

# Beta diversity: unconstrained ordination (NMDS)

Stijn Schreven

1 June 2021

## Contents

<b>Introduction</b>	<b>1</b>
<b>Load packages</b>	<b>2</b>
<b>Input files</b>	<b>2</b>
<b>1. Prepare data</b>	<b>2</b>
1.1. Transform and subsets . . . . .	2
1.2. Distance matrices . . . . .	3
1.3. Plot presets . . . . .	3
<b>2. Substrates</b>	<b>3</b>
2.1. Ordinations . . . . .	3
2.2. Plots . . . . .	19
<b>3. Substrates and larvae</b>	<b>24</b>
3.1. Ordinations . . . . .	24
3.2. Plots . . . . .	38
<b>4. Export plots</b>	<b>44</b>

## 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.

## 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

Using balanced subsets for the NMDS ordinations, because these are required for the PERMANOVA (anova.cca) with restricted permutations by plots = containerID in the constrained ordination dbRDA. In order to match that output, we used the same subsets here in the unconstrained ordinations (although the NMDS could also be run on the complete, unbalanced dataset).

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

# substrates

# total, excl. timepoint 0, excl. larvae
pstot.s <- subset_samples(pstot.g.r, Type == "substrate" & Timepoint != 0)
pstot.s <- prune_taxa(taxa_sums(otu_table(pstot.s)) > 0, pstot.s)

# per diet, incl. timepoint 0 (for CF and CM)
CFs <- subset_samples(pstot.g.r, Diet == "CF" & Type == "substrate")
CFs <- prune_taxa(taxa_sums(otu_table(CFs)) > 0, CFs)
CSs <- subset_samples(pstot.g.r, Diet == "CS" & Type == "substrate")
CSs <- prune_taxa(taxa_sums(otu_table(CSs)) > 0, CSs)
CMs <- subset_samples(pstot.g.r, Diet == "CM" & Type == "substrate")
CMs <- prune_taxa(taxa_sums(otu_table(CMs)) > 0, CMs)

# larvae and substrates (excl. timepoint 0, excl. density 0)

# total
pstot.ls <- subset_samples(pstot.g.r, Timepoint != 0 & Density != 0)
```

```
pstot.ls <- prune_taxa(taxa_sums(otu_table(pstot.ls)) > 0, pstot.ls)

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

## 1.2. Distance matrices

```
# substrates
wuf.sub <- distance(pstot.s, "wunifrac")
wuf.cfs <- distance(CFs, "wunifrac")
wuf.cms <- distance(CMs, "wunifrac")
wuf.css <- distance(CSs, "wunifrac")

# larvae and substrates
wuf.ls <- distance(pstot.ls, "wunifrac")
wuf.cf <- distance(CF.ls, "wunifrac")
wuf.cm <- distance(CM.ls, "wunifrac")
wuf.cs <- distance(CS.ls, "wunifrac")
```

## 1.3. Plot presets

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

## 2. Substrates

Creating the NMDS ordinations for Figure 4 in the manuscript.

### 2.1. Ordinations

### 2.1.1. Total

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

```
## Run 0 stress 0.1180697
## Run 1 stress 0.1568746
## Run 2 stress 0.1303104
## Run 3 stress 0.1489096
## Run 4 stress 0.124252
## Run 5 stress 0.1402495
## Run 6 stress 0.1417659
## Run 7 stress 0.1294245
## Run 8 stress 0.121731
## Run 9 stress 0.14718
## Run 10 stress 0.1409381
## Run 11 stress 0.1369262
## Run 12 stress 0.1315303
## Run 13 stress 0.1520987
## Run 14 stress 0.1647805
## Run 15 stress 0.1729451
## Run 16 stress 0.1162403
## ... New best solution
## ... Procrustes: rmse 0.01888923  max resid 0.1837255
## Run 17 stress 0.1620097
## Run 18 stress 0.1523475
## Run 19 stress 0.1328119
## Run 20 stress 0.1297647
## Run 21 stress 0.1449724
## Run 22 stress 0.1416893
## Run 23 stress 0.157521
## Run 24 stress 0.1349329
## Run 25 stress 0.1431541
## Run 26 stress 0.1264975
## Run 27 stress 0.1282049
## Run 28 stress 0.1507894
## Run 29 stress 0.1465086
## Run 30 stress 0.1685061
## Run 31 stress 0.1567057
## Run 32 stress 0.1645531
## Run 33 stress 0.1451822
## Run 34 stress 0.1301304
## Run 35 stress 0.1230756
## Run 36 stress 0.1753347
## Run 37 stress 0.1616606
## Run 38 stress 0.1251128
## Run 39 stress 0.149638
## Run 40 stress 0.1491089
## Run 41 stress 0.1580758
## Run 42 stress 0.1376263
## Run 43 stress 0.1320628
## Run 44 stress 0.1572501
## Run 45 stress 0.160017
```

```
## Run 46 stress 0.1616655
## Run 47 stress 0.1597836
## Run 48 stress 0.1462999
## Run 49 stress 0.1567357
## Run 50 stress 0.1382701
## Run 51 stress 0.1349634
## Run 52 stress 0.1444791
## Run 53 stress 0.1225262
## Run 54 stress 0.1570561
## Run 55 stress 0.1601141
## Run 56 stress 0.1570033
## Run 57 stress 0.156507
## Run 58 stress 0.1610289
## Run 59 stress 0.1542874
## Run 60 stress 0.1496545
## Run 61 stress 0.1370299
## Run 62 stress 0.1544458
## Run 63 stress 0.159712
## Run 64 stress 0.1324484
## Run 65 stress 0.1368015
## Run 66 stress 0.1325635
## Run 67 stress 0.1544161
## Run 68 stress 0.1696119
## Run 69 stress 0.1585747
## Run 70 stress 0.1334684
## Run 71 stress 0.1592808
## Run 72 stress 0.1242603
## Run 73 stress 0.1292186
## Run 74 stress 0.1205825
## Run 75 stress 0.1264822
## Run 76 stress 0.1315492
## Run 77 stress 0.1626947
## Run 78 stress 0.1324816
## Run 79 stress 0.1433402
## Run 80 stress 0.1896046
## Run 81 stress 0.1841674
## Run 82 stress 0.1593699
## Run 83 stress 0.1420647
## Run 84 stress 0.127214
## Run 85 stress 0.1242307
## Run 86 stress 0.1572998
## Run 87 stress 0.1452281
## Run 88 stress 0.1212953
## Run 89 stress 0.1359451
## Run 90 stress 0.1352967
## Run 91 stress 0.1418383
## Run 92 stress 0.1735743
## Run 93 stress 0.1332049
## Run 94 stress 0.1310684
## Run 95 stress 0.1216716
## Run 96 stress 0.1269248
## Run 97 stress 0.1365887
## Run 98 stress 0.1689319
## Run 99 stress 0.1501899
```

```
## Run 100 stress 0.1403841
## Run 101 stress 0.1369886
## Run 102 stress 0.1479344
## Run 103 stress 0.1159221
## ... New best solution
## ... Procrustes: rmse 0.004501123  max resid 0.04845101
## Run 104 stress 0.14284
## Run 105 stress 0.1347063
## Run 106 stress 0.1398518
## Run 107 stress 0.1508209
## Run 108 stress 0.1494229
## Run 109 stress 0.1364384
## Run 110 stress 0.1336048
## Run 111 stress 0.118289
## Run 112 stress 0.1488659
## Run 113 stress 0.1630319
## Run 114 stress 0.1728886
## Run 115 stress 0.1285906
## Run 116 stress 0.128778
## Run 117 stress 0.1200651
## Run 118 stress 0.1248751
## Run 119 stress 0.1594905
## Run 120 stress 0.1333557
## Run 121 stress 0.1233412
## Run 122 stress 0.1541007
## Run 123 stress 0.1401718
## Run 124 stress 0.1422998
## Run 125 stress 0.1507622
## Run 126 stress 0.1576336
## Run 127 stress 0.1385733
## Run 128 stress 0.1320645
## Run 129 stress 0.14113
## Run 130 stress 0.1480268
## Run 131 stress 0.1564268
## Run 132 stress 0.1413524
## Run 133 stress 0.1355052
## Run 134 stress 0.1646934
## Run 135 stress 0.1349738
## Run 136 stress 0.1575147
## Run 137 stress 0.1194157
## Run 138 stress 0.1298137
## Run 139 stress 0.1476143
## Run 140 stress 0.1502579
## Run 141 stress 0.1309212
## Run 142 stress 0.1430472
## Run 143 stress 0.1665027
## Run 144 stress 0.1705172
## Run 145 stress 0.12406
## Run 146 stress 0.1624839
## Run 147 stress 0.130989
## Run 148 stress 0.1500381
## Run 149 stress 0.1549263
## Run 150 stress 0.13506
## Run 151 stress 0.1357374
```

```

## Run 152 stress 0.1523682
## Run 153 stress 0.1287064
## Run 154 stress 0.1522582
## Run 155 stress 0.1502752
## Run 156 stress 0.1391444
## Run 157 stress 0.1465015
## Run 158 stress 0.1451106
## Run 159 stress 0.1379125
## Run 160 stress 0.126779
## Run 161 stress 0.1225264
## Run 162 stress 0.1834231
## Run 163 stress 0.1426987
## Run 164 stress 0.1436441
## Run 165 stress 0.1522653
## Run 166 stress 0.1577373
## Run 167 stress 0.1487375
## Run 168 stress 0.1509899
## Run 169 stress 0.1549757
## Run 170 stress 0.1443619
## Run 171 stress 0.1279587
## Run 172 stress 0.1909755
## Run 173 stress 0.1519347
## Run 174 stress 0.1332674
## Run 175 stress 0.1243563
## Run 176 stress 0.1304385
## Run 177 stress 0.1674986
## Run 178 stress 0.1399718
## Run 179 stress 0.1251867
## Run 180 stress 0.1275657
## Run 181 stress 0.1503402
## Run 182 stress 0.1725252
## Run 183 stress 0.1550571
## Run 184 stress 0.1189177
## Run 185 stress 0.1285237
## Run 186 stress 0.1455193
## Run 187 stress 0.1246755
## Run 188 stress 0.1527641
## Run 189 stress 0.1288535
## Run 190 stress 0.1448762
## Run 191 stress 0.1573952
## Run 192 stress 0.1386971
## Run 193 stress 0.1706915
## Run 194 stress 0.1524877
## Run 195 stress 0.1357139
## Run 196 stress 0.1282681
## Run 197 stress 0.1546814
## Run 198 stress 0.149198
## Run 199 stress 0.1300483
## Run 200 stress 0.1672303
## *** No convergence -- monoMDS stopping criteria:
##      1: no. of iterations >= maxit
##     194: stress ratio > sratmax
##      5: scale factor of the gradient < sfgrmin

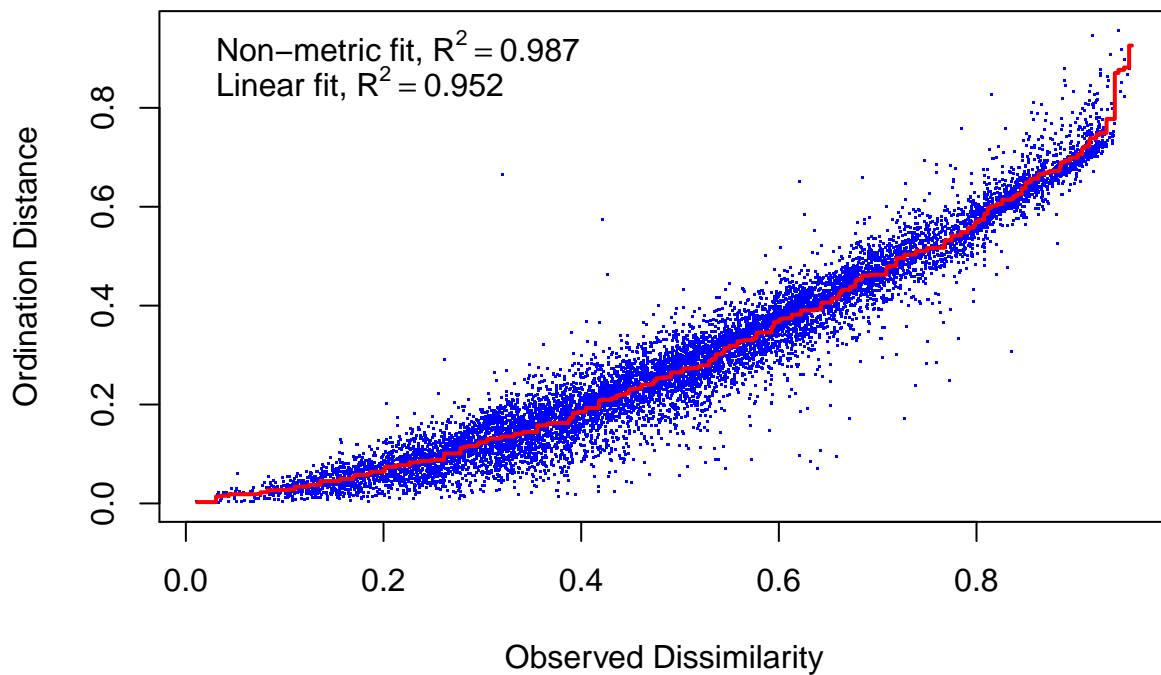
```

```
nm.ds.wuf # no convergent solution after 200 tries, stress = 0.116
```

```
##  
## Call:  
## metaMDS(comm = wuf.sub, k = 2, try = 100, trymax = 200, autotransform = F)  
##  
## global Multidimensional Scaling using monoMDS  
##  
## Data:      wuf.sub  
## Distance: user supplied  
##  
## Dimensions: 2  
## Stress:    0.1159221  
## Stress type 1, weak ties  
## No convergent solutions - best solution after 200 tries  
## Scaling: centring, PC rotation  
## Species: scores missing
```

```
## with try = 200 and trymax = 400, convergent solution of stress = 0.116 after 322 tries
```

