Deep Dive Into Gradients

Smit, A. J.
University of the Western Cape

2021-01-01

Table of contents

Material required for this chapter

Type	Name	Link
Reading	Smit et al. (2017)	`Smit_et_al_2017.pdf`
	Smit et al. (2013)	`Smit_et_al_2013.pdf`
	Supp. to Smit et al. (2017)	`Smit_the_seaweed_data.pdf`
Related	Appendices to Smit et al. (2017)	`Appendices`
Data	The seaweed environmental data	`SeaweedEnv.RData`
	The seaweed species data	`dists_mat.RData`
	The bioregions	`bioregions.csv`

In the previous chapter we looked at calculations involving biodiversity (specifically the dissimilarity matrices made from a species table) and environmental variables (distances) from the paper

by Smit et al. (2017). What can we do with the two forms of contemporary β-diversity? What do they mean? Can we look to environmental distances for more insight?

Let's do a deeper analysis and create a figure to demonstrate these findings. I regress $\beta_{\text{sør}}$ on the spatial distance between section pairs (see below) and on the environmental distance β_{E} in each bioregion and used the magnitude of the slope (per 100 km) of this relationship as a metric of β-diversity or 'distance decay' of dissimilarity.

What these lines of code do is recreate Figure 5 in Smit et al. (2017). Please read the paper for an interpretation of this figure as this is critical for an understanding of the role that gradients play in structuring patterns of biodiversity.

(To be updated...)

```
## Setting up the analysis environment
library(tidyverse)
library(plyr)
library(vegan)
library(betapart) # for partitioning beta-diversity
```

1 Load and Prepare All the Data

1.1 The environmental data

```
# load the environmental data...
load("../data/seaweed/SeaweedEnv.RData")
env <- as.data.frame(env)
# keep only some...
env <- env[, c("annMean", "annRange", "annSD", "febMean", "febRange",
               "febSD", "augMean", "augRange", "augSD")]
```

1.2 The bioregional classification

Various bioregions have been defined for South African marine biota. I prefer to use the one made by Bolton and Stegenga (2002):

```
# load the bioregions data...
bioreg <- read.csv("../data/seaweed/bioregions.csv",
                   header = TRUE)
rbind(head(bioreg, 3), tail(bioreg, 3))
```

```
   spal.prov spal.ecoreg lombard bolton
1        BMP          NE   NamBR    BMP
2        BMP          NE   NamBR    BMP
3        BMP          NE   NamBR    BMP
56       AMP          NE     NBR   ECTZ
```

```
57       AMP          NE     NBR   ECTZ
58       AMP          NE     NBR   ECTZ
```

1.3 The geographic distances

Since the connectivity between sections is constrained by their location along a shoreline, we calculated the distances between sections not as 'as the crow flies' distances (e.g. Section 1 is not connected in a straight line to Section 58 because of the intervening land in-between), but as the great circle geodesic distances between each pair of sections along a 'route'. Travelling from **1** to **58** therefore requires visiting **2**, then **3**, and eventually all the way up to **58**. The total distance between a pair of arbitrary sections is thus the cumulative sum of the great circle distances between each consecutive pair of intervening sections along the route. These data are contained in `dists_mat.RData` (I prepared it earlier):

```
# load the distances matrix...
load("../data/seaweed/dists_mat.RData")
# loaded as dists_mat
dists.mat[1:10, 1:8]
```

```
         1       2       3       4       5       6       7       8
1    0.000  51.138 104.443 153.042 207.386 253.246 305.606 359.799
2   51.138   0.000  53.305 101.904 156.248 202.108 254.468 308.661
3  104.443  53.305   0.000  48.599 102.943 148.803 201.163 255.356
4  153.042 101.904  48.599   0.000  54.344 100.204 152.564 206.757
5  207.386 156.248 102.943  54.344   0.000  45.860  98.220 152.413
6  253.246 202.108 148.803 100.204  45.860   0.000  52.360 106.553
7  305.606 254.468 201.163 152.564  98.220  52.360   0.000  54.193
8  359.799 308.661 255.356 206.757 152.413 106.553  54.193   0.000
9  409.263 358.125 304.820 256.221 201.877 156.017 103.657  49.464
10 457.857 406.719 353.414 304.815 250.471 204.611 152.251  98.058
```

