 0.1612521
## Run 42 stress 0.1331043
## ... Procrustes: rmse 0.001293678  max resid 0.009797952
## ... Similar to previous best
## Run 43 stress 0.1647113
## Run 44 stress 0.164112
## Run 45 stress 0.1339718
## Run 46 stress 0.1526019
## Run 47 stress 0.1586995
## Run 48 stress 0.1360551
## Run 49 stress 0.1489613
## Run 50 stress 0.1579312
## Run 51 stress 0.154239
## Run 52 stress 0.1688564
## Run 53 stress 0.1464325
## Run 54 stress 0.1360455
## Run 55 stress 0.1562436
## Run 56 stress 0.1596744
## Run 57 stress 0.140502
## Run 58 stress 0.1731303
## Run 59 stress 0.1549823
## Run 60 stress 0.4119824
## Run 61 stress 0.164813
## Run 62 stress 0.1331186
## ... Procrustes: rmse 0.003736819  max resid 0.03345005
## Run 63 stress 0.1414963
## Run 64 stress 0.1465396
## Run 65 stress 0.1658103
## Run 66 stress 0.1603984
## Run 67 stress 0.1625131
## Run 68 stress 0.1615488
## Run 69 stress 0.1716889
## Run 70 stress 0.1583026
## Run 71 stress 0.1402613
## Run 72 stress 0.1580212
## Run 73 stress 0.1701335
## Run 74 stress 0.1644419
## Run 75 stress 0.1350472
## Run 76 stress 0.1650427
## Run 77 stress 0.1339713
## Run 78 stress 0.1713735
## Run 79 stress 0.1546565
## Run 80 stress 0.1769311
## Run 81 stress 0.1666482
## Run 82 stress 0.1622313
## Run 83 stress 0.1330839
## ... Procrustes: rmse 0.0001095239  max resid 0.0008841646
## ... Similar to previous best
## Run 84 stress 0.1330848
## ... Procrustes: rmse 0.0008559171  max resid 0.006238747