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





### 2.1.2. Chicken feed

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

```
## Run 0 stress 0.0701227
## Run 1 stress 0.1027511
## Run 2 stress 0.07014639
## ... Procrustes: rmse 0.00117308  max resid 0.007699023
## ... Similar to previous best
## Run 3 stress 0.1053806
## Run 4 stress 0.07012325
## ... Procrustes: rmse 0.0002818144  max resid 0.001597708
## ... Similar to previous best
## Run 5 stress 0.07014612
## ... Procrustes: rmse 0.00122643  max resid 0.008134482
## ... Similar to previous best
## Run 6 stress 0.07012325
## ... Procrustes: rmse 0.0002836007  max resid 0.001609525
## ... Similar to previous best
## Run 7 stress 0.08348797
## Run 8 stress 0.0701227
## ... New best solution
## ... Procrustes: rmse 6.727154e-06  max resid 3.434e-05
## ... Similar to previous best
## Run 9 stress 0.09999758
## Run 10 stress 0.09505076
## Run 11 stress 0.08273826
## Run 12 stress 0.09845327
## Run 13 stress 0.07012325
## ... Procrustes: rmse 0.0002841154  max resid 0.001605749
## ... Similar to previous best
## Run 14 stress 0.08351977
## Run 15 stress 0.1045434
## Run 16 stress 0.1077771
## Run 17 stress 0.1024334
## Run 18 stress 0.09402394
## Run 19 stress 0.09637961
## Run 20 stress 0.1004558
## Run 21 stress 0.08349683
## Run 22 stress 0.1088282
## Run 23 stress 0.1041095
## Run 24 stress 0.1003838
## Run 25 stress 0.1081225
## Run 26 stress 0.08599451
## Run 27 stress 0.08273863
## Run 28 stress 0.08750849
## Run 29 stress 0.08729879
## Run 30 stress 0.1088665
## Run 31 stress 0.07012259
## ... New best solution
## ... Procrustes: rmse 0.000268023  max resid 0.001529749
## ... Similar to previous best
```

```

## Run 32 stress 0.1027041
## Run 33 stress 0.10137
## Run 34 stress 0.08351958
## Run 35 stress 0.07012315
## ... Procrustes: rmse 0.0002820889 max resid 0.001599841
## ... Similar to previous best
## Run 36 stress 0.09503094
## Run 37 stress 0.1062239
## Run 38 stress 0.108687
## Run 39 stress 0.08273826
## Run 40 stress 0.1033701
## Run 41 stress 0.08978764
## Run 42 stress 0.09580084
## Run 43 stress 0.08280279
## Run 44 stress 0.1039211
## Run 45 stress 0.105063
## Run 46 stress 0.08754165
## Run 47 stress 0.1031581
## Run 48 stress 0.07012325
## ... Procrustes: rmse 0.0003896877 max resid 0.001596973
## ... Similar to previous best
## Run 49 stress 0.08277103
## Run 50 stress 0.08782036
## Run 51 stress 0.1003828
## Run 52 stress 0.0701464
## ... Procrustes: rmse 0.001222584 max resid 0.007827796
## ... Similar to previous best
## Run 53 stress 0.08729897
## Run 54 stress 0.08273912
## Run 55 stress 0.09365868
## Run 56 stress 0.09754542
## Run 57 stress 0.07012325
## ... Procrustes: rmse 0.0003908796 max resid 0.00161134
## ... Similar to previous best
## Run 58 stress 0.08356211
## Run 59 stress 0.1024499
## Run 60 stress 0.09420365
## Run 61 stress 0.09458284
## Run 62 stress 0.1049409
## Run 63 stress 0.0827575
## Run 64 stress 0.09644006
## Run 65 stress 0.09570917
## Run 66 stress 0.09402465
## Run 67 stress 0.09634003
## Run 68 stress 0.08356216
## Run 69 stress 0.08348926
## Run 70 stress 0.07012259
## ... New best solution
## ... Procrustes: rmse 6.473643e-06 max resid 1.854577e-05
## ... Similar to previous best
## Run 71 stress 0.07014617
## ... Procrustes: rmse 0.001181887 max resid 0.007775129
## ... Similar to previous best
## Run 72 stress 0.09644361

```

```

## Run 73 stress 0.1026028
## Run 74 stress 0.1039812
## Run 75 stress 0.105632
## Run 76 stress 0.0701227
## ... Procrustes: rmse 0.0002677393  max resid 0.001524656
## ... Similar to previous best
## Run 77 stress 0.09478227
## Run 78 stress 0.1025892
## Run 79 stress 0.0963953
## Run 80 stress 0.1011581
## Run 81 stress 0.08273836
## Run 82 stress 0.1023489
## Run 83 stress 0.1028651
## Run 84 stress 0.087186
## Run 85 stress 0.1014396
## Run 86 stress 0.1026459
## Run 87 stress 0.1050228
## Run 88 stress 0.1085576
## Run 89 stress 0.1071051
## Run 90 stress 0.09155253
## Run 91 stress 0.08273857
## Run 92 stress 0.1034249
## Run 93 stress 0.1047059
## Run 94 stress 0.08599444
## Run 95 stress 0.1027821
## Run 96 stress 0.07014636
## ... Procrustes: rmse 0.001244694  max resid 0.008221195
## ... Similar to previous best
## Run 97 stress 0.0701227
## ... Procrustes: rmse 0.0002676815  max resid 0.001527115
## ... Similar to previous best
## Run 98 stress 0.1082932
## Run 99 stress 0.106563
## Run 100 stress 0.07555787
## *** Solution reached

```

```

nmds.cfs # convergent solution, stress = 0.070

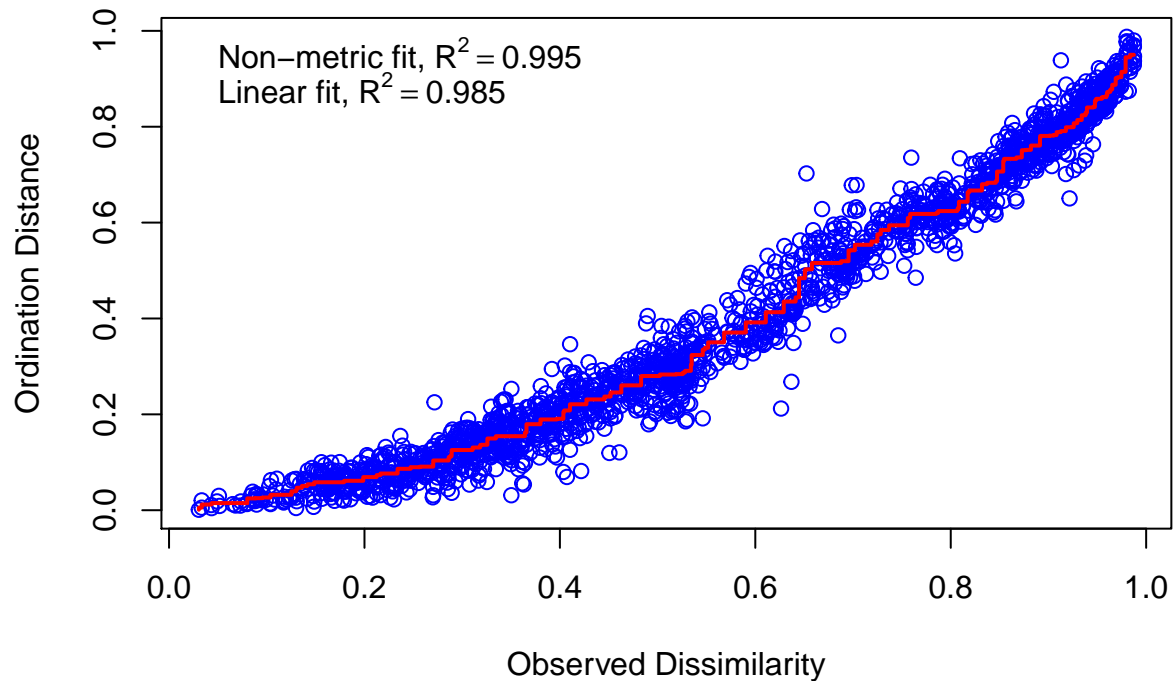
```

```

##
## Call:
## metaMDS(comm = wuf.cfs, k = 2, try = 100, trymax = 200, autotransform = F)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      wuf.cfs
## Distance: user supplied
##
## Dimensions: 2
## Stress:      0.07012259
## Stress type 1, weak ties
## Two convergent solutions found after 100 tries
## Scaling: centring, PC rotation
## Species: scores missing

```

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



### 2.1.3. Camelina

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

```
## Run 0 stress 0.1183744
## Run 1 stress 0.1109437
## ... New best solution
## ... Procrustes: rmse 0.02603273 max resid 0.1571174
## Run 2 stress 0.1191395
## Run 3 stress 0.147648
## Run 4 stress 0.1215535
## Run 5 stress 0.1169571
## Run 6 stress 0.1226685
## Run 7 stress 0.1190191
## Run 8 stress 0.1096839
## ... New best solution
## ... Procrustes: rmse 0.0574084 max resid 0.3667051
## Run 9 stress 0.1186349
```

```

## Run 10 stress 0.1094796
## ... New best solution
## ... Procrustes: rmse 0.01118708  max resid 0.07534826
## Run 11 stress 0.09983427
## ... New best solution
## ... Procrustes: rmse 0.0416571  max resid 0.2696868
## Run 12 stress 0.1109436
## Run 13 stress 0.1162285
## Run 14 stress 0.1094796
## Run 15 stress 0.09983425
## ... New best solution
## ... Procrustes: rmse 3.34712e-05  max resid 0.0001589007
## ... Similar to previous best
## Run 16 stress 0.1169571
## Run 17 stress 0.1516799
## Run 18 stress 0.1070512
## Run 19 stress 0.1109437
## Run 20 stress 0.121977
## Run 21 stress 0.1430277
## Run 22 stress 0.109684
## Run 23 stress 0.1191395
## Run 24 stress 0.1076521
## Run 25 stress 0.117283
## Run 26 stress 0.1109436
## Run 27 stress 0.1172829
## Run 28 stress 0.1172831
## Run 29 stress 0.09976729
## ... New best solution
## ... Procrustes: rmse 0.01036691  max resid 0.06980696
## Run 30 stress 0.1190018
## Run 31 stress 0.1109436
## Run 32 stress 0.1109439
## Run 33 stress 0.1391813
## Run 34 stress 0.1169571
## Run 35 stress 0.09976729
## ... New best solution
## ... Procrustes: rmse 1.391219e-05  max resid 5.156918e-05
## ... Similar to previous best
## Run 36 stress 0.127497
## Run 37 stress 0.107652
## Run 38 stress 0.09976729
## ... New best solution
## ... Procrustes: rmse 1.100717e-05  max resid 6.313738e-05
## ... Similar to previous best
## Run 39 stress 0.09976733
## ... Procrustes: rmse 6.235943e-05  max resid 0.0002855521
## ... Similar to previous best
## Run 40 stress 0.1388963
## Run 41 stress 0.1226685
## Run 42 stress 0.09983425
## ... Procrustes: rmse 0.01036568  max resid 0.06969785
## Run 43 stress 0.1070511
## Run 44 stress 0.1440417
## Run 45 stress 0.09976729

