

# Appendix S3 - Processing CRF Model Results

*Nicholas Clark, James Kerry, Ceridwen Fraser*

## Assessing variation in species richness and functional diversity

Process the CRF model that was saved to the `Results` directory in **Appendix S2**

```
load("RCC_files/MRF.results.rda")
```

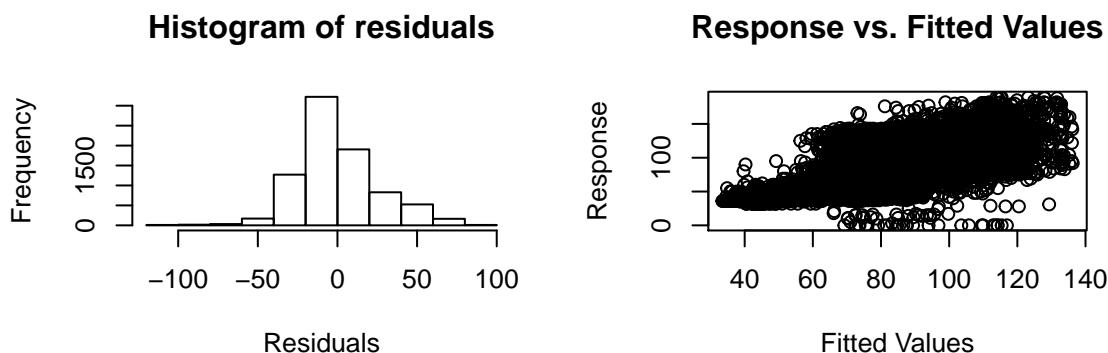
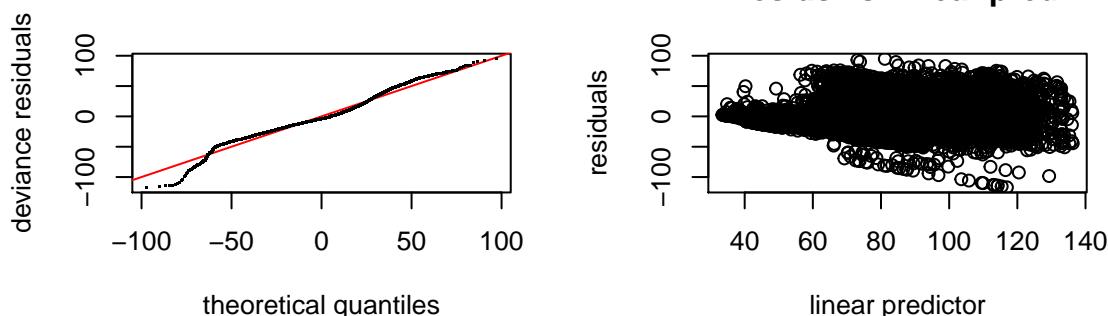
Assess how observed species richness varies across SST gradients using a spatially smoothed Generalised Additive Model (GAM) to observed richness. Load the phylogenetic and trait data that was downloaded and stored in the `Processed data` directory in **Appendix S1**

```
load("Processed data/MRF.data.rda")
```

```
source("Functions/fit_rich_gams.R")
richness.gams <- fit_rich_gams(climate_data = Obs.1980.Climate.dat,
  richness = rowSums(Medfish.mrf.dat %>%
    dplyr::select(-mean.summ.temp, -mean.wint.temp)),
  n_cores = 3)
```

Visualise diagnostic plots to ensure assumptions of Gaussian error structure are met

```
par(mfrow = c(2, 2))
gam.check(richness.gams$gam_model)
```



```
##  
## Method: fREML Optimizer: perf chol
```

```

## $grad
## [1] 6.847856e-13 -1.091571e-12 -6.608047e-13 -1.136868e-11
##
## $hess
##      [,1]      [,2]      [,3]      [,4]
## 1.97274784 -0.01160313 -0.3361680 -4.964521
## -0.01160313 1.86829950 -0.5337181 -3.981359
## -0.33616804 -0.53371811 36.8540946 -45.591929
## d -4.96452121 -3.98135890 -45.5919289 4074.500000
##
## Model rank = 198 / 198
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k'    edf k-index p-value
## s(Summer_SST) 49.00 10.93   1.01   0.69
## s(Winter_SST) 49.00  8.96   0.99   0.24
## s(Longitude, Latitude) 99.00 93.18   0.78 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Inspect the model summary

```
richness.gams$gam_summary
```

```
##  
## Family: gaussian  
## Link function: identity  
##  
## Formula:  
## richness ~ s(Summer_SST, k = 50) + s(Winter_SST, k = 50) + s(Longitude,  
##      Latitude, k = 100)  
##  
## Parametric coefficients:  
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept)  96.923     1.907   50.83 <2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Approximate significance of smooth terms:  
##             edf Ref.df      F p-value  
## s(Summer_SST) 10.929 14.19 3.715 2.46e-06 ***  
## s(Winter_SST)  8.963 11.72 21.062 < 2e-16 ***  
## s(Longitude,Latitude) 93.184 98.12 38.117 < 2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## R-sq.(adj) =  0.383  Deviance explained = 39.2%  
## fREML = 38134  Scale est. = 641.96    n = 8154
```

Inspect variogram estimates of spatial autocorrelation (nugget:sill ratio) before and after modelling. A lower ratio implies more of the variance is governed by spatial dependence. An optimal nugget:sill ratio is  $\geq 0.75$ , implying that a small proportion of variance remains governed by spatial dependence.