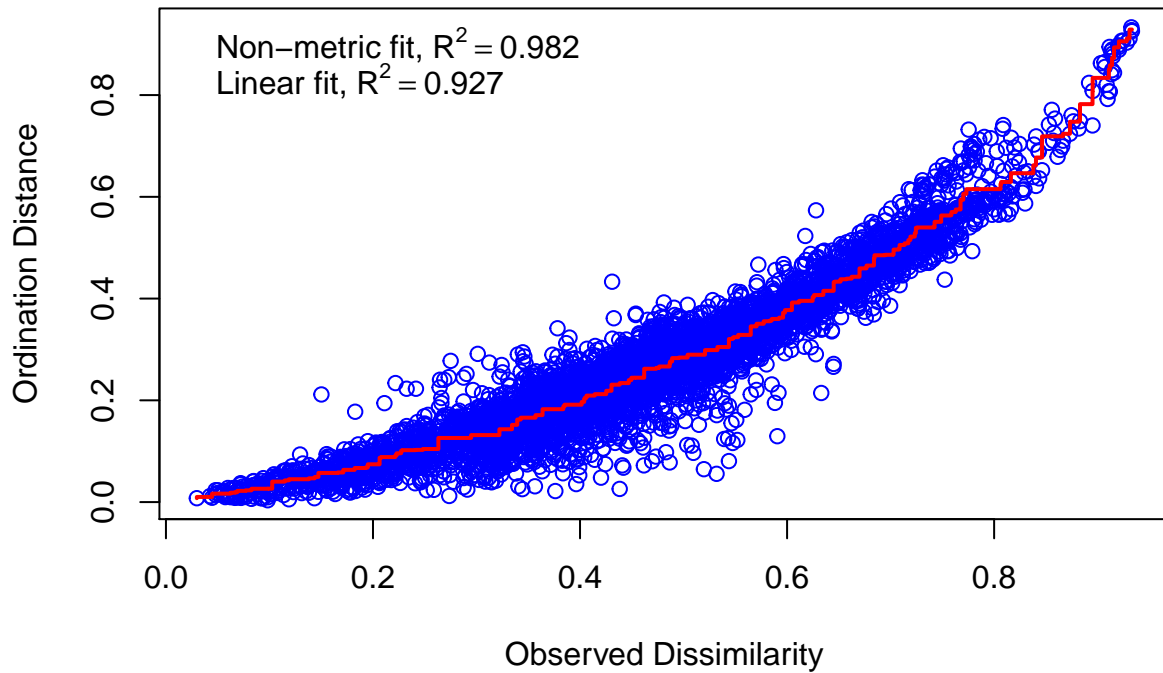
```

```
## ... Similar to previous best
## Run 85 stress 0.145159
## Run 86 stress 0.139347
## Run 87 stress 0.1652316
## Run 88 stress 0.171926
## Run 89 stress 0.162854
## Run 90 stress 0.1532136
## Run 91 stress 0.1356255
## Run 92 stress 0.1377094
## Run 93 stress 0.1464458
## Run 94 stress 0.1593895
## Run 95 stress 0.159071
## Run 96 stress 0.1580536
## Run 97 stress 0.1629052
## Run 98 stress 0.135958
## Run 99 stress 0.1478083
## Run 100 stress 0.1350478
## *** Solution reached
```

```
nmfs.cm0 # convergent solution, stress = 0.133
```

```
##
## Call:
## metaMDS(comm = wuf.cm0, k = 2, try = 100, trymax = 200, autotransform = F)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      wuf.cm0
## Distance: user supplied
##
## Dimensions: 2
## Stress:      0.1330838
## Stress type 1, weak ties
## Two convergent solutions found after 100 tries
## Scaling: centring, PC rotation
## Species: scores missing
```

```
# Shepard stress plot
stressplot(nmfs.cm0)
```

3. Plots

Figure 3 in manuscript submitted to *Applied and Environmental Microbiology*.

3.1. All diets

Figure 3A in manuscript.

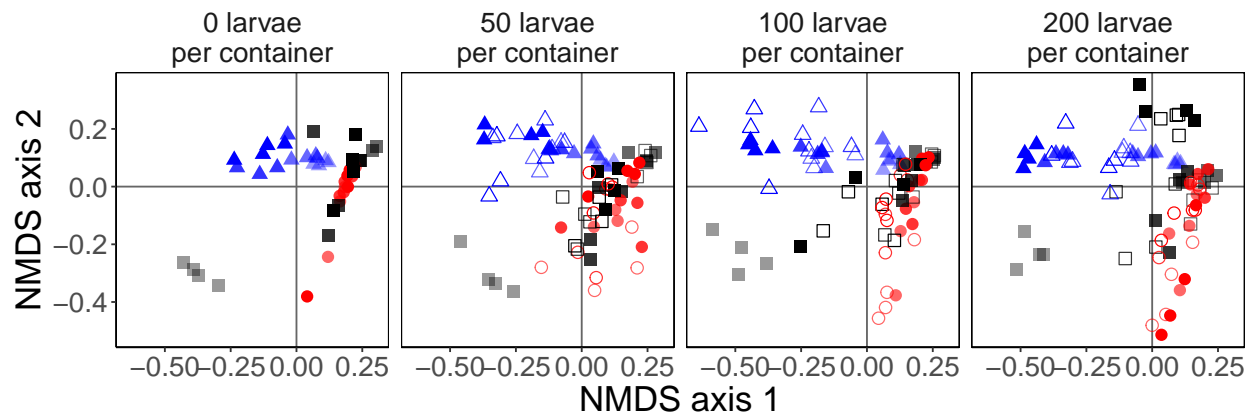
```
# extract plotting data
nmds.tot.df <- plot_ordination(pstot.g.r, nmds.tot, "samples", axes = 1:2, justDF = T)
# rename levels of Diet
nmds.tot.df$Diet <- revalue(nmds.tot.df$Diet, c("CF"="chicken feed",
                                                "CS"="camelina", "CM"="chicken manure"))
# order by time
nmds.tot.df <- with(nmds.tot.df, nmds.tot.df[order(Timepoint),])

# plot without legend.
pNMDS0x <- ggplot(nmds.tot.df, aes(x=NMDS1, y=NMDS2, colour = Diet,
                                   alpha = Timepoint,
                                   group = interaction(Type,ContainerID)))
pNMDS0x <- pNMDS0x +
  geom_hline(yintercept=0, linetype="solid", color="grey40") +
  geom_vline(xintercept=0, linetype="solid", color="grey40") +
```