```

```

## ... New best solution
## ... Procrustes: rmse 9.533045e-06  max resid 5.099107e-05
## ... Similar to previous best
## Run 46 stress 0.1076522
## Run 47 stress 0.1169572
## Run 48 stress 0.1169571
## Run 49 stress 0.1190191
## Run 50 stress 0.1162287
## Run 51 stress 0.1117555
## Run 52 stress 0.09983425
## ... Procrustes: rmse 0.01036568  max resid 0.0696978
## Run 53 stress 0.1096838
## Run 54 stress 0.1109436
## Run 55 stress 0.122946
## Run 56 stress 0.09983456
## ... Procrustes: rmse 0.01036892  max resid 0.06970662
## Run 57 stress 0.1596994
## Run 58 stress 0.09976734
## ... Procrustes: rmse 5.980594e-05  max resid 0.0002347846
## ... Similar to previous best
## Run 59 stress 0.1117555
## Run 60 stress 0.1229461
## Run 61 stress 0.1109437
## Run 62 stress 0.1226685
## Run 63 stress 0.1183744
## Run 64 stress 0.1186349
## Run 65 stress 0.09983431
## ... Procrustes: rmse 0.01036421  max resid 0.06969482
## Run 66 stress 0.09976739
## ... Procrustes: rmse 7.281531e-05  max resid 0.0003445085
## ... Similar to previous best
## Run 67 stress 0.1226685
## Run 68 stress 0.1070511
## Run 69 stress 0.1190193
## Run 70 stress 0.1117555
## Run 71 stress 0.1430275
## Run 72 stress 0.121977
## Run 73 stress 0.09976735
## ... Procrustes: rmse 6.3987e-05  max resid 0.0003309939
## ... Similar to previous best
## Run 74 stress 0.1109437
## Run 75 stress 0.09983425
## ... Procrustes: rmse 0.01036315  max resid 0.06969118
## Run 76 stress 0.1094796
## Run 77 stress 0.1430276
## Run 78 stress 0.09983431
## ... Procrustes: rmse 0.01036846  max resid 0.06970439
## Run 79 stress 0.1162288
## Run 80 stress 0.1109437
## Run 81 stress 0.09983427
## ... Procrustes: rmse 0.01036382  max resid 0.06969315
## Run 82 stress 0.122946
## Run 83 stress 0.1190016
## Run 84 stress 0.1109436

```

```

## Run 85 stress 0.09976729
## ... Procrustes: rmse 1.023942e-05  max resid 3.817356e-05
## ... Similar to previous best
## Run 86 stress 0.1109436
## Run 87 stress 0.1172831
## Run 88 stress 0.1186349
## Run 89 stress 0.1190191
## Run 90 stress 0.09983434
## ... Procrustes: rmse 0.01037339  max resid 0.06971737
## Run 91 stress 0.09983425
## ... Procrustes: rmse 0.01036517  max resid 0.06969666
## Run 92 stress 0.09983438
## ... Procrustes: rmse 0.0103765  max resid 0.06972541
## Run 93 stress 0.1187601
## Run 94 stress 0.1094797
## Run 95 stress 0.1070512
## Run 96 stress 0.143243
## Run 97 stress 0.1190194
## Run 98 stress 0.09976729
## ... Procrustes: rmse 1.187267e-05  max resid 6.337226e-05
## ... Similar to previous best
## Run 99 stress 0.1215535
## Run 100 stress 0.1391325
## *** Solution reached

```

```

nm.ds.css # convergent solution, stress = 0.100

```

```

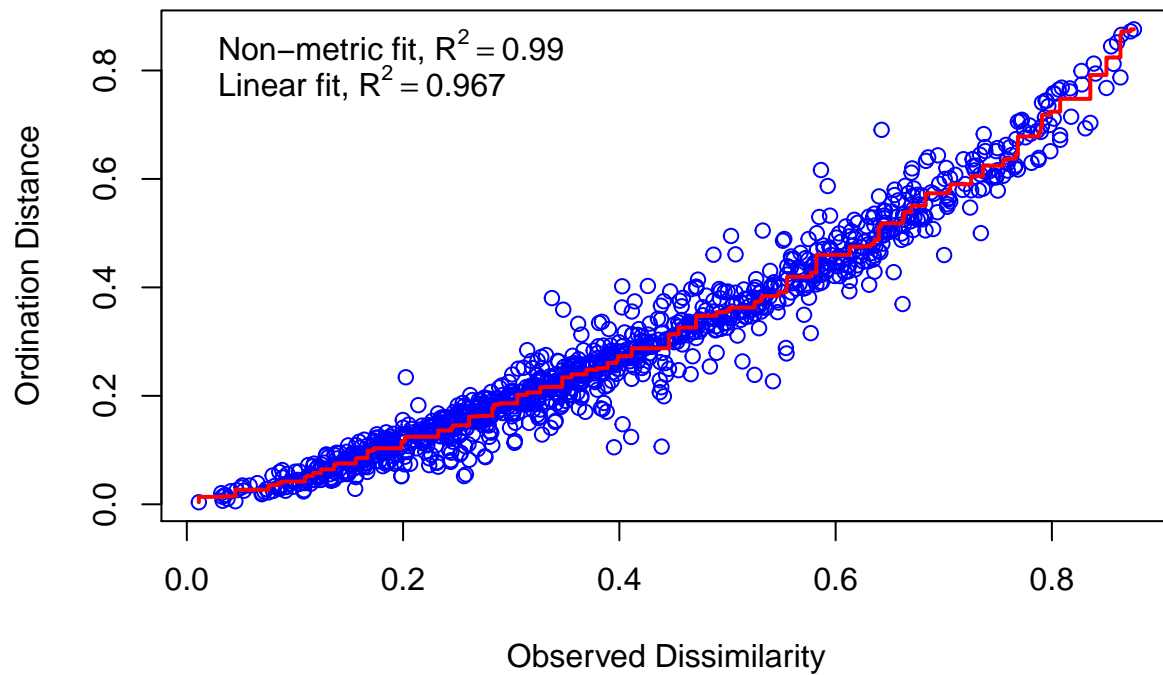
##
## Call:
## metaMDS(comm = wuf.css, k = 2, try = 100, trymax = 200, autotransform = F)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      wuf.css
## Distance: user supplied
##
## Dimensions: 2
## Stress:    0.09976729
## Stress type 1, weak ties
## Two convergent solutions found after 100 tries
## Scaling: centring, PC rotation
## Species: scores missing

```

```

# Shepard stress plot
stressplot(nm.ds.css)

```



#### 2.1.4. Chicken manure

```
# ordination
set.seed(200)
nmfds.cms <- metaMDS(comm = wuf.cms, autotransform = F, k = 2, try = 100, trymax = 200)
```

```
## Run 0 stress 0.07369834
## Run 1 stress 0.08703563
## Run 2 stress 0.08938714
## Run 3 stress 0.09231786
## Run 4 stress 0.07980377
## Run 5 stress 0.08896901
## Run 6 stress 0.1080089
## Run 7 stress 0.08301797
## Run 8 stress 0.1175103
## Run 9 stress 0.08804155
## Run 10 stress 0.09343357
## Run 11 stress 0.07980376
## Run 12 stress 0.1120913
## Run 13 stress 0.1138282
## Run 14 stress 0.0828106
## Run 15 stress 0.1177193
## Run 16 stress 0.08355032
## Run 17 stress 0.1159579
```



```

## Run 18 stress 0.08954607
## Run 19 stress 0.09173847
## Run 20 stress 0.1202028
## Run 21 stress 0.08939152
## Run 22 stress 0.1102395
## Run 23 stress 0.07339893
## ... New best solution
## ... Procrustes: rmse 0.008451895  max resid 0.0652676
## Run 24 stress 0.09219086
## Run 25 stress 0.08119355
## Run 26 stress 0.07369834
## ... Procrustes: rmse 0.008450798  max resid 0.06522884
## Run 27 stress 0.09105983
## Run 28 stress 0.1171407
## Run 29 stress 0.08938731
## Run 30 stress 0.08184314
## Run 31 stress 0.1127085
## Run 32 stress 0.08301797
## Run 33 stress 0.08292029
## Run 34 stress 0.1068334
## Run 35 stress 0.105217
## Run 36 stress 0.08355032
## Run 37 stress 0.1026751
## Run 38 stress 0.115781
## Run 39 stress 0.1073338
## Run 40 stress 0.08292006
## Run 41 stress 0.102003
## Run 42 stress 0.07339987
## ... Procrustes: rmse 0.0006070743  max resid 0.003715349
## ... Similar to previous best
## Run 43 stress 0.07339893
## ... Procrustes: rmse 1.593509e-05  max resid 9.859338e-05
## ... Similar to previous best
## Run 44 stress 0.08119359
## Run 45 stress 0.07980376
## Run 46 stress 0.4067376
## Run 47 stress 0.07369935
## ... Procrustes: rmse 0.00847056  max resid 0.06525965
## Run 48 stress 0.09091767
## Run 49 stress 0.1099082
## Run 50 stress 0.106399
## Run 51 stress 0.07339893
## ... New best solution
## ... Procrustes: rmse 5.248447e-06  max resid 2.676212e-05
## ... Similar to previous best
## Run 52 stress 0.07339893
## ... New best solution
## ... Procrustes: rmse 7.60482e-06  max resid 3.704885e-05
## ... Similar to previous best
## Run 53 stress 0.07980447
## Run 54 stress 0.09114844
## Run 55 stress 0.1071593
## Run 56 stress 0.1070606
## Run 57 stress 0.1114359

```

```

## Run 58 stress 0.1132684
## Run 59 stress 0.083552
## Run 60 stress 0.1083681
## Run 61 stress 0.08119355
## Run 62 stress 0.1146422
## Run 63 stress 0.07980378
## Run 64 stress 0.08938708
## Run 65 stress 0.07339893
## ... Procrustes: rmse 6.399159e-06  max resid 1.513688e-05
## ... Similar to previous best
## Run 66 stress 0.1073337
## Run 67 stress 0.07980376
## Run 68 stress 0.0981572
## Run 69 stress 0.1062566
## Run 70 stress 0.09218582
## Run 71 stress 0.1168999
## Run 72 stress 0.4067374
## Run 73 stress 0.1070646
## Run 74 stress 0.09378718
## Run 75 stress 0.08714284
## Run 76 stress 0.07980385
## Run 77 stress 0.09114844
## Run 78 stress 0.09533624
## Run 79 stress 0.09177891
## Run 80 stress 0.08804173
## Run 81 stress 0.07339893
## ... Procrustes: rmse 4.93126e-06  max resid 1.851456e-05
## ... Similar to previous best
## Run 82 stress 0.1174185
## Run 83 stress 0.09174994
## Run 84 stress 0.1049995
## Run 85 stress 0.07369834
## ... Procrustes: rmse 0.00845129  max resid 0.06523439
## Run 86 stress 0.1107663
## Run 87 stress 0.07369834
## ... Procrustes: rmse 0.008450966  max resid 0.06523091
## Run 88 stress 0.08136019
## Run 89 stress 0.08292006
## Run 90 stress 0.1147686
## Run 91 stress 0.1152095
## Run 92 stress 0.08135913
## Run 93 stress 0.1121475
## Run 94 stress 0.08301796
## Run 95 stress 0.1110581
## Run 96 stress 0.1212112
## Run 97 stress 0.09529459
## Run 98 stress 0.1171369
## Run 99 stress 0.106468
## Run 100 stress 0.1044625
## *** Solution reached