Make a copy of the original matrix of distances between pairs of sites to create a full matrix which constrains pairwise comparisons to pairs within bioregions:

```
bioreg_mat <- dists.mat
bioreg_mat[1:58, 1:58] <- "out"
bioreg_mat[1:16, 1:16] <- "BMP"
bioreg_mat[17:21, 17:21] <- "B-ATZ"
bioreg_mat[22:41, 22:41] <- "AMP"
bioreg_mat[42:58, 42:58] <- "ECTZ"
dim(bioreg_mat)
```

```
[1] 58 58
```

```
# see what is inside the matrix...
bioreg_mat[1:3, 1:10]
```

```
  1     2     3     4     5     6     7     8     9     10
1 "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP"
2 "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP"
3 "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP" "BMP"
```

```
bioreg_mat[56:58, 53:58]
```

```
   53     54     55     56     57     58
56 "ECTZ" "ECTZ" "ECTZ" "ECTZ" "ECTZ" "ECTZ"
57 "ECTZ" "ECTZ" "ECTZ" "ECTZ" "ECTZ" "ECTZ"
58 "ECTZ" "ECTZ" "ECTZ" "ECTZ" "ECTZ" "ECTZ"
```

```
# convert to show only the lower left triangle (not used later)
# requires the gdata package...
bioreg_tri <- gdata::lowerTriangle(bioreg_mat, diag = FALSE)
```

In `bioreg_._mat`, pairs of sites that do not fall within any of the bioregions are labelled 'out':

```
# print output below...
bioreg_mat[1:3, 53:58]
```

```
  53    54    55    56    57    58
1 "out" "out" "out" "out" "out" "out"
2 "out" "out" "out" "out" "out" "out"
3 "out" "out" "out" "out" "out" "out"
```

We extract the slices (groups of rows) of the original species table into separate dataframes, one for each of the four bioregions:

```
env_BMP <- env[1:16, ]
env_BATZ <- env[17:21, ]
env_AMP <- env[22:41, ]
env_ECTZ <- env[42:58, ]
```

Now we make an environmental dataframe for use with plots of pairwise correlations etc.:

```
env_df <- data.frame(bio = bioreg$bolton, round(env, 3))
rbind(head(env_df, 3), tail(env_df, 3))
```

```
    bio annMean annRange annSD febMean febRange febSD augMean augRange augSD
1   BMP  12.335    1.249 1.255  13.001    6.070 1.626  11.752    2.502 0.767
2   BMP  12.388    1.802 1.402  13.379    5.889 1.754  11.577    2.973 0.897
3   BMP  12.243    2.068 1.475  13.362    5.431 1.704  11.294    3.084 0.941
56 ECTZ  23.729    4.609 1.942  26.227    3.474 1.191  21.618    2.163 0.663
57 ECTZ  24.710    4.969 1.976  27.328    3.372 1.143  22.359    1.584 0.499
58 ECTZ  25.571    5.574 2.023  28.457    3.267 1.000  22.883    1.098 0.349
```

1.4 The seaweed species data

```
# load the seaweed data...
spp <- read.csv('../data/seaweed/SeaweedSpp.csv')
spp <- dplyr::select(spp, -1)
spp[1:10, 1:10]
```

```
   ACECAL ACEMOE ACRVIR AROSP1 ANAWRI AVRSP1 BIDMAG BIDMIN BOEFOR BOOCOM
1       0      0      0      0      0      0      0      0      0      0
2       0      0      0      0      0      0      0      0      0      0
3       0      0      0      0      0      0      0      0      0      0
4       0      0      0      0      0      0      0      0      0      0
5       0      0      0      0      0      0      0      0      0      0
6       0      0      0      0      0      0      0      0      0      0
7       0      0      0      0      0      0      0      0      0      0
8       0      0      0      0      0      0      0      0      0      0
9       0      0      0      0      0      0      0      0      0      0
10      0      0      0      0      0      0      0      0      0      0
```

2 Calculate β-Diversity Indices

Calculate β-diversity using the Sørensen index of dissimilarity. This is used throughout; binary Bray-Curtis is equivalent to Sørensen in **vegan**. I then extract the subdiagonal from this matrix of species dissimilarities. The subdiagonal refers to the elements immediately below the main diagonal. For a matrix Y with elements y_{ij}, the subdiagonal elements are $y_{i,i+1}$.

```
# ---- Sorensen-index ----
## this is used throughout...
Y <- vegdist(spp, binary = TRUE)
Y_mat <- as.matrix(Y)
# extract the subdiagonal...
Y_diag <- diag(Y_mat[-1, -nrow(Y_mat)])
# add a zero in front...
Y_diag <- append(0, Y_diag, after = 1)
```