```

#geom_path(size = .7) +
geom_point(aes(shape = interaction(Diet,Type)), size = 3) +
scale_shape_manual(values = c(15,16,17,0,1,2)) +
scale_color_manual("Feed substrate", values = c("black", "red", "blue")) +
scale_alpha_ordinal(range = c(0.4, 1)) +
labs(x = "NMDS axis 1", y = "NMDS axis 2") +
facet_grid(~Density, labeller = labs_nm) +
theme_nm + theme(legend.position = "none")
pNMDS0x

```



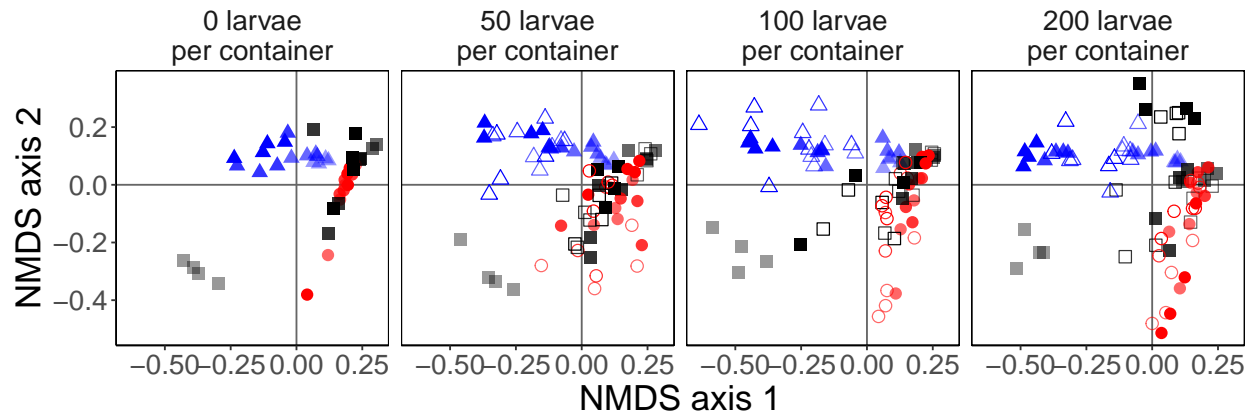
```

# plot with legend.
pNMDS0.3 <- ggplot(nm) + aes(x=N1, y=N2, alpha = Timepoint))
pNMDS0.3 <- pNMDS0.3 +
  geom_point(aes(shape = interaction(Type,Diet)), size = 3) +
  scale_shape_manual("Diet & Sample type", values = c(15,0,16,1,17,2)) +
  scale_alpha_ordinal("Timepoint (d)", range = c(0.4, 1)) +
  guides(alpha = guide_legend(order = 2),
         shape = guide_legend(order = 1)) +
  theme_nm + theme(legend.margin = margin(2,5,0,0))