```

```

nmds.cms # convergent solution, stress = 0.073

```

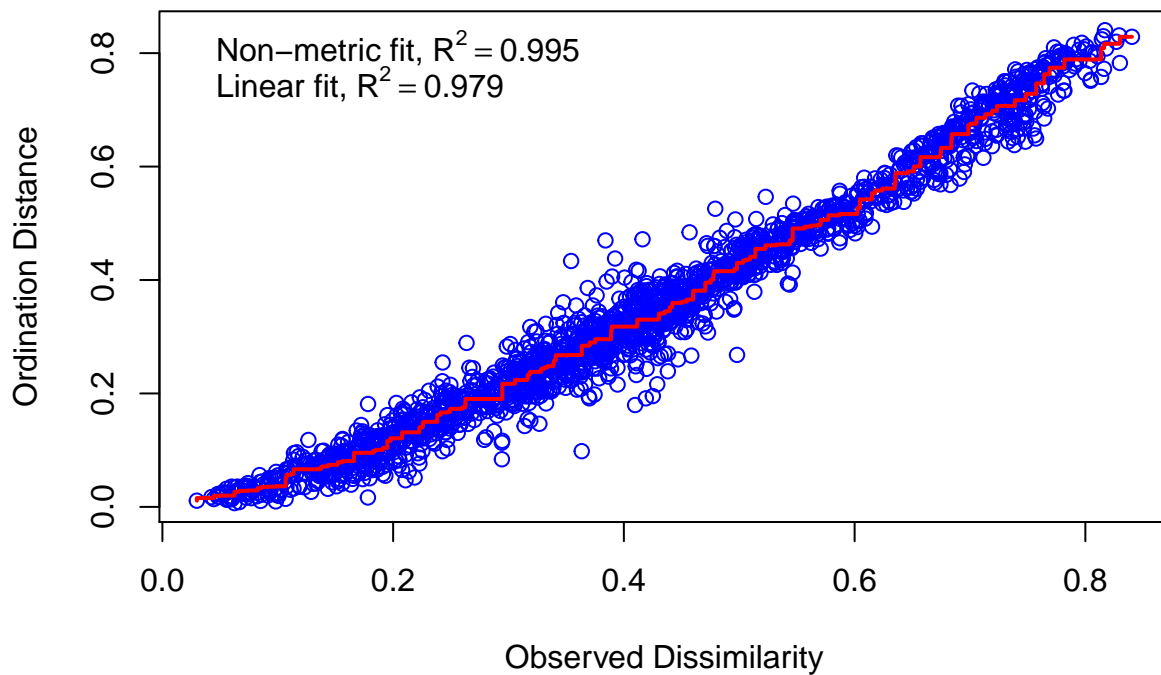
```

##

```

```
## Call:
## metaMDS(comm = wuf.cms, k = 2, try = 100, trymax = 200, autotransform = F)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      wuf.cms
## Distance:  user supplied
##
## Dimensions: 2
## Stress:    0.07339893
## Stress type 1, weak ties
## Two convergent solutions found after 100 tries
## Scaling:  centring, PC rotation
## Species:  scores missing

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



## 2.2. Plots

Figure 4 in manuscript Chapter 3 of PhD thesis.

### 2.2.1. Total

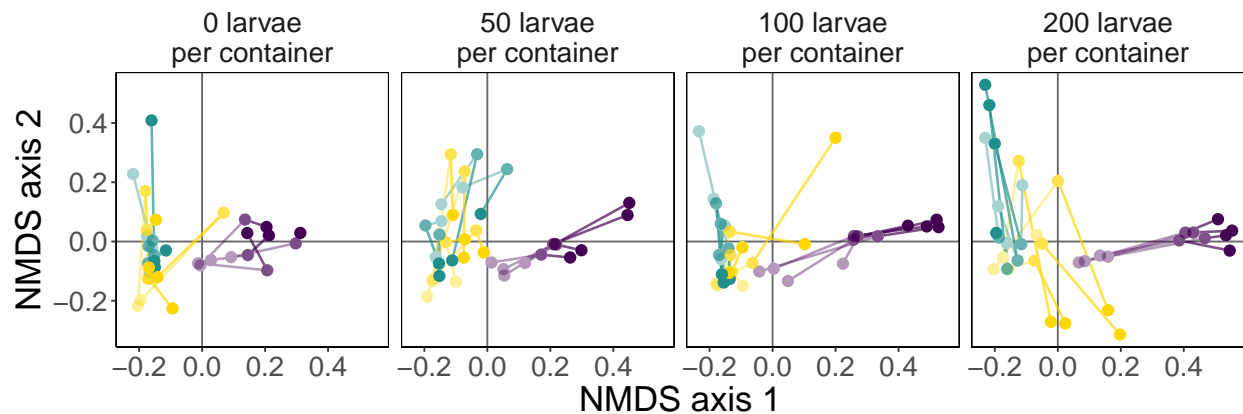
```
# extract plotting data
nmds.df <- plot_ordination(pstot.s, nmds.wuf, "samples", axes = c(1,2), justDF = T)

# rename levels of Diet
nmds.df$Diet <- revalue(nmds.df$Diet, c("CF" = "chicken feed",
                                         "CS" = "camelina", "CM" = "chicken manure"))

# order by time
nmds.df <- with(nmds.df, nmds.df[order(Timepoint),])

# plot without legend.
pNMDS <- ggplot(nmds.df, aes(x = NMDS1, y = NMDS2, colour = Diet,
                             alpha = Timepoint, group = interaction(Type, ContainerID)))
pNMDS <- pNMDS +
  geom_hline(yintercept = 0, linetype = "solid", color = "grey40") +
  geom_vline(xintercept = 0, linetype = "solid", color = "grey40") +
  geom_path(size = .7) +
  geom_point(shape = 16, size = 3, color = "white", alpha = 1) +
  geom_point(shape = 16, size = 3) +
  scale_alpha_ordinal(range = c(0.4, 1)) +
  scale_color_manual(values = c("gold", "#21908CFF", "#440154FF")) +
  labs(x = "NMDS axis 1", y = "NMDS axis 2") +
  facet_grid(~ Density, labeller = labs_nmds) +
  theme_nmds + theme(legend.position = "none")

pNMDS
```



```
# extract colour codes
g <- ggplot_build(pNMDS)
unique(g$data[[3]]["colour"])
```

```
##      colour
## 1      gold
## 5 #440154FF
## 8 #21908CFF
```

```

# viridis colour codes Diets
# CF = #FDE725FF instead used "gold" now.
# CS = #21908CFF
# CM = #440154FF

```

## 2.2.2. Chicken feed

```

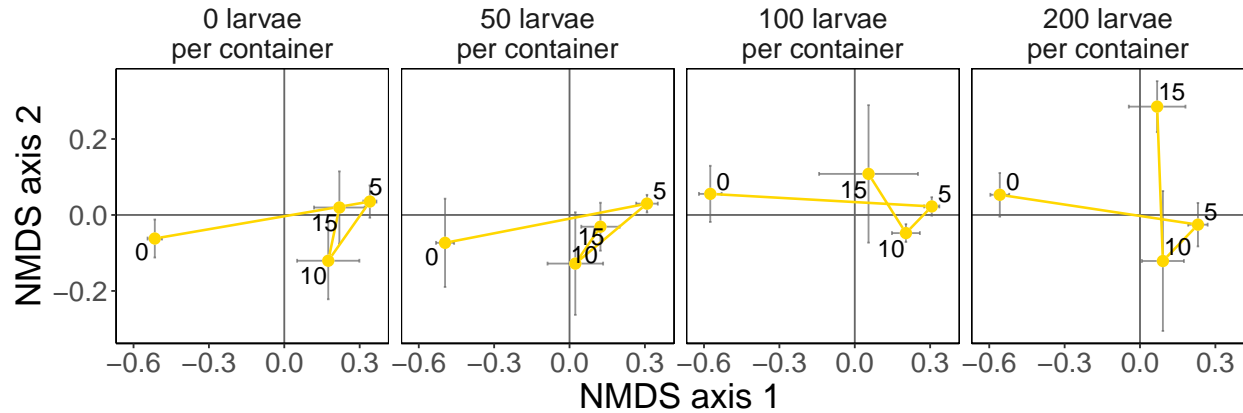
# extract plotting data
nm.ds.cf.df <- plot_ordination(CFs, nm.ds.cfs, "samples", axes = c(1,2), justDF = T)

# order by time
nm.ds.cf.df <- with(nm.ds.cf.df, nm.ds.cf.df[order(Timepoint),])
# summarise scores
nm.ds.cf.sum <- dplyr::ddply(nm.ds.cf.df, .(Density, Timepoint, Type), summarise,
                             mean1 = mean(NMDS1), mean2 = mean(NMDS2),
                             se1 = se(NMDS1), se2 = se(NMDS2),
                             sd1 = sd(NMDS1), sd2 = sd(NMDS2))

# errorbarplot
pNMDS.cf.sum <- ggplot(nm.ds.cf.sum, aes(x = mean1, y = mean2,
                                           group = interaction(Type, Density)))
pNMDS.cf.sum <- pNMDS.cf.sum +
  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, colour = "gold") +
  geom_point(shape = 16, size = 3, colour = "white") +
  geom_point(shape = 16, size = 3, colour = "gold") +
  labs(x = "NMDS axis 1", y = "NMDS axis 2") +
  scale_x_continuous(breaks = c(-.6, -.3, 0, .3)) +
  geom_text_repel(aes(label = Timepoint), colour = "black", size = 5) +
  facet_grid(~ Density, labeller = labs_nm.ds) +
  theme_nm.ds + theme(legend.position = "none")

pNMDS.cf.sum

```



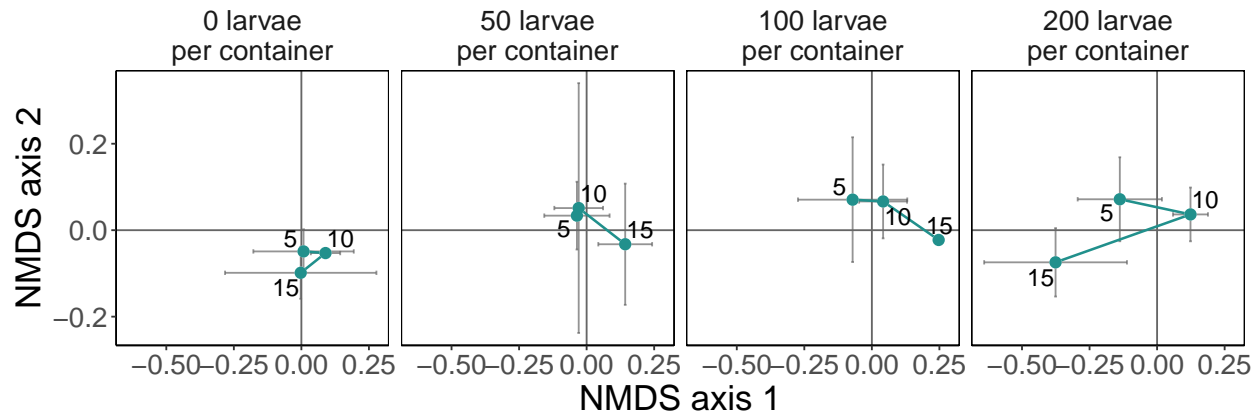
### 2.2.3. Camelina

```
# extract plotting data
nmds.cs.df <- plot_ordination(CSs, nmds.css, "samples", axes = 1:2, justDF = T)

# order by time
nmds.cs.df <- with(nmds.cs.df, nmds.cs.df[order(Timepoint),])
# summarise scores
nmds.cs.sum <- ddply(nmds.cs.df, .(Density, Timepoint, Type), summarise,
  mean1 = mean(NMDS1), mean2 = mean(NMDS2),
  se1 = se(NMDS1), se2 = se(NMDS2),
  sd1 = sd(NMDS1), sd2 = sd(NMDS2))

# errorbarplot, axes 1 and 2
pNMDs.cs.sum <- ggplot(nmds.cs.sum, aes(x=mean1, y=mean2,
  group = interaction(Type, Density)))
pNMDs.cs.sum <- pNMDs.cs.sum +
  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, colour = "#21908CFF") +
  geom_point(shape = 16, colour = "white", size = 3, alpha = 1) +
  geom_point(shape = 16, size=3, colour = "#21908CFF") +
  labs(x = "NMDS axis 1", y = "NMDS axis 2") +
  geom_text_repel(aes(label = Timepoint), colour = "black", size = 5) +
  facet_grid(~Density, labeller = labs_nmds) +
  theme_nmds + theme(legend.position = "none")

pNMDs.cs.sum
```



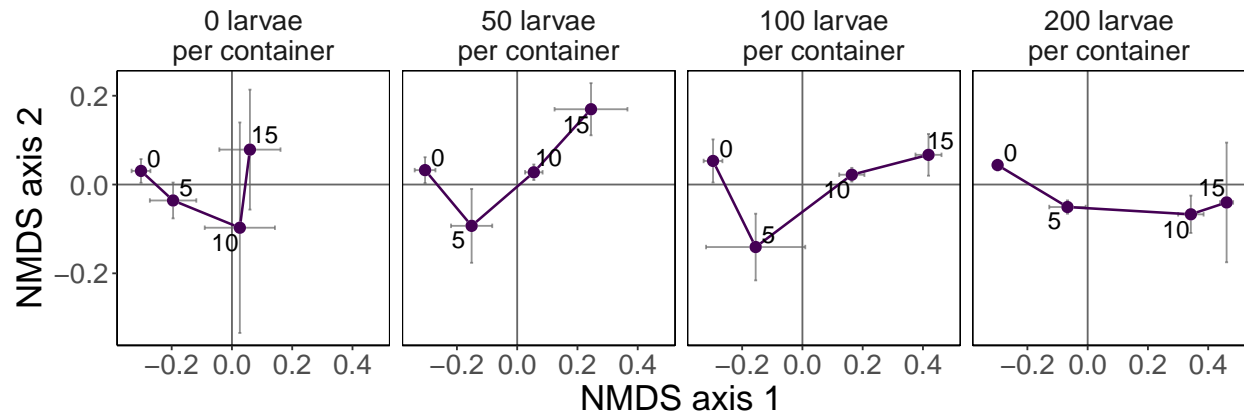
## 2.2.4. Chicken manure

```
# extract plotting data
nm.ds.cm.df <- plot_ordination(CMs, nm.ds.cms, "samples", axes = 1:2, justDF = T)

# order by time
nm.ds.cm.df <- with(nm.ds.cm.df, nm.ds.cm.df[order(Timepoint),])
# summarise scores
nm.ds.cm.sum <- ddply(nm.ds.cm.df, .(Density, Timepoint, Type), summarise,
  mean1 = mean(NMDS1), mean2 = mean(NMDS2),
  se1 = se(NMDS1), se2 = se(NMDS2),
  sd1 = sd(NMDS1), sd2 = sd(NMDS2))

# errorbarplot
pNMDS.cm.sum <- ggplot(nm.ds.cm.sum, aes(x=mean1, y=mean2,
  group = interaction(Type, Density)))
pNMDS.cm.sum <- pNMDS.cm.sum +
  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, colour = "#440154FF") +
  geom_point(shape = 16, colour = "white", size = 3, alpha = 1) +
  geom_point(shape = 16, size=3, colour = "#440154FF") +
  labs(x = "NMDS axis 1", y = "NMDS axis 2") +
  geom_text_repel(aes(label = Timepoint), colour = "black", size = 5) +
  facet_grid(~Density, labeller = labs_nm.ds) +
  theme_nm.ds + theme(legend.position = "none")

pNMDS.cm.sum
```



### 3. Substrates and larvae

Creating the NMDS ordinations for Figure 6 in the manuscript.

#### 3.1. Ordinations

##### 3.1.1. Total

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