Decompose into turnover and nestedness-resultant beta-diversity:

```
# ---- do-betapart ----
## Calculations with betapart...
Y.core <- betapart.core(spp)

# Using the Sørensen index, compute three distance matrices accounting for
# the (i) turnover (replacement), (ii) nestedness-resultant component, and
# (iii) total dissimilarity (i.e. the sum of both components)
# use for pairwise plotting...
Y.pair <- beta.pair(Y.core, index.family = "sor")
```

Extract the subdiagonal for plotting later on:

```
# Y1 will be the turnover component
Y1_mat <- as.matrix(Y.pair$beta.sim)
# extract the subdiagonal...
Y1_diag <- diag(Y1_mat  [-1, -nrow(Y1_mat)])
# add a zero in front...
Y1_diag <- append(0, Y1_diag, after = 1)

# Y2 will be the nestedness-resultant component
Y2_mat <- as.matrix(Y.pair$beta.sne)
Y2_diag <- diag(Y2_mat[-1, -nrow(Y2_mat)])
Y2_diag <- append(0, Y2_diag, after = 1)
```

Create separate matrices for each bioregion:

```
# ---- spp-bioregion ----
spp.BMP <- spp[1:16, ]
Y.BMP <- vegdist(spp.BMP, binary = TRUE)
spp.core.BMP <- betapart.core(spp.BMP)
# use below for pairwise plotting...
Y.pair.BMP <- beta.pair(spp.core.BMP, index.family = "sor")

spp.BATZ <- spp[17:21, ]
Y.BATZ <- vegdist(spp.BATZ, binary = TRUE)
spp.core.BATZ <- betapart.core(spp.BATZ)
# use below for pairwise plotting...
Y.pair.BATZ <- beta.pair(spp.core.BATZ, index.family = "sor")

spp.AMP <- spp[22:41, ]
Y.AMP <- vegdist(spp.AMP, binary = TRUE)
spp.core.AMP <- betapart.core(spp.AMP)
# use below for pairwise plotting...
Y.pair.AMP <- beta.pair(spp.core.AMP, index.family = "sor")

spp.ECTZ <- spp[42:58, ]
Y.ECTZ <- vegdist(spp.ECTZ, binary = TRUE)
```

```
spp.core.ECTZ <- betapart.core(spp.ECTZ)
# use below for pairwise plotting...
Y.pair.ECTZ <- beta.pair(spp.core.ECTZ, index.family = "sor")
```

Calculate species richness (α-diversity):

```
# ---- do-species-richness ----
spp.richness.site <- specnumber(spp)
```

Calculate the environmental distances:

```
# ---- environmental-distance ----
# Euclidian distances on temperatures
# first make a copy so we can use untransformed data later on...
env_raw <- env
# calculate z-scores...
env <- decostand(env, method = "standardize")
```