# plot only legend
nm) <- get_legend(pNMDS0.3)
pNMDS0.leg0x <- as_ggplot(nm)

pNMDS0x; pNMDS0.leg0x

```



Diet & Sample type

- substrate.chicken feed
- larvae.chicken feed
- substrate.camelina
- larvae.camelina
- ▲ substrate.chicken manure
- △ larvae.chicken manure

Timepoint (d)

- 0
- 5
- 10
- 15

3.2. Chicken feed

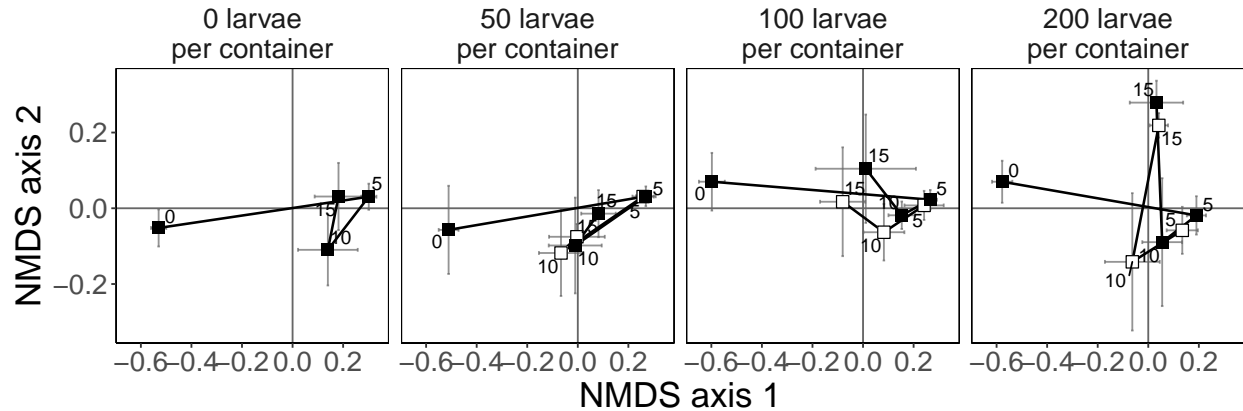
```
# extract plotting data
nm.ds.cf0.df <- plot_ordination(CF, nm.ds.cf0, "samples", axes = 1:2, justDF = T)
# order by time
nm.ds.cf0.df <- with(nm.ds.cf0.df, nm.ds.cf0.df[order(Timepoint),])

# summarize
nm.ds.sum.cf0 <- ddply(nm.ds.cf0.df, .(Density,Timepoint,Type), summarise,
  mean1 = mean(NMDS1), mean2 = mean(NMDS2),
  se1 = se(NMDS1), se2 = se(NMDS2),
  sd1 = sd(NMDS1), sd2 = sd(NMDS2))

# errorbarplot SD
pNMDS.sum.cf0 <- ggplot(nm.ds.sum.cf0, aes(x=mean1, y=mean2,
  group = interaction(Density,Type)))
pNMDS.sum.cf0 <- pNMDS.sum.cf0 +
  geom_hline(yintercept=0, linetype="solid", color="grey40") +
  geom_vline(xintercept=0, linetype="solid", color="grey40") +
  geom_errorbar(aes(ymin=mean2-sd2, ymax=mean2+sd2), width=rel(.01),
    alpha = .7, colour = "grey40") +
  geom_errorbarh(aes(xmin=mean1-sd1, xmax=mean1+sd1), height=rel(.01),
    alpha = .7, colour = "grey40") +
  geom_path(size = .7) +
```

```
geom_point(shape = 15, size = 3, color = "white") +
geom_point(aes(shape = Type), size = 3) +
scale_shape_manual(values = c(15,0)) +
labs(x = "NMDS axis 1", y = "NMDS axis 2") +
geom_text_repel(aes(label = Timepoint), size = 4) +
facet_grid(~Density, labeller = labs_nmDS) +
theme_nmDS + theme(legend.position = "none")
```

pNMDS.sum.cf0



3.3. Camelina

```
# extract plotting data
nmDS.cs0.df <- plot_ordination(CS, nmDS.cs0, "samples", axes = 1:2, justDF = T)
# order by time
nmDS.cs0.df <- with(nmDS.cs0.df, nmDS.cs0.df[order(Timepoint),])

# summarize
nmDS.sum.cs0 <- ddply(nmDS.cs0.df, .(Density, Timepoint, Type), summarise,
  mean1 = mean(NMDS1), mean2 = mean(NMDS2),
  se1 = se(NMDS1), se2 = se(NMDS2),
  sd1 = sd(NMDS1), sd2 = sd(NMDS2))