```
## Run 0 stress 0.1314485
## Run 1 stress 0.1449012
## Run 2 stress 0.1314487
## ... Procrustes: rmse 0.0002600501 max resid 0.003320581
## ... Similar to previous best
## Run 3 stress 0.1424914
## Run 4 stress 0.1457622
## Run 5 stress 0.1386768
## Run 6 stress 0.1384643
## Run 7 stress 0.1439083
## Run 8 stress 0.141846
## Run 9 stress 0.1345629
## Run 10 stress 0.1347634
## Run 11 stress 0.1413109
## Run 12 stress 0.1549656
## Run 13 stress 0.1500855
## Run 14 stress 0.1352499
## Run 15 stress 0.1417823
## Run 16 stress 0.1376084
## Run 17 stress 0.1417999
## Run 18 stress 0.134918
## Run 19 stress 0.1459344
## Run 20 stress 0.1469849
## Run 21 stress 0.1453649
```



```
## Run 22 stress 0.1462015
## Run 23 stress 0.1508197
## Run 24 stress 0.1372155
## Run 25 stress 0.1457807
## Run 26 stress 0.1314972
## ... Procrustes: rmse 0.006999026 max resid 0.0632989
## Run 27 stress 0.1315038
## ... Procrustes: rmse 0.006847838 max resid 0.06322839
## Run 28 stress 0.1417222
## Run 29 stress 0.1377276
## Run 30 stress 0.1434827
## Run 31 stress 0.1470047
## Run 32 stress 0.1352098
## Run 33 stress 0.1316676
## ... Procrustes: rmse 0.003093247 max resid 0.04405345
## Run 34 stress 0.1414178
## Run 35 stress 0.1520391
## Run 36 stress 0.1615762
## Run 37 stress 0.1448244
## Run 38 stress 0.416972
## Run 39 stress 0.1730993
## Run 40 stress 0.1385985
## Run 41 stress 0.1386852
## Run 42 stress 0.137954
## Run 43 stress 0.1417258
## Run 44 stress 0.1315726
## ... Procrustes: rmse 0.005652798 max resid 0.06189077
## Run 45 stress 0.144965
## Run 46 stress 0.1351454
## Run 47 stress 0.1349933
## Run 48 stress 0.1386021
## Run 49 stress 0.1345957
## Run 50 stress 0.1441536
## Run 51 stress 0.1418068
## Run 52 stress 0.1374951
## Run 53 stress 0.1341785
## Run 54 stress 0.1347624
## Run 55 stress 0.1426973
## Run 56 stress 0.1485809
## Run 57 stress 0.1394825
## Run 58 stress 0.1414207
## Run 59 stress 0.1431297
## Run 60 stress 0.1416106
## Run 61 stress 0.1430518
## Run 62 stress 0.1439079
## Run 63 stress 0.1471176
## Run 64 stress 0.1315624
## ... Procrustes: rmse 0.005818309 max resid 0.06201689
## Run 65 stress 0.1395418
## Run 66 stress 0.1375505
## Run 67 stress 0.152434
## Run 68 stress 0.137608
## Run 69 stress 0.142831
## Run 70 stress 0.1422639
```

```

## Run 71 stress 0.1421982
## Run 72 stress 0.1439079
## Run 73 stress 0.1318763
## ... Procrustes: rmse 0.008277259  max resid 0.06404493
## Run 74 stress 0.1388917
## Run 75 stress 0.1373396
## Run 76 stress 0.1371959
## Run 77 stress 0.1485942
## Run 78 stress 0.1385333
## Run 79 stress 0.1315006
## ... Procrustes: rmse 0.006803653  max resid 0.06313996
## Run 80 stress 0.138691
## Run 81 stress 0.1496473
## Run 82 stress 0.1379092
## Run 83 stress 0.1314511
## ... Procrustes: rmse 0.001472838  max resid 0.02012191
## Run 84 stress 0.1348778
## Run 85 stress 0.1345575
## Run 86 stress 0.1395421
## Run 87 stress 0.1500893
## Run 88 stress 0.1470508
## Run 89 stress 0.1418386
## Run 90 stress 0.1376072
## Run 91 stress 0.1316246
## ... Procrustes: rmse 0.00213928  max resid 0.03041737
## Run 92 stress 0.1459671
## Run 93 stress 0.1415765
## Run 94 stress 0.1476636
## Run 95 stress 0.1416324
## Run 96 stress 0.1413769
## Run 97 stress 0.1381511
## Run 98 stress 0.1375781
## Run 99 stress 0.1416111
## Run 100 stress 0.1390084
## *** Solution reached

```

```

nmds.ls # convergent solution, stress = 0.131

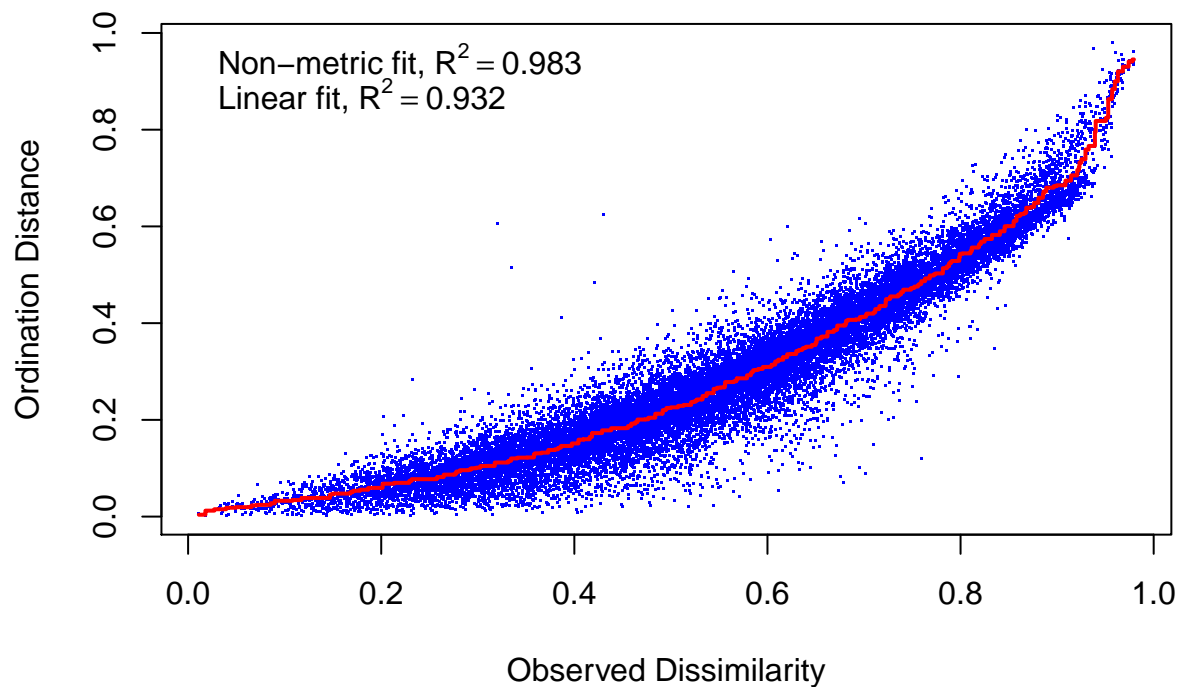
```

```

##
## Call:
## metaMDS(comm = wuf.ls, k = 2, try = 100, trymax = 200, autotransform = F)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      wuf.ls
## Distance: user supplied
##
## Dimensions: 2
## Stress:     0.1314485
## Stress type 1, weak ties
## Two convergent solutions found after 100 tries
## Scaling: centring, PC rotation
## Species: scores missing

```

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



### 3.1.2. Chicken feed

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

```
## Run 0 stress 0.1253233
## Run 1 stress 0.1867034
## Run 2 stress 0.2152776
## Run 3 stress 0.2103564
## Run 4 stress 0.1247862
## ... New best solution
## ... Procrustes: rmse 0.01251463 max resid 0.08855014
## Run 5 stress 0.2180703
## Run 6 stress 0.2138255
## Run 7 stress 0.1252814
## ... Procrustes: rmse 0.01123373 max resid 0.08613365
## Run 8 stress 0.1253223
## Run 9 stress 0.1521754
## Run 10 stress 0.1252857
```

```

## ... Procrustes: rmse 0.01142829  max resid 0.08678793
## Run 11 stress 0.1800028
## Run 12 stress 0.2006823
## Run 13 stress 0.1790316
## Run 14 stress 0.1247876
## ... Procrustes: rmse 0.002172364  max resid 0.01313593
## Run 15 stress 0.1247861
## ... New best solution
## ... Procrustes: rmse 0.0001610908  max resid 0.001226484
## ... Similar to previous best
## Run 16 stress 0.125282
## ... Procrustes: rmse 0.01129299  max resid 0.08616494
## Run 17 stress 0.1252795
## ... Procrustes: rmse 0.01129224  max resid 0.08694833
## Run 18 stress 0.2161542
## Run 19 stress 0.2049535
## Run 20 stress 0.1525126
## Run 21 stress 0.125281
## ... Procrustes: rmse 0.01133797  max resid 0.08707125
## Run 22 stress 0.1722989
## Run 23 stress 0.1252759
## ... Procrustes: rmse 0.01115571  max resid 0.0865333
## Run 24 stress 0.1253342
## Run 25 stress 0.2111306
## Run 26 stress 0.1517929
## Run 27 stress 0.1969971
## Run 28 stress 0.1256231
## Run 29 stress 0.1929524
## Run 30 stress 0.1256375
## Run 31 stress 0.2247325
## Run 32 stress 0.1248139
## ... Procrustes: rmse 0.002533209  max resid 0.01319465
## Run 33 stress 0.1256184
## Run 34 stress 0.1521754
## Run 35 stress 0.1622596
## Run 36 stress 0.1252838
## ... Procrustes: rmse 0.01138927  max resid 0.0864883
## Run 37 stress 0.2288184
## Run 38 stress 0.4077202
## Run 39 stress 0.125274
## ... Procrustes: rmse 0.01103307  max resid 0.08596085
## Run 40 stress 0.1252761
## ... Procrustes: rmse 0.01116482  max resid 0.086569
## Run 41 stress 0.2054591
## Run 42 stress 0.12528
## ... Procrustes: rmse 0.0109832  max resid 0.08489676
## Run 43 stress 0.1518652
## Run 44 stress 0.1956555
## Run 45 stress 0.1248138
## ... Procrustes: rmse 0.00251783  max resid 0.0131992
## Run 46 stress 0.1604299
## Run 47 stress 0.204246
## Run 48 stress 0.2186735
## Run 49 stress 0.2148586

```

```

## Run 50 stress 0.210183
## Run 51 stress 0.2088337
## Run 52 stress 0.1248137
## ... Procrustes: rmse 0.002518193  max resid 0.01319918
## Run 53 stress 0.1626796
## Run 54 stress 0.1248282
## ... Procrustes: rmse 0.001956977  max resid 0.01222985
## Run 55 stress 0.1252727
## ... Procrustes: rmse 0.01094304  max resid 0.08573019
## Run 56 stress 0.165226
## Run 57 stress 0.1256238
## Run 58 stress 0.1593364
## Run 59 stress 0.1252799
## ... Procrustes: rmse 0.01096406  max resid 0.08470849
## Run 60 stress 0.1758511
## Run 61 stress 0.1855064
## Run 62 stress 0.1256378
## Run 63 stress 0.1253388
## Run 64 stress 0.1248137
## ... Procrustes: rmse 0.002519872  max resid 0.0131987
## Run 65 stress 0.2050742
## Run 66 stress 0.1619243
## Run 67 stress 0.1248422
## ... Procrustes: rmse 0.002727829  max resid 0.01316975
## Run 68 stress 0.1248427
## ... Procrustes: rmse 0.002737263  max resid 0.01314734
## Run 69 stress 0.2522437
## Run 70 stress 0.1252845
## ... Procrustes: rmse 0.01141809  max resid 0.08657927
## Run 71 stress 0.1256369
## Run 72 stress 0.1940927
## Run 73 stress 0.1247874
## ... Procrustes: rmse 0.00218208  max resid 0.01312273
## Run 74 stress 0.2116577
## Run 75 stress 0.1252849
## ... Procrustes: rmse 0.01129965  max resid 0.08619528
## Run 76 stress 0.1860618
## Run 77 stress 0.1253194
## Run 78 stress 0.1247873
## ... Procrustes: rmse 0.002174966  max resid 0.01313378
## Run 79 stress 0.1252799
## ... Procrustes: rmse 0.01100824  max resid 0.08502192
## Run 80 stress 0.18421
## Run 81 stress 0.1247978
## ... Procrustes: rmse 0.0008707054  max resid 0.006078773
## ... Similar to previous best
## Run 82 stress 0.2092518
## Run 83 stress 0.1252798
## ... Procrustes: rmse 0.01100541  max resid 0.08492602
## Run 84 stress 0.1248421
## ... Procrustes: rmse 0.002711921  max resid 0.01316951
## Run 85 stress 0.2165778
## Run 86 stress 0.1618867
## Run 87 stress 0.1247861

```