Using individual thermal variables, calculate Euclidian distances, make a matrix, and extract the subdiagonal. The data have already been standardised in `env`:

```
# augMean
# to be used in env_rda2...
env4_mat <- env |>
  dplyr::select(augMean) |>
  vegdist(method = 'euclidian') |>
  as.matrix()

env4_diag <- diag(env4_mat[-1, -nrow(env4_mat)])
env4_diag <- append(0, env4_diag, after = 1)
```

```
# febRange
# to be used in env_rda2...
env5_mat <- env |>
  dplyr::select(febRange) |>
  vegdist(method = 'euclidian') |>
  as.matrix()

env5_diag <- diag(env5_mat[-1, -nrow(env5_mat)])
env5_diag <- append(0, env5_diag, after = 1)
```

```
# febSD
# to be used in env_rda2...
env6_mat <- env |>
```

```
  dplyr::select(febSD) |>
  vegdist(method = 'euclidian') |>
  as.matrix()

env6_diag <- diag(env6_mat[-1, -nrow(env6_mat)])
env6_diag <- append(0, env6_diag, after = 1)
```

```
# augSD
# to be used in env_rda2...
env7_mat <- env |>
  dplyr::select(augSD) |>
  vegdist(method = 'euclidian') |>
  as.matrix()

env7_diag <- diag(env7_mat[-1, -nrow(env7_mat)])
env7_diag <- append(0, env7_diag, after = 1)
```

```
# annMean
# to be used in env_rda2...
env8_mat <- env |>
  dplyr::select(annMean) |>
  vegdist(method = 'euclidian') |>
  as.matrix()

env8_diag <- diag(env8_mat[-1, -nrow(env8_mat)])
env8_diag <- append(0, env8_diag, after = 1)
```

```
# combined variables selected with the db-RDA
# these have a far poorer fit...
env_comb_mat <- env |>
  dplyr::select(augMean, febRange, febSD, augSD) |>
  vegdist(method = 'euclidian') |>
  as.matrix()

env_comb_diag <- diag(env_comb_mat[-1, -nrow(env_comb_mat)])
env_comb_diag <- append(0, env_comb_diag, after = 1)
```

```
# ---- do-figure-5 ----
# assemble data frame for plotting...
spp_df <- data.frame(dist = as.vector(dists.mat),
                     bio = as.vector(bioreg_mat),
                     augMean = as.vector(env4_mat),
                     febRange = as.vector(env5_mat),
                     febSD = as.vector(env6_mat),
```

```
                    augSD = as.vector(env7_mat),
                    annMean = as.vector(env8_mat),
                    Y = as.vector(Y_mat),
                    Y1 = as.vector(Y1_mat),
                    Y2 = as.vector(Y2_mat))

# include only site pairs that fall within bioregions...
spp_df2 <- droplevels(subset(spp_df, bio !=  "out"))
rbind(head(spp_df2, 3), tail(spp_df2, 3))
```

```
        dist  bio    augMean   febRange      febSD     augSD    annMean
1      0.000  BMP 0.00000000 0.00000000 0.00000000 0.0000000 0.00000000
2     51.138  BMP 0.05741369 0.09884404 0.16295271 0.3132800 0.01501846
3    104.443  BMP 0.15043904 0.34887754 0.09934163 0.4188239 0.02602247
3362 102.649 ECTZ 0.41496099 0.11330069 0.24304493 0.7538546 0.52278161
3363  49.912 ECTZ 0.17194242 0.05756093 0.18196664 0.3604341 0.24445006
3364   0.000 ECTZ 0.00000000 0.00000000 0.00000000 0.0000000 0.00000000
                Y        Y1          Y2
1    0.000000000 0.0000000 0.000000000
2    0.003610108 0.0000000 0.003610108
3    0.003610108 0.0000000 0.003610108
3362 0.198728140 0.1948882 0.003839961
3363 0.069337442 0.0443038 0.025033645
3364 0.000000000 0.0000000 0.000000000
```

I'll save this file with the combined data for use later in the Multiple Regression Chapter:

```
write.csv(spp_df2, file = "../data/seaweed/spp_df2.csv")
```

Do the various linear regressions of Sørensen dissimilarities ($\beta_{\text{sør}}$), turnover (β_{sim}) and nestedness-related β-diversity (β_{sne}) as a function of the various thermal distances. I only display the results of the linear regression for $Y1$ regressed on geographical distance, `dist`, but do all the calculations:

```
# turnover...
Y1_lm1 <- dlply(spp_df2, .(bio), function(x) lm(Y1 ~ dist, data = x))
lapply(Y1_lm1, summary)
```

```
$AMP

Call:
lm(formula = Y1 ~ dist, data = x)

Residuals:
      Min        1Q    Median        3Q       Max
-0.059575 -0.019510 -0.004546  0.015061  0.067655
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.175e-03  2.406e-03  -2.151   0.0321 *
dist         2.939e-04  6.567e-06  44.751   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02786 on 398 degrees of freedom
Multiple R-squared:  0.8342,    Adjusted R-squared:  0.8338
F-statistic:  2003 on 1 and 398 DF,  p-value: < 2.2e-16


$`B-ATZ`

Call:
lm(formula = Y1 ~ dist, data = x)

Residuals:
      Min        1Q    Median        3Q       Max
-0.070629 -0.024865  0.008058  0.022698  0.059443

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.008058   0.013645  -0.591    0.561
dist         0.001093   0.000159   6.873 5.23e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0411 on 23 degrees of freedom
Multiple R-squared:  0.6726,    Adjusted R-squared:  0.6583
F-statistic: 47.24 on 1 and 23 DF,  p-value: 5.229e-07


$BMP

Call:
lm(formula = Y1 ~ dist, data = x)

Residuals:
      Min        1Q    Median        3Q       Max
-0.037751 -0.027462 -0.023894  0.001529  0.269377

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.392e-02  5.500e-03   4.350 1.97e-05 ***
dist        7.095e-05  1.826e-05   3.886  0.00013 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05129 on 254 degrees of freedom
Multiple R-squared:  0.05613,   Adjusted R-squared:  0.05241
F-statistic:  15.1 on 1 and 254 DF,  p-value: 0.0001299


$ECTZ

Call:
lm(formula = Y1 ~ dist, data = x)

Residuals:
     Min       1Q   Median       3Q      Max
-0.11882 -0.02685  0.00540  0.02440  0.11961

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.400e-03  4.257e-03  -1.268    0.206
dist         7.860e-04  1.209e-05  65.033   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04194 on 287 degrees of freedom
Multiple R-squared:  0.9365,    Adjusted R-squared:  0.9362
F-statistic:  4229 on 1 and 287 DF,  p-value: < 2.2e-16
```

```
Y1_lm2 <- dlply(spp_df2, .(bio), function(x) lm(Y1 ~ augMean , data = x))
# lapply(Y1_lm2, summary)
Y1_lm3 <- dlply(spp_df2, .(bio), function(x) lm(Y1 ~ augSD , data = x))
# lapply(Y1_lm3, summary)
Y1_lm4 <- dlply(spp_df2, .(bio), function(x) lm(Y1 ~ febRange , data = x))
# lapply(Y1_lm4, summary)
Y1_lm5 <- dlply(spp_df2, .(bio), function(x) lm(Y1 ~ febSD , data = x))
# lapply(Y1_lm5, summary)

# nestedness-resultant...
Y2_lm1 <- dlply(spp_df2, .(bio), function(x) lm(Y2 ~ dist, data = x))
# lapply(Y2_lm1, summary)
Y2_lm2 <- dlply(spp_df2, .(bio), function(x) lm(Y2 ~ annMean , data = x))
# lapply(Y2_lm2, summary)
```