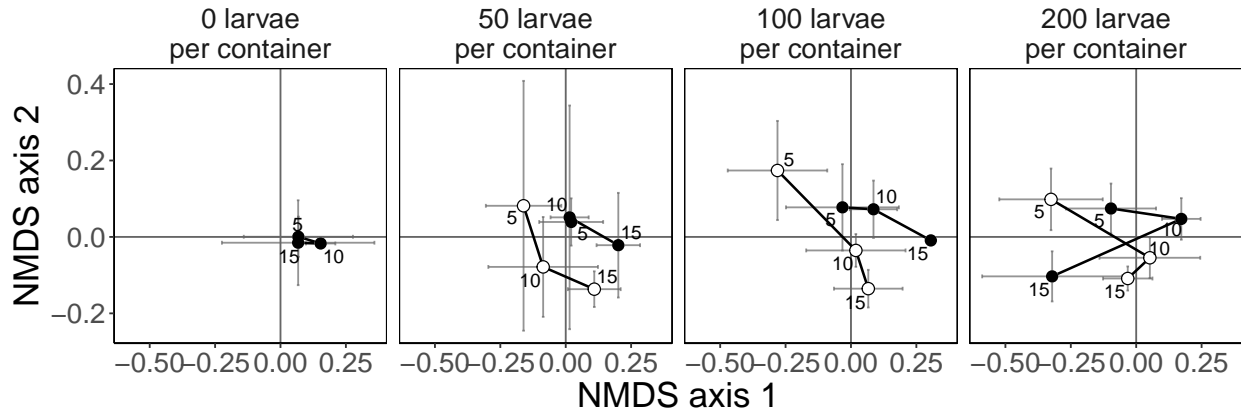
# errorbarplot SD
pNMDS.sum.cs0 <- ggplot(nmDS.sum.cs0, aes(x=mean1, y=mean2,
  group = interaction(Density, Type)))
pNMDS.sum.cs0 <- pNMDS.sum.cs0 +
  geom_hline(yintercept=0, linetype="solid", color="grey40") +
  geom_vline(xintercept=0, linetype="solid", color="grey40") +
  geom_errorbar(aes(ymin=mean2-sd2, ymax=mean2+sd2), width=rel(.01),
    alpha = .7, colour = "grey40") +
  geom_errorbarh(aes(xmin=mean1-sd1, xmax=mean1+sd1), height=rel(.01),
    alpha = .7, colour = "grey40") +
  geom_path(size = .7) +
  geom_point(shape = 16, size = 3, color = "white") +
  geom_point(aes(shape = Type), size = 3) +
  scale_shape_manual(values = c(16,1)) +
```

```

scale_x_continuous(breaks = c(-.5, -.25, 0, .25)) +
labs(x = "NMDS axis 1", y = "NMDS axis 2") +
geom_text_repel(aes(label = Timepoint), size = 4) +
facet_grid(~Density, labeller = labs_nmDS) +
theme_nmDS + theme(legend.position = "none")

```

pNMDS.sum.cm0



3.4. Chicken manure

```

# extract plotting data
nmDS.cm0.df <- plot_ordination(CM, nmDS.cm0, "samples", axes = 1:2, justDF = T)
# order by time
nmDS.cm0.df <- with(nmDS.cm0.df, nmDS.cm0.df[order(Timepoint),])

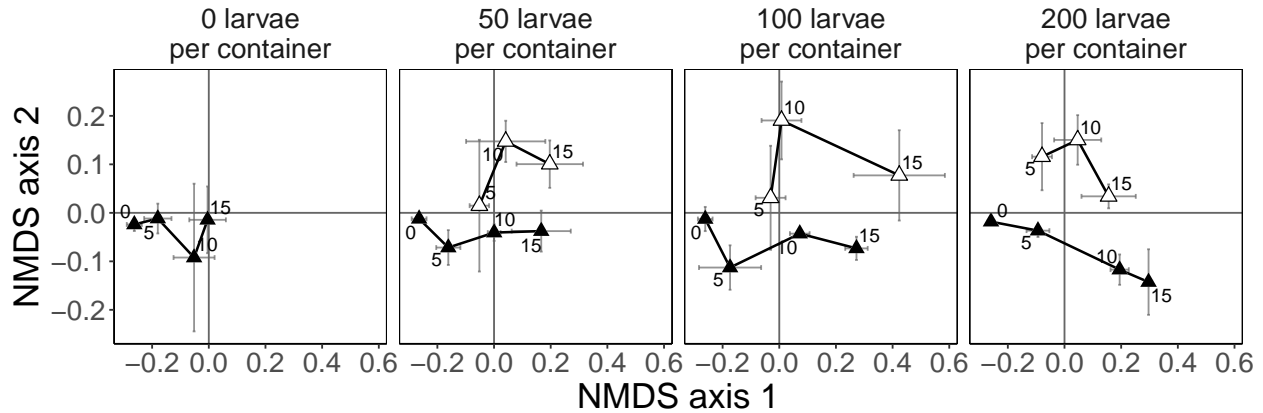
# summarize
nmDS.sum.cm0 <- ddply(nmDS.cm0.df, .(Density,Timepoint,Type), summarise,
  mean1 = mean(NMDS1), mean2 = mean(NMDS2),
  se1 = se(NMDS1), se2 = se(NMDS2),
  sd1 = sd(NMDS1), sd2 = sd(NMDS2))