```

## ... New best solution
## ... Procrustes: rmse 9.58362e-05  max resid 0.0006874437
## ... Similar to previous best
## Run 88 stress 0.1252743
## ... Procrustes: rmse 0.0110556  max resid 0.08628512
## Run 89 stress 0.2100236
## Run 90 stress 0.1252798
## ... Procrustes: rmse 0.01097459  max resid 0.08489648
## Run 91 stress 0.1248137
## ... Procrustes: rmse 0.002520711  max resid 0.01320879
## Run 92 stress 0.1248422
## ... Procrustes: rmse 0.002723509  max resid 0.01317339
## Run 93 stress 0.1518069
## Run 94 stress 0.2080201
## Run 95 stress 0.2105623
## Run 96 stress 0.124814
## ... Procrustes: rmse 0.002528128  max resid 0.0131871
## Run 97 stress 0.1980168
## Run 98 stress 0.1252797
## ... Procrustes: rmse 0.01103106  max resid 0.08517545
## Run 99 stress 0.1247862
## ... Procrustes: rmse 3.198498e-05  max resid 0.0001802555
## ... Similar to previous best
## Run 100 stress 0.1774763
## *** Solution reached

```

```

nm.ds.cf # convergent solution, stress = 0.125

```

```

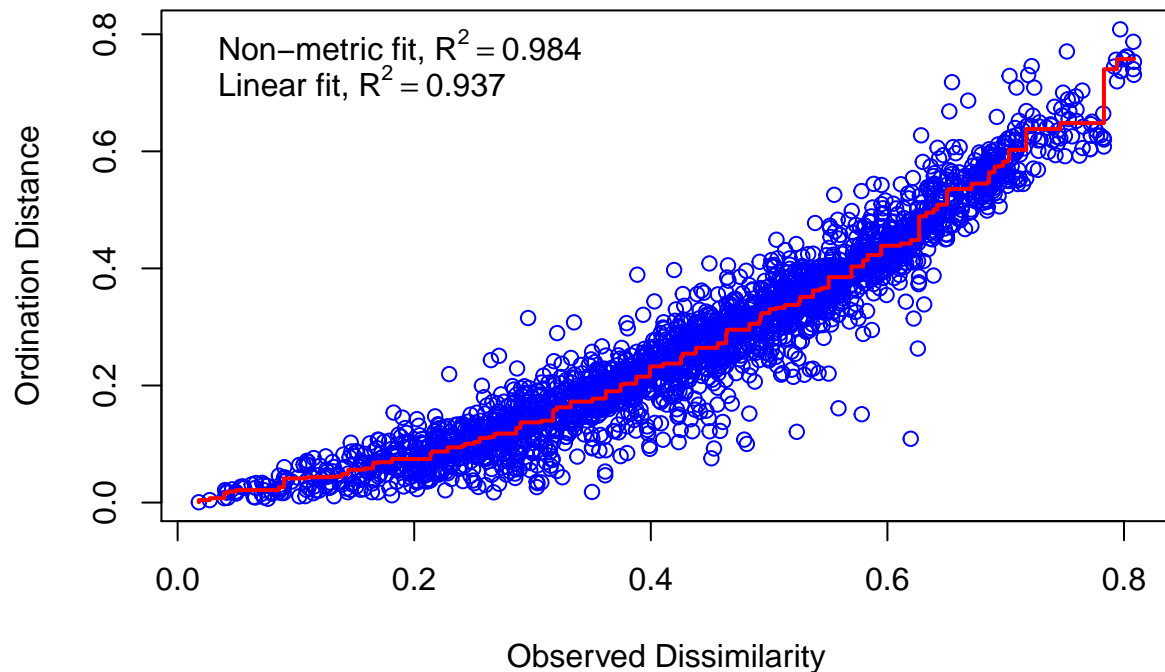
##
## Call:
## metaMDS(comm = wuf.cf, k = 2, try = 100, trymax = 200, autotransform = F)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      wuf.cf
## Distance: user supplied
##
## Dimensions: 2
## Stress:    0.1247861
## Stress type 1, weak ties
## Two convergent solutions found after 100 tries
## Scaling: centring, PC rotation
## Species: scores missing

```

```

# Shepard stress plot
stressplot(nm.ds.cf)

```



### 3.1.3. Camelina

```
# ordination
set.seed(200)
nmfs.cs <- metaMDS(comm = wuf.cs, autotransform = F, k = 2, try = 100, trymax = 200)
```

```
## Run 0 stress 0.1327104
## Run 1 stress 0.1234846
## ... New best solution
## ... Procrustes: rmse 0.03675515 max resid 0.2996029
## Run 2 stress 0.1658518
## Run 3 stress 0.1648589
## Run 4 stress 0.1234811
## ... New best solution
## ... Procrustes: rmse 0.001276087 max resid 0.008067139
## ... Similar to previous best
## Run 5 stress 0.1366957
## Run 6 stress 0.1706722
## Run 7 stress 0.162975
## Run 8 stress 0.1737215
## Run 9 stress 0.1305797
## Run 10 stress 0.1234811
## ... Procrustes: rmse 4.946273e-06 max resid 2.110834e-05
## ... Similar to previous best
```

```

## Run 11 stress 0.1234846
## ... Procrustes: rmse 0.001276439  max resid 0.008069844
## ... Similar to previous best
## Run 12 stress 0.1234811
## ... New best solution
## ... Procrustes: rmse 2.835934e-06  max resid 1.30728e-05
## ... Similar to previous best
## Run 13 stress 0.1480773
## Run 14 stress 0.1661672
## Run 15 stress 0.1234846
## ... Procrustes: rmse 0.001275635  max resid 0.008064057
## ... Similar to previous best
## Run 16 stress 0.1234846
## ... Procrustes: rmse 0.001276281  max resid 0.00806833
## ... Similar to previous best
## Run 17 stress 0.1629474
## Run 18 stress 0.1234811
## ... Procrustes: rmse 5.394668e-06  max resid 1.967703e-05
## ... Similar to previous best
## Run 19 stress 0.1477192
## Run 20 stress 0.1661758
## Run 21 stress 0.1740358
## Run 22 stress 0.1295499
## Run 23 stress 0.1305615
## Run 24 stress 0.1442618
## Run 25 stress 0.1358502
## Run 26 stress 0.1234811
## ... Procrustes: rmse 3.210332e-06  max resid 1.30947e-05
## ... Similar to previous best
## Run 27 stress 0.1694743
## Run 28 stress 0.1234846
## ... Procrustes: rmse 0.001276484  max resid 0.008068079
## ... Similar to previous best
## Run 29 stress 0.1474486
## Run 30 stress 0.145326
## Run 31 stress 0.1487436
## Run 32 stress 0.1463465
## Run 33 stress 0.1974966
## Run 34 stress 0.1360609
## Run 35 stress 0.1295544
## Run 36 stress 0.1234846
## ... Procrustes: rmse 0.001276372  max resid 0.008068728
## ... Similar to previous best
## Run 37 stress 0.1463465
## Run 38 stress 0.1683359
## Run 39 stress 0.1358502
## Run 40 stress 0.1305797
## Run 41 stress 0.1234846
## ... Procrustes: rmse 0.001275643  max resid 0.008066167
## ... Similar to previous best
## Run 42 stress 0.1234811
## ... New best solution
## ... Procrustes: rmse 2.719114e-06  max resid 1.622039e-05
## ... Similar to previous best

```



```

## Run 43 stress 0.1406517
## Run 44 stress 0.1438012
## Run 45 stress 0.1295544
## Run 46 stress 0.1975618
## Run 47 stress 0.1295544
## Run 48 stress 0.1234846
## ... Procrustes: rmse 0.001276798  max resid 0.008073267
## ... Similar to previous best
## Run 49 stress 0.1234811
## ... Procrustes: rmse 5.06887e-06  max resid 2.228674e-05
## ... Similar to previous best
## Run 50 stress 0.1683363
## Run 51 stress 0.1687633
## Run 52 stress 0.1234846
## ... Procrustes: rmse 0.001276328  max resid 0.008069651
## ... Similar to previous best
## Run 53 stress 0.1406784
## Run 54 stress 0.1697631
## Run 55 stress 0.1460861
## Run 56 stress 0.1442618
## Run 57 stress 0.1305615
## Run 58 stress 0.1234846
## ... Procrustes: rmse 0.001276186  max resid 0.008068671
## ... Similar to previous best
## Run 59 stress 0.1791827
## Run 60 stress 0.1619877
## Run 61 stress 0.1360609
## Run 62 stress 0.1295499
## Run 63 stress 0.1614545
## Run 64 stress 0.1234811
## ... Procrustes: rmse 7.591725e-06  max resid 2.845234e-05
## ... Similar to previous best
## Run 65 stress 0.1234846
## ... Procrustes: rmse 0.001277028  max resid 0.008072575
## ... Similar to previous best
## Run 66 stress 0.1629509
## Run 67 stress 0.1327177
## Run 68 stress 0.1463465
## Run 69 stress 0.1673157
## Run 70 stress 0.137929
## Run 71 stress 0.1700691
## Run 72 stress 0.1687633
## Run 73 stress 0.1923113
## Run 74 stress 0.1438036
## Run 75 stress 0.1614833
## Run 76 stress 0.1327104
## Run 77 stress 0.1234811
## ... Procrustes: rmse 7.943986e-06  max resid 3.331675e-05
## ... Similar to previous best
## Run 78 stress 0.1673157
## Run 79 stress 0.1919346
## Run 80 stress 0.1463688
## Run 81 stress 0.1477959
## Run 82 stress 0.1305717

```

```

## Run 83 stress 0.1619655
## Run 84 stress 0.1295544
## Run 85 stress 0.1234846
## ... Procrustes: rmse 0.00127732  max resid 0.008075438
## ... Similar to previous best
## Run 86 stress 0.1305615
## Run 87 stress 0.1658518
## Run 88 stress 0.1492119
## Run 89 stress 0.1493377
## Run 90 stress 0.1295544
## Run 91 stress 0.1661672
## Run 92 stress 0.1474813
## Run 93 stress 0.1234846
## ... Procrustes: rmse 0.001276456  max resid 0.008071621
## ... Similar to previous best
## Run 94 stress 0.1480773
## Run 95 stress 0.1360609
## Run 96 stress 0.1480773
## Run 97 stress 0.1614545
## Run 98 stress 0.1461141
## Run 99 stress 0.1731816
## Run 100 stress 0.1360609
## *** Solution reached

```

```

nm.ds.cs # convergent solution, stress = 0.123

```

```

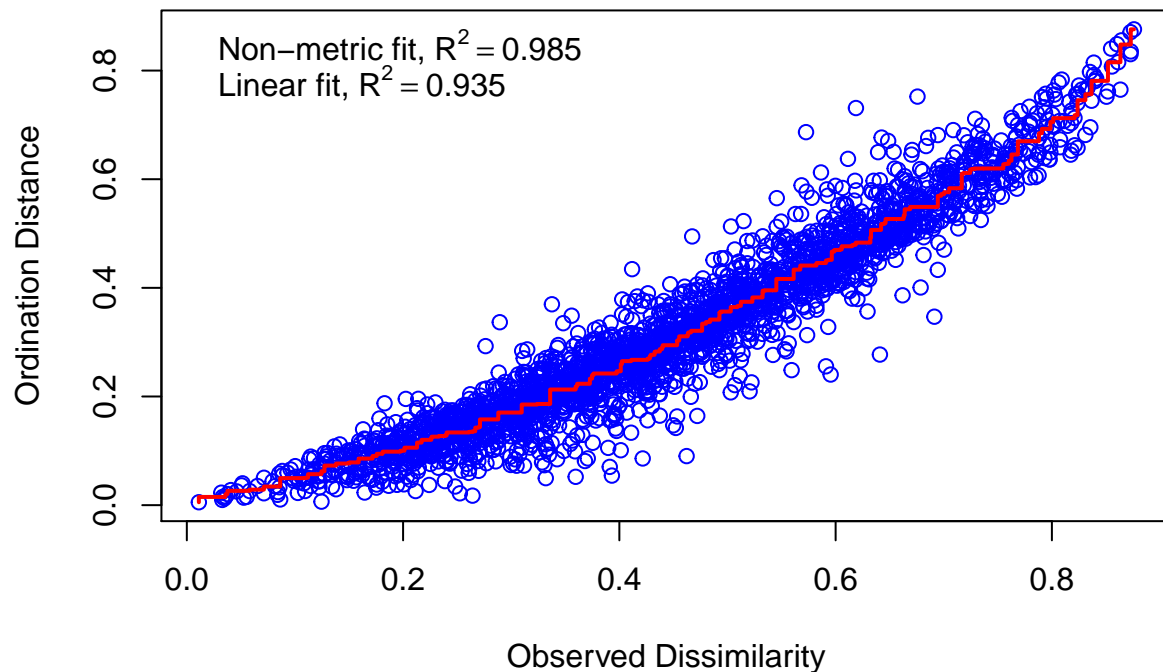
##
## Call:
## metaMDS(comm = wuf.cs, k = 2, try = 100, trymax = 200, autotransform = F)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      wuf.cs
## Distance: user supplied
##
## Dimensions: 2
## Stress:      0.1234811
## Stress type 1, weak ties
## Two convergent solutions found after 100 tries
## Scaling: centring, PC rotation
## Species: scores missing

```