3 Make the Plots

Now assemble **Figure 5.** in Smit et al. (2017). It is a plot of pairwise (a) Sørensen dissimilarities ($\beta_{\text{sør}}$), (b) turnover (β_{sim}) and (c) nestedness-related β-diversity (β_{sne}) (Baselga 2010) as a function of distance between sections. Section pairs falling within individual bioregions are colour-coded;

where the pairs include sections across different bioregions the symbols are coloured grey and labeled 'out'.

Combine the data in a way that makes for easy plotting:

```
# Plots...
spp_long <- spp_df %>%
  gather(beta, dissim, Y:Y2) %>%
  gather(metric, distance, c(dist, augMean:annMean))
spp_long$metric = factor(spp_long$metric,
                         levels = c('dist', 'augMean', 'febRange',
                                    'febSD', 'augSD', 'annMean'))
```

The repetitive portions of code needed to create each of the panels. I was too lazy to write neater and more concise code:

```
# sim as a function of geographic distance...
plt5a <- spp_long %>%
  dplyr::filter(beta %in% "Y1" & metric %in% "dist") %>%
  ggplot(aes(x = distance, y = dissim, group = bio)) +
  geom_point(aes(colour = bio, shape = bio), size = 1.2, alpha = 0.8) +
  geom_point(aes(colour = bio, size = bio, alpha = bio, shape = bio)) +
  geom_line(stat = "smooth", method = "lm", formula = y ~ x, alpha = 1.0,
            size = 0.6, colour = "black", aes(linetype = bio)) +
  scale_linetype_manual(name = "Bioregion",
                        values = c("dashed", "solid", "dotted",
                                   "longdash", "blank")) +
  scale_colour_brewer(name = "Bioregion",
                      palette = "Set1") +
  scale_shape_manual(name = "Bioregion",
                     values = c(0, 19, 2, 5, 46)) +
  scale_size_manual(name = "Bioregion",
                    values = c(1.0, 1.2, 1.0, 1.0, 0.6)) +
  scale_alpha_manual(name = "Bioregion",
                     values = c(0.85, 1.0, 0.85, 0.85, 0.1)) +
  xlab(expression(paste("Distance (km)"))) +
  ylab(expression(paste(beta[sim]))) +
  scale_y_continuous(limits = c(0, 0.75)) +
  scale_x_continuous(limits = c(0, 1000)) +
  theme_grey() +
  theme(panel.grid.minor = element_line(colour = NA),
        plot.title = element_text(hjust = 0, size = 10),
        # legend.position = c(0.2, 0.7),
        # legend.direction = "vertical",
        aspect.ratio = 0.6) +
  ggtitle(expression(paste(beta[sim], " as a function of distance")))
```

```r
# sim as a function of augMean...
plt5b <- spp_long %>%
  dplyr::filter(beta %in% "Y1" & metric %in% "augMean") %>%
  ggplot(aes(x = distance, y = dissim, group = bio)) +
  geom_point(aes(colour = bio, shape = bio), size = 1.2, alpha = 0.8) +
  # geom_point(aes(colour = bio, size = bio, alpha = bio, shape = bio)) +
  geom_line(stat = "smooth", method = "lm", formula = y ~ x,
            alpha = 1.0, size = 0.6, colour = "black", aes(linetype = bio)) +
  scale_linetype_manual(values = c("dashed", "solid", "dotted",
                                   "longdash", "blank")) +
  scale_colour_brewer(palette = "Set1") +
  scale_shape_manual(values = c(0, 19, 2, 5, 46)) +
  scale_size_manual(values = c(1.0, 1.2, 1.0, 1.0, 0.6)) +
  scale_alpha_manual(values = c(0.85, 1.0, 0.85, 0.85, 0.1)) +
  xlab(expression(paste(d[E]))) +
  ylab(expression(paste(beta[sim]))) +
  scale_y_continuous(limits = c(0, 0.75)) +
  scale_x_continuous(limits = c(0, 2)) +
  theme_grey() +