# errorbarplot SD
pNMDS.sum.cm0 <- ggplot(nmDS.sum.cm0, aes(x=mean1, y=mean2,
  group = interaction(Density,Type)))
pNMDS.sum.cm0 <- pNMDS.sum.cm0 +
  geom_hline(yintercept=0, linetype="solid", color="grey40") +
  geom_vline(xintercept=0, linetype="solid", color="grey40") +
  geom_errorbar(aes(ymin=mean2-sd2, ymax=mean2+sd2), width=rel(.01),
    alpha = .7, colour = "grey40") +
  geom_errorbarh(aes(xmin=mean1-sd1, xmax=mean1+sd1), height=rel(.01),
    alpha = .7, colour = "grey40") +
  geom_path(size = .7) +
  geom_point(shape = 17, size = 3, color = "white") +
  geom_point(aes(shape = Type), size = 3) +
  scale_shape_manual(values = c(17,2)) +
  labs(x = "NMDS axis 1", y = "NMDS axis 2") +
  geom_text_repel(aes(label = Timepoint), size = 4) +

```

```
facet_grid(~Density, labeller = labs_nmds) +
  theme_nmds + theme(legend.position = "none")
```

pNMDS.sum.cm0



4. Export plots

```
# PNG files:
# total: individual points
ggsave(plot = pNMDSOx, "./figures/NMDS_all.png", h = 3.5, w = 10)
ggsave(plot = pNMDS.leg0x, "./figures/NMDS_leg-all.png", h = 4, w = 4)

# per diet: errorbarplots
ggsave(plot = pNMDS.sum.cf0, "./figures/NMDS_CF_sd-all.png", h = 3.5, w = 10)
ggsave(plot = pNMDS.sum.cm0, "./figures/NMDS_CM_sd-all.png", h = 3.5, w = 10)
ggsave(plot = pNMDS.sum.cs0, "./figures/NMDS_CS_sd-all.png", h = 3.5, w = 10)

# PDF files
ggsave(plot = pNMDSOx, "./figures/NMDS_all.pdf", h = 87.5, w = 250, u = "mm")
ggsave(plot = pNMDS.leg0x, "./figures/NMDS_leg-all.pdf", h = 100, w = 100, u = "mm")
ggsave(plot = pNMDS.sum.cf0, "./figures/NMDS_CF_sd-all.pdf", h = 87.5, w = 250, u = "mm")
ggsave(plot = pNMDS.sum.cm0, "./figures/NMDS_CM_sd-all.pdf", h = 87.5, w = 250, u = "mm")
ggsave(plot = pNMDS.sum.cs0, "./figures/NMDS_CS_sd-all.pdf", h = 87.5, w = 250, u = "mm")

# TIFF files: used in AEM submission
ggsave(plot = pNMDSOx, "./figures/NMDS_all.tiff", h = 3.5, w = 10, u = "in", dpi = 600)
ggsave(plot = pNMDS.leg0x, "./figures/NMDS_leg-all.tiff", h = 4, w = 4, u = "in", dpi = 600)
ggsave(plot = pNMDS.sum.cf0, "./figures/NMDS_CF_sd-all.tiff", h = 3.5, w = 10, u = "in", dpi = 600)
ggsave(plot = pNMDS.sum.cm0, "./figures/NMDS_CM_sd-all.tiff", h = 3.5, w = 10, u = "in", dpi = 600)
ggsave(plot = pNMDS.sum.cs0, "./figures/NMDS_CS_sd-all.tiff", h = 3.5, w = 10, u = "in", dpi = 600)
```