```

# Shepard stress plot
stressplot(nm.ds.cs)

```



#### 3.1.4. Chicken manure

```
# ordination
set.seed(200)
nmfds.cm <- metaMDS(comm = wuf.cm, autotransform = F, k = 2, try = 100, trymax = 200)
```

```
## Run 0 stress 0.1425174
## Run 1 stress 0.1438113
## Run 2 stress 0.1440782
## Run 3 stress 0.1422133
## ... New best solution
## ... Procrustes: rmse 0.00654926 max resid 0.04068727
## Run 4 stress 0.1441218
## Run 5 stress 0.1422596
## ... Procrustes: rmse 0.004952539 max resid 0.04005622
## Run 6 stress 0.1441226
## Run 7 stress 0.1642023
## Run 8 stress 0.1464668
## Run 9 stress 0.1776937
## Run 10 stress 0.1625709
## Run 11 stress 0.1440896
## Run 12 stress 0.1631678
## Run 13 stress 0.20339
## Run 14 stress 0.1441218
```

```

## Run 15 stress 0.1603938
## Run 16 stress 0.1448286
## Run 17 stress 0.1464672
## Run 18 stress 0.1448288
## Run 19 stress 0.163481
## Run 20 stress 0.1461826
## Run 21 stress 0.1448285
## Run 22 stress 0.1447244
## Run 23 stress 0.1440795
## Run 24 stress 0.1437894
## Run 25 stress 0.1440059
## Run 26 stress 0.1460187
## Run 27 stress 0.1608596
## Run 28 stress 0.1607464
## Run 29 stress 0.1447243
## Run 30 stress 0.1783682
## Run 31 stress 0.16279
## Run 32 stress 0.1441217
## Run 33 stress 0.1680941
## Run 34 stress 0.14409
## Run 35 stress 0.1787987
## Run 36 stress 0.1441224
## Run 37 stress 0.1957612
## Run 38 stress 0.1792859
## Run 39 stress 0.143983
## Run 40 stress 0.1461773
## Run 41 stress 0.1632396
## Run 42 stress 0.1476332
## Run 43 stress 0.1767431
## Run 44 stress 0.1957022
## Run 45 stress 0.1631683
## Run 46 stress 0.1651373
## Run 47 stress 0.1441227
## Run 48 stress 0.1422131
## ... New best solution
## ... Procrustes: rmse 0.0001393065 max resid 0.001069597
## ... Similar to previous best
## Run 49 stress 0.1422588
## ... Procrustes: rmse 0.005474481 max resid 0.04087919
## Run 50 stress 0.1422599
## ... Procrustes: rmse 0.005538216 max resid 0.04092405
## Run 51 stress 0.1629705
## Run 52 stress 0.1476323
## Run 53 stress 0.1797526
## Run 54 stress 0.1447243
## Run 55 stress 0.1422131
## ... New best solution
## ... Procrustes: rmse 2.935773e-05 max resid 0.0001991158
## ... Similar to previous best
## Run 56 stress 0.1460197
## Run 57 stress 0.1793203
## Run 58 stress 0.1629704
## Run 59 stress 0.1441219
## Run 60 stress 0.1669073

```

```

## Run 61 stress 0.1441217
## Run 62 stress 0.1441225
## Run 63 stress 0.1889157
## Run 64 stress 0.1441217
## Run 65 stress 0.1461787
## Run 66 stress 0.1448367
## Run 67 stress 0.1440782
## Run 68 stress 0.1823682
## Run 69 stress 0.1461774
## Run 70 stress 0.1604521
## Run 71 stress 0.1680986
## Run 72 stress 0.1773757
## Run 73 stress 0.1422273
## ... Procrustes: rmse 0.00161945  max resid 0.01137058
## Run 74 stress 0.1441218
## Run 75 stress 0.1603944
## Run 76 stress 0.1642271
## Run 77 stress 0.1448374
## Run 78 stress 0.1671516
## Run 79 stress 0.1461758
## Run 80 stress 0.1448286
## Run 81 stress 0.1760647
## Run 82 stress 0.1440895
## Run 83 stress 0.1821208
## Run 84 stress 0.1448367
## Run 85 stress 0.1448368
## Run 86 stress 0.1680973
## Run 87 stress 0.1422601
## ... Procrustes: rmse 0.004964037  max resid 0.04005418
## Run 88 stress 0.1460186
## Run 89 stress 0.1864423
## Run 90 stress 0.1439818
## Run 91 stress 0.1441218
## Run 92 stress 0.1634472
## Run 93 stress 0.142227
## ... Procrustes: rmse 0.001576281  max resid 0.01110712
## Run 94 stress 0.1440064
## Run 95 stress 0.1441217
## Run 96 stress 0.1439821
## Run 97 stress 0.164836
## Run 98 stress 0.1439818
## Run 99 stress 0.1441217
## Run 100 stress 0.1603937
## *** Solution reached

```

```

nmds.cm # convergent solution, stress = 0.142

```

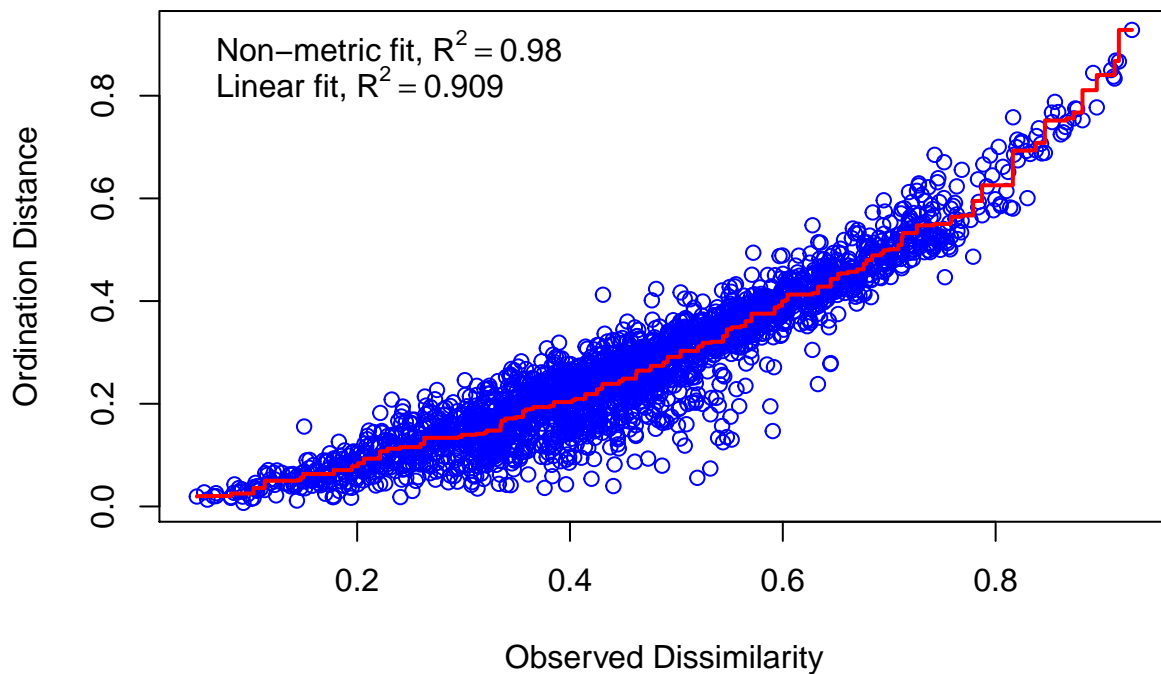
```

##
## Call:
## metaMDS(comm = wuf.cm, k = 2, try = 100, trymax = 200, autotransform = F)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      wuf.cm

```

```
## Distance: user supplied
##
## Dimensions: 2
## Stress:      0.1422131
## Stress type 1, weak ties
## Two convergent solutions found after 100 tries
## Scaling: centring, PC rotation
## Species: scores missing
```

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



## 3.2. Plots

Figure 6 in manuscript Chapter 3 of PhD thesis.

### 3.2.1. Total

```
# extract plotting data
nmds.ls.df <- plot_ordination(pstot.ls, nmds.ls, "samples", axes = 1:2, justDF = T)

# rename levels of Diet
nmds.ls.df$Diet <- revalue(nmds.ls.df$Diet, c("CF"="chicken feed",
```

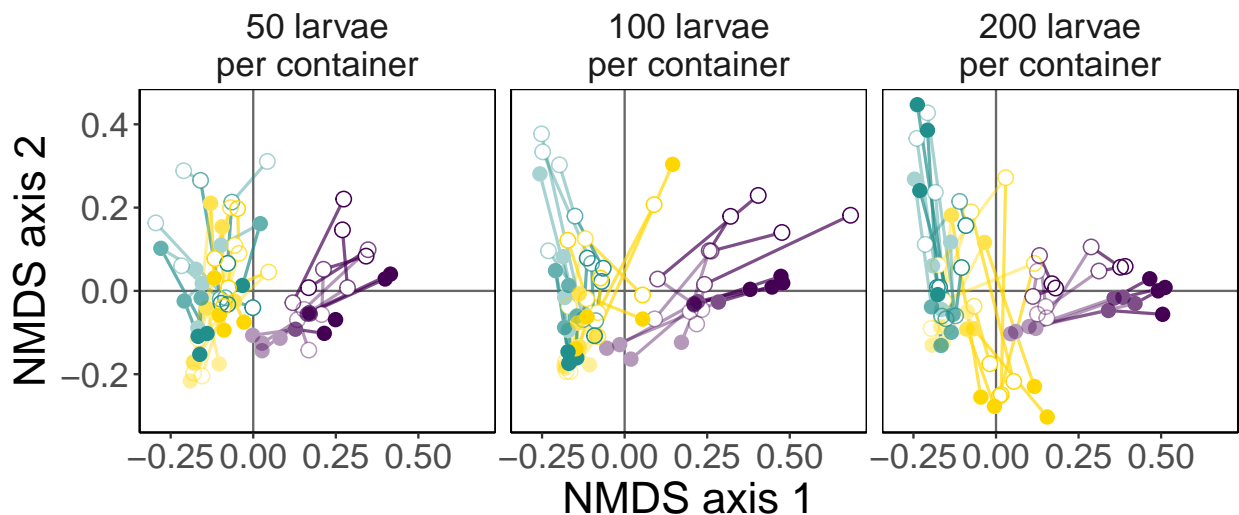
```

      "CS"="camelina", "CM"="chicken manure"))
# order by time
nmds.ls.df <- with(nmds.ls.df, nmds.ls.df[order(Timepoint),])

# plot without legend.
pNMDS1 <- ggplot(nmds.ls.df, aes(x=NMDS1, y=NMDS2, colour = Diet,
                                alpha = Timepoint,
                                group = interaction(Type,ContainerID)))
pNMDS1 <- pNMDS1 +
  geom_hline(yintercept=0, linetype="solid", color="grey40") +
  geom_vline(xintercept=0, linetype="solid", color="grey40") +
  geom_path(size = .7) +
  geom_point(shape = 16, size = 3, colour = "white", alpha = 1) +
  geom_point(aes(shape = Type), size = 3) +
  scale_shape_manual(values = c(16,1)) +
  scale_alpha_ordinal(range = c(0.4, 1)) +
  scale_color_manual(values = c("gold", "#21908CFF", "#440154FF")) +
  labs(x = "NMDS axis 1", y = "NMDS axis 2") +
  facet_grid(~ Density, labeller = labs_nmds) +
  theme_nmds + theme(legend.position = "none")

pNMDS1

```



### 3.2.1.B Legend

```

# plot with legend.
pNMDS2 <- ggplot(nmds.ls.df, aes(x=NMDS1, y=NMDS2, colour = Diet))
pNMDS2 <- pNMDS2 +
  geom_point(aes(shape = Type), size = 3) +
  scale_shape_manual(name = "Sample type", values = c(16,1)) +
  scale_color_manual(name = "Substrate",
                    values = c("gold", "#21908CFF", "#440154FF")) +
  guides(colour = guide_legend(order = 1),
         shape = guide_legend(order = 2)) +

```

```

theme_nmds
# plot only legend
nmds.legend <- get_legend(pNMDS2)
pNMDS.leg <- as_ggplot(nmds.legend)

pNMDS.leg

```

## Substrate

- chicken feed
- camelina
- chicken manure

## Sample type

- substrate
- larvae

```

# only colour legend.
pNMDS3 <- ggplot(nmds.ls.df, aes(x=NMDS1, y=NMDS2, colour = Diet))
pNMDS3 <- pNMDS3 +
  geom_point(size = 3) +
  scale_color_manual(name = "Substrate", values = c("gold", "#21908CFF", "#440154FF")) +
  theme_nmds
# plot only legend
nmds.legend2 <- get_legend(pNMDS3)
pNMDS.leg2 <- as_ggplot(nmds.legend2)

pNMDS.leg2

```



## Substrate

- chicken feed
- camelina
- chicken manure

### 3.2.2. Chicken feed

```
# extract plotting data
nm.ds.cf.df2 <- plot_ordination(CF.ls, nm.ds.cf, "samples", axes = 1:2, justDF = T)