  theme(panel.grid.minor = element_line(colour = NA),
        plot.title = element_text(hjust = 0, size = 10),
        legend.position = "none",
        # legend.title = element_blank(),
        legend.title = element_text(size = 8),
        legend.text = element_text(size = 8),
        legend.key = element_blank(),
        legend.key.height = unit(.22, "cm"),
        legend.background = element_blank(),
        aspect.ratio = 0.6) +
  ggtitle(expression(paste(beta[sim], " as a function of augMean")))
```

```r
# sim as a function of febRange...
plt5c <- spp_long %>%
  dplyr::filter(beta %in% "Y1" & metric %in% "febRange") %>%
  ggplot(aes(x = distance, y = dissim, group = bio)) +
  geom_point(aes(colour = bio, shape = bio), size = 1.2, alpha = 0.8) +
  # geom_point(aes(colour = bio, size = bio, alpha = bio, shape = bio)) +
  geom_line(stat = "smooth", method = "lm", formula = y ~ x,
            alpha = 1.0, size = 0.6, colour = "black", aes(linetype = bio)) +
  scale_linetype_manual(values = c("dashed", "solid", "dotted",
                                   "longdash", "blank")) +
  scale_colour_brewer(palette = "Set1") +
  scale_shape_manual(values = c(0, 19, 2, 5, 46)) +
  scale_size_manual(values = c(1.0, 1.2, 1.0, 1.0, 0.6)) +
  scale_alpha_manual(values = c(0.85, 1.0, 0.85, 0.85, 0.1)) +
  xlab(expression(paste(d[E]))) +
  ylab(expression(paste(beta[sim]))) +
```

```
  scale_y_continuous(limits = c(0, 0.75)) +
  scale_x_continuous(limits = c(0, 4)) +
  theme_grey() +
  theme(panel.grid.minor = element_line(colour = NA),
        plot.title = element_text(hjust = 0, size = 10),
        legend.position = "none",
        aspect.ratio = 0.6) +
  ggtitle(expression(paste(beta[sim], " as a function of febRange")))
```

```
# sim as a function of febSD...
plt5d <- spp_long %>%
  dplyr::filter(beta %in% "Y1" & metric %in% "febSD") %>%
  ggplot(aes(x = distance, y = dissim, group = bio)) +
  geom_point(aes(colour = bio, shape = bio), size = 1.2, alpha = 0.8) +
  # geom_point(aes(colour = bio, size = bio, alpha = bio, shape = bio)) +
  geom_line(stat = "smooth", method = "lm", formula = y ~ x,
            alpha = 1.0, size = 0.6, colour = "black", aes(linetype = bio)) +
  scale_linetype_manual(values = c("dashed", "solid", "dotted",
                                   "longdash", "blank")) +
  scale_colour_brewer(palette = "Set1") +
  scale_shape_manual(values = c(0, 19, 2, 5, 46)) +
  scale_size_manual(values = c(1.0, 1.2, 1.0, 1.0, 0.6)) +
  scale_alpha_manual(values = c(0.85, 1.0, 0.85, 0.85, 0.1)) +
  xlab(expression(paste(d[E]))) +
  ylab(expression(paste(beta[sim]))) +
  scale_y_continuous(limits = c(0, 0.75)) +
  scale_x_continuous(limits = c(0, 3)) +
  theme_grey() +
  theme(panel.grid.minor = element_line(colour = NA),
        plot.title = element_text(hjust = 0, size = 10),
        legend.position = "none",
        aspect.ratio = 0.6) +
  ggtitle(expression(paste(beta[sim], " as a function of febSD")))
```

```
# sim as a function of augSD...
plt5e <- spp_long %>%
  dplyr::filter(beta %in% "Y1" & metric %in% "augSD") %>%
  ggplot(aes(x = distance, y = dissim, group = bio)) +
  geom_point(aes(colour = bio, shape = bio), size = 1.2, alpha = 0.8) +
  # geom_point(aes(colour = bio, size = bio, alpha = bio, shape = bio)) +
  geom_line(stat = "smooth", method = "lm", formula = y ~ x,
            alpha = 1.0, size = 0.6, colour = "black", aes(linetype = bio)) +
  scale_linetype_manual(values = c("dashed", "solid", "dotted",
                                   "longdash", "blank")) +
  scale_colour_brewer(palette = "Set1") +
  scale_shape_manual(values = c(0, 19, 2, 5, 46)) +
```

```
  scale_size_manual(values = c(1.0, 1.2, 1.0, 1.0, 0.6)) +
  scale_alpha_manual(values = c(0.85, 1.0, 0.85, 0.85, 0.1)) +
  xlab(expression(paste(d[E]))) +
  ylab(expression(paste(beta[sim]))) +
  scale_y_continuous(limits = c(0, 0.75)) +
  scale_x_continuous(limits = c(0, 3)) +
  theme_grey() +
  theme(panel.grid.minor = element_line(colour = NA),
        plot.title = element_text(hjust = 0, size = 10),
        legend.position = "none",
        aspect.ratio = 0.6) +
  ggtitle(expression(paste(beta[sim], " as a function of augSD")))
```

```
# sne as a function of distance...
plt5f <- spp_long %>%
  dplyr::filter(beta %in% "Y2" & metric %in% "dist") %>%
  ggplot(aes(x = distance, y = dissim, group = bio)) +
  geom_point(aes(colour = bio, shape = bio), size = 1.2, alpha = 0.8) +
  # geom_point(aes(colour = bio, size = bio, alpha = bio, shape = bio)) +
  geom_line(stat = "smooth", method = "lm", formula = y ~ x,
            alpha = 1.0, size = 0.6, colour = "black", aes(linetype = bio)) +
  scale_linetype_manual(values = c("dashed", "solid", "dotted",
                                   "longdash", "blank")) +
  scale_colour_brewer(palette = "Set1") +
  scale_shape_manual(values = c(0, 19, 2, 5, 46)) +
  scale_size_manual(values = c(1.0, 1.2, 1.0, 1.0, 0.6)) +
  scale_alpha_manual(values = c(0.85, 1.0, 0.85, 0.85, 0.1)) +
  xlab(expression(paste("Distance (km)"))) +
  ylab(expression(paste(beta[sne]))) +
  scale_y_continuous(limits = c(0, 0.22)) +
  scale_x_continuous(limits = c(0, 1000)) +
  theme_grey() +
  theme(panel.grid.minor = element_line(colour = NA),
        plot.title = element_text(hjust = 0, size = 10),
        legend.position = "none",
        aspect.ratio = 0.6) +
  ggtitle(expression(paste(beta[sne], " as a function of distance")))
```

```
# sne as a function of annMean...
plt5g <- spp_long %>%
  dplyr::filter(beta %in% "Y2" & metric %in% "annMean") %>%
  ggplot(aes(x = distance, y = dissim, group = bio)) +
  geom_point(aes(colour = bio, shape = bio), size = 1.2, alpha = 0.8) +
  # geom_point(aes(colour = bio, size = bio, alpha = bio, shape = bio)) +
  geom_line(stat = "smooth", method = "lm", formula = y~x, alpha = 1.0, size =
0.6,
```

```
            colour = "black",
            aes(linetype = bio)) +
  scale_linetype_manual(values = c("dashed", "solid", "dotted", "longdash",
"blank")) +
  scale_colour_brewer(palette = "Set1") +
  scale_shape_manual(values = c(0, 19, 2, 5, 46)) +
  scale_size_manual(values = c(1.0, 1.2, 1.0, 1.0, 0.6)) +
  scale_alpha_manual(values = c(0.85, 1.0, 0.85, 0.85, 0.1)) +
  xlab(expression(paste(d[E]))) +
  ylab(expression(paste(beta[sne]))) +
  scale_y_continuous(limits = c(0, 0.22)) +
  scale_x_continuous(limits = c(0, 2)) +
  theme_grey() +
  theme(panel.grid.minor = element_line(colour = NA),
        plot.title = element_text(hjust = 0, size = 10),
        legend.position = "none",
        aspect.ratio = 0.6) +
  ggtitle(expression(paste(beta[sne], " as a function of annMean")))
```

```
plt5h <- ggplot(spp_long, aes(x = distance, y = dissim)) +
  geom_blank() +
  theme(plot.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank(),
        axis.line = element_blank())
```