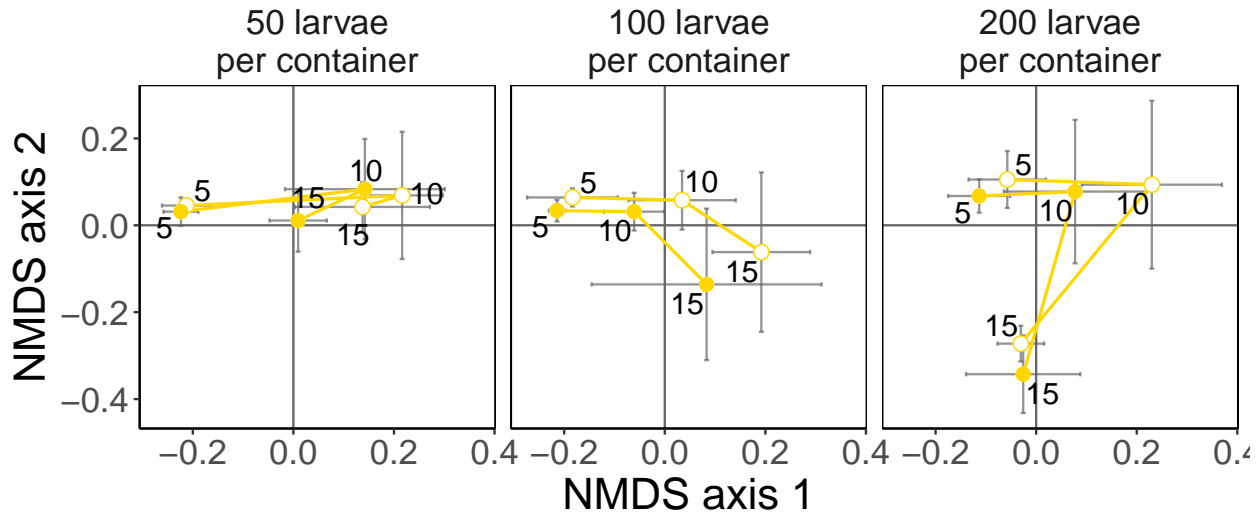
# order by time
nm.ds.cf.df2 <- with(nm.ds.cf.df2, nm.ds.cf.df2[order(Timepoint),])
# summarise scores
nm.ds.cf.sum2 <- ddp1y(nm.ds.cf.df2, .(Density, Timepoint, Type), summarise,
  mean1 = mean(NMDS1), mean2 = mean(NMDS2),
  se1 = se(NMDS1), se2 = se(NMDS2),
  sd1 = sd(NMDS1), sd2 = sd(NMDS2))

# errorbarplot
pNMDS.cf.sum2 <- ggplot(nm.ds.cf.sum2, aes(x=mean1, y=mean2,
  group = interaction(Type, Density)))
pNMDS.cf.sum2 <- pNMDS.cf.sum2 +
  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, colour = "gold") +
  geom_point(shape = 16, size=3, colour = "white") +
  geom_point(aes(shape = Type), size = 3, colour = "gold") +
  scale_shape_manual(values = c(16,1)) +
```

```

labs(x = "NMDS axis 1", y = "NMDS axis 2") +
geom_text_repel(aes(label = Timepoint), colour = "black", size = 5) +
facet_grid(~Density, labeller = labs_nmds) +
theme_nmds + theme(legend.position = "none")
pNMDS.cf.sum2

```



### 3.2.3. Camelina

```

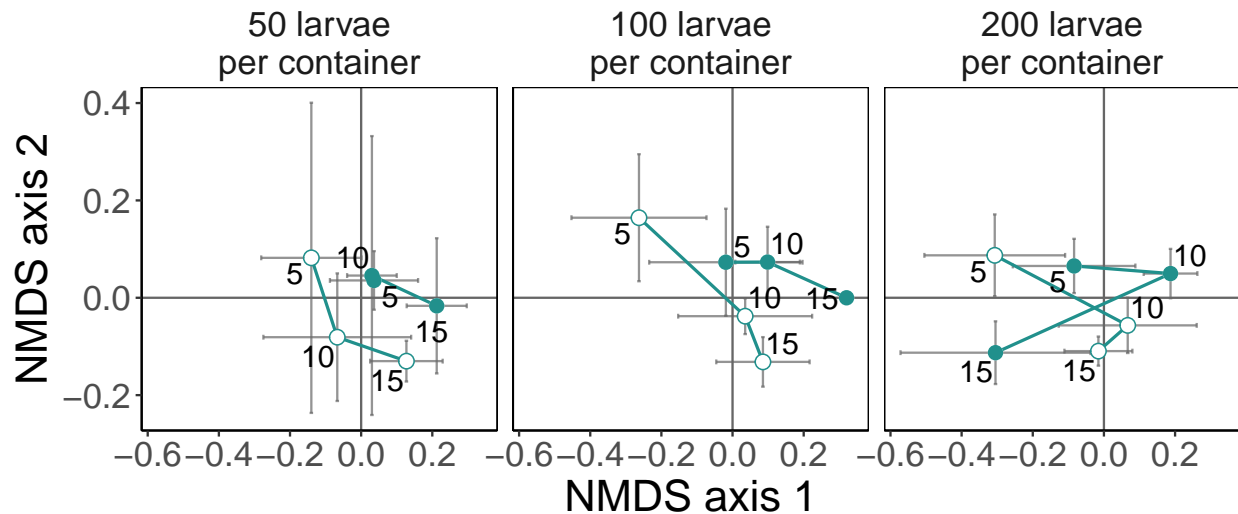
# extract plotting data
nmDS.cs.df2 <- plot_ordination(CS.ls, nmDS.cs, "samples", axes = 1:2, justDF = T)

# order by time
nmDS.cs.df2 <- with(nmDS.cs.df2, nmDS.cs.df2[order(Timepoint),])
# summarise scores
nmDS.cs.sum2 <- ddpIy(nmDS.cs.df2, .(Density, Timepoint, Type), summarise,
  mean1 = mean(NMDS1), mean2 = mean(NMDS2),
  se1 = se(NMDS1), se2 = se(NMDS2),
  sd1 = sd(NMDS1), sd2 = sd(NMDS2))

# errorbarplot, axes 1 and 2
pNMDS.cs.sum2 <- ggplot(nmDS.cs.sum2, aes(x=mean1, y=mean2,
  group = interaction(Type, Density)))
pNMDS.cs.sum2 <- pNMDS.cs.sum2 +
  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, colour = "#21908CFF") +
  geom_point(shape = 16, colour = "white", size = 3, alpha = 1) +
  geom_point(aes(shape = Type), size=3, colour = "#21908CFF") +
  scale_shape_manual(values = c(16,1)) +

```

```
labs(x = "NMDS axis 1", y = "NMDS axis 2") +
  geom_text_repel(aes(label = Timepoint), colour = "black", size = 5) +
  facet_grid(~Density, labeller = labs_nmds) +
  theme_nmds + theme(legend.position = "none")
pNMDS.cs.sum2
```



### 3.2.4. Chicken manure

```
# extract plotting data
nmds.cm.df2 <- plot_ordination(CM.ls, nmds.cm, "samples", axes = 1:2, justDF = T)

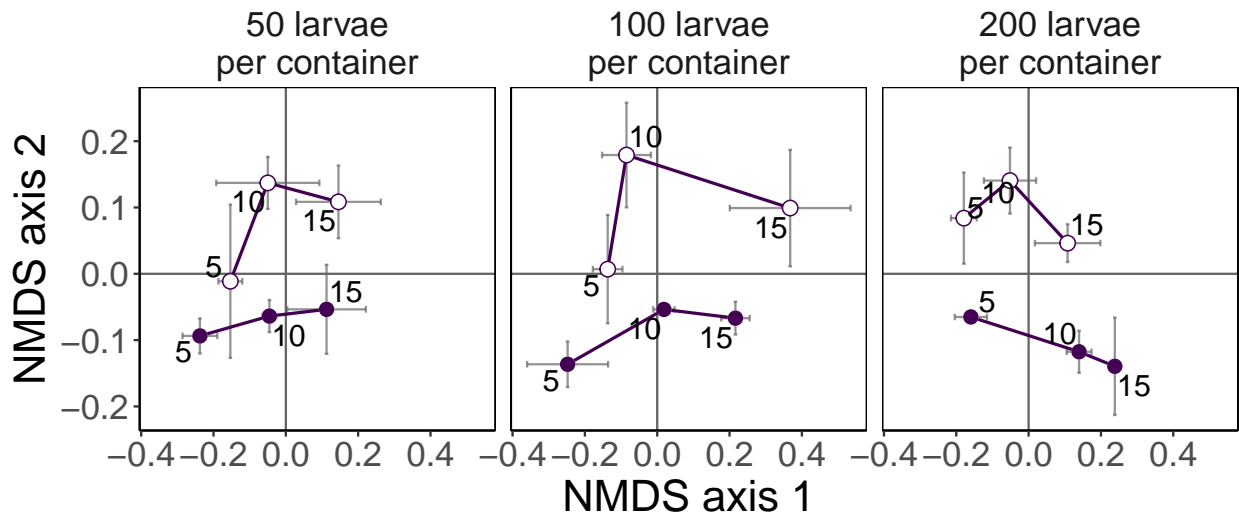
# order by time
nmds.cm.df2 <- with(nmds.cm.df2, nmds.cm.df2[order(Timepoint),])
# summarise scores
nmds.cm.sum2 <- ddpby(nmds.cm.df2, .(Density, Timepoint, Type), summarise,
  mean1 = mean(NMDS1), mean2 = mean(NMDS2),
  se1 = se(NMDS1), se2 = se(NMDS2),
  sd1 = sd(NMDS1), sd2 = sd(NMDS2))

# errorbarplot, axes 1 and 2
pNMDS.cm.sum2 <- ggplot(nmds.cm.sum2, aes(x=mean1, y=mean2,
  group = interaction(Type, Density)))
pNMDS.cm.sum2 <- pNMDS.cm.sum2 +
  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, colour = "#440154FF") +
  geom_point(shape = 16, colour = "white", size = 3, alpha = 1) +
  geom_point(aes(shape = Type), size=3, colour = "#440154FF") +
  scale_shape_manual(values = c(16,1)) +
```

```

labs(x = "NMDS axis 1", y = "NMDS axis 2") +
geom_text_repel(aes(label = Timepoint), colour = "black", size = 5) +
facet_grid(~Density, labeller = labs_nmds) +
theme_nmds + theme(legend.position = "none")
pNMDS.cm.sum2

```



#### 4. Export plots

```

# substrates

# total: individual points
ggsave(plot = pNMDS, "./figures/NMDS_S.png", h = 3.5, w = 10)
ggsave(plot = pNMDS.leg2, "./figures/NMDS_leg-S.png", h = 4, w = 2)

# per diet: errorbarplots
ggsave(plot = pNMDS.cf.sum, "./figures/NMDS_CF_sd-S.png", h = 3.5, w = 10)
ggsave(plot = pNMDS.cm.sum, "./figures/NMDS_CM_sd-S.png", h = 3.5, w = 10)
ggsave(plot = pNMDS.cs.sum, "./figures/NMDS_CS_sd-S.png", h = 3.5, w = 10)

# larvae and substrates

# total: individual points
ggsave(plot = pNMDS1, "./figures/NMDS_LS.png", h = 3.5, w = 8)
ggsave(plot = pNMDS.leg, "./figures/NMDS_leg-LS.png", h = 4, w = 2)

# per diet: errorbarplots
ggsave(plot = pNMDS.cf.sum2, "./figures/NMDS_CF_sd-LS.png", h = 3.5, w = 8)
ggsave(plot = pNMDS.cm.sum2, "./figures/NMDS_CM_sd-LS.png", h = 3.5, w = 8)
ggsave(plot = pNMDS.cs.sum2, "./figures/NMDS_CS_sd-LS.png", h = 3.5, w = 8)

# PDF files

```

```

ggsave(plot = pNMDS, "./figures/NMDS_S.pdf", h = 87.5, w = 250, u = "mm")
ggsave(plot = pNMDS.leg2, "./figures/NMDS_leg-S.pdf", h = 100, w = 50, u = "mm")
ggsave(plot = pNMDS.cf.sum, "./figures/NMDS_CF_sd-S.pdf", h = 87.5, w = 250, u = "mm")
ggsave(plot = pNMDS.cm.sum, "./figures/NMDS_CM_sd-S.pdf", h = 87.5, w = 250, u = "mm")
ggsave(plot = pNMDS.cs.sum, "./figures/NMDS_CS_sd-S.pdf", h = 87.5, w = 250, u = "mm")

ggsave(plot = pNMDS1, "./figures/NMDS_LS.pdf", h = 87.5, w = 200, u = "mm")
ggsave(plot = pNMDS.leg, "./figures/NMDS_leg-LS.pdf", h = 100, w = 50, u = "mm")
ggsave(plot = pNMDS.cf.sum2, "./figures/NMDS_CF_sd-LS.pdf", h = 87.5, w = 200, u = "mm")
ggsave(plot = pNMDS.cm.sum2, "./figures/NMDS_CM_sd-LS.pdf", h = 87.5, w = 200, u = "mm")
ggsave(plot = pNMDS.cs.sum2, "./figures/NMDS_CS_sd-LS.pdf", h = 87.5, w = 200, u = "mm")

```