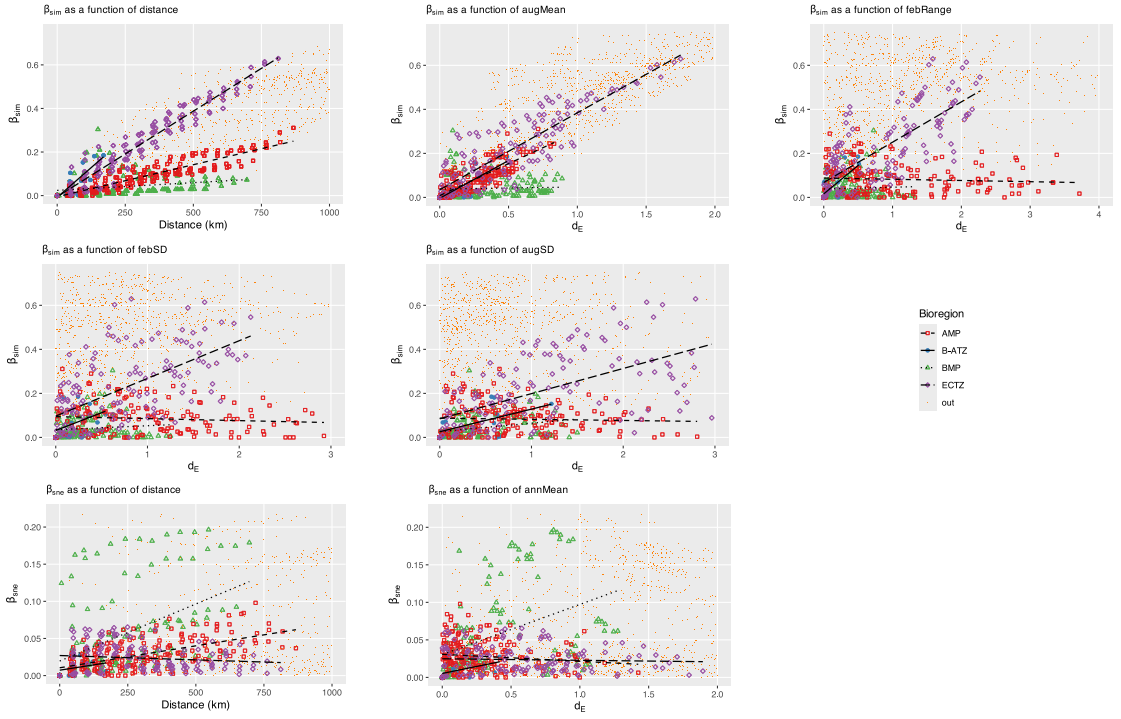
Assemble using the **cowplot** package:

```
library(cowplot)

# turn off warnings...
oldw <- getOption("warn")
options(warn = -1)

l <- get_legend(plt5a)
# pdf("Fig5.pdf", width = 9, height = 6.5)
ggdraw() +
  draw_plot(plot_grid(plt5a + theme(legend.position = 'none'), plt5b, plt5c,
                      plt5d, plt5e, l,
                      plt5f, plt5g, plt5h,
```

```
                    ncol = 3, align = 'hv'),
          width = 1.0)
```



4 References

Bibliography

Baselga A (2010) Partitioning the turnover and nestedness components of beta diversity. Global Ecology and Biogeography 19:134–143.

Bolton J, Stegenga H (2002) Seaweed species diversity in South Africa. South African Journal of Marine Science 24:9–18.

Smit AJ, Bolton JJ, Anderson RJ (2017) Seaweeds in two oceans: beta-diversity. Frontiers in Marine Science 4:404.