```
richness.gams$raw_autocorrelation
```

```
## [1] 0.4872868  
richness.gams$modelled_autocorrelation
```

```
## [1] 0.7448661
```

Estimated F-statistics and deviance explained show the model explains nearly 40 percent of variation, with winter SST variation showing a stronger correlation with observed richness. Spatial effects are important for capturing autocorrelation: model residuals exhibit far less spatial autocorrelation than the raw data.

Use the model's predictions alongside species' phylogenetic and functional distances to determine how diversity is predicted to change across SST gradients. Scaling branch lengths of both trees allows for easier direct comparisons between functional and phylogenetic diversity

```
Medfish.tree$edge.length <- Medfish.tree$edge.length/
max(Medfish.tree$edge.length)

Medfish.func.dendrogram$edge.length <- Medfish.func.dendrogram$edge.length/
max(Medfish.func.dendrogram$edge.length)
```

Calculate phylogenetic and functional diversity in sites using functions in the `PhyloMeasures` library. Remove any sites that don't have predicted species' occurrences (e.g. predicted species richness is zero)

```
richness.1980 <- rowSums(preds$Binary_predictions)
zeros <- which(richness.1980 == 0)
pdiv.1980 <- PhyloMeasures::mpd.query(Medfish.tree,
preds$Binary_predictions, standardize = F)
fdiv.1980 <- PhyloMeasures::mpd.query(Medfish.func.dendrogram,
preds$Binary_predictions, standardize = F)
```

Inspect whether there is a strong correlation between functional and phylogenetic diversities

```
cor.test(pdiv.1980[-zeros], fdiv.1980[-zeros])
```

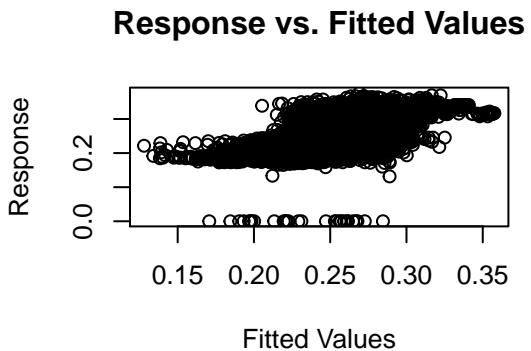
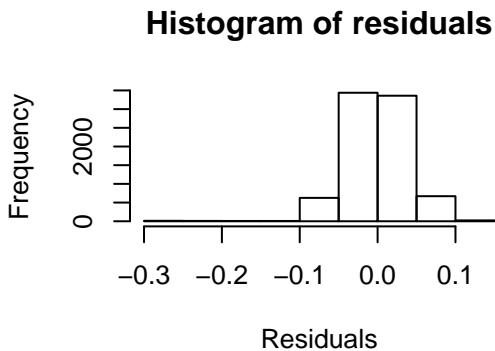
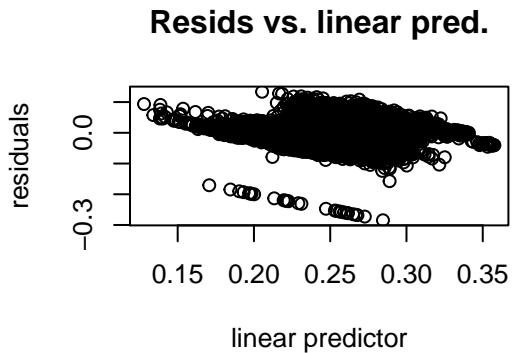
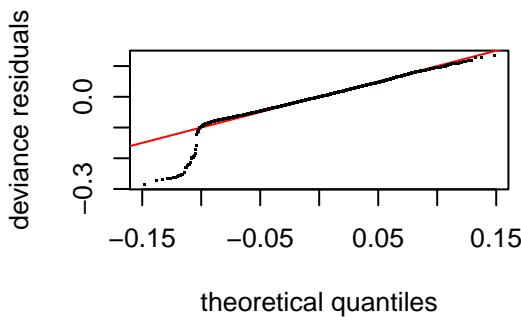
```
##
## Pearson's product-moment correlation
##
## data: pdiv.1980[-zeros] and fdiv.1980[-zeros]
## t = 117.91, df = 8124, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.7862985 0.8023425
## sample estimates:
## cor
## 0.7944591
```

A strong positive correlation. We therefore focus analyses on assessing how functional diversity of fish communities changes across SST gradients. Run the primary analysis of this section by fitting spatially smoothed GAMs to explore possible non-linear relationships between SST values and predicted functional diversity. The workflow of the `fit_div_gams` function is to fit two models, one using a thin plate spatial smooth and another using a Gaussian process spatial smooth. Model deviances are compared to determine which smooth fits the data best, and then the best-fitting model is processed to present diagnostics. Note, the function will fit models for both phylogenetic and functional diversity, but these results are likely to be similar and so we focus on the functional diversity results.

```
source("Functions/fit_div_gams.R")
diversity.gams <- fit_div_gams(climate_data = Obs.1980.Climate.dat,
phylo_div = pdiv.1980, functional_div = fdiv.1980,
n_cores = 3)
```

Visualise diagnostic plots

```
par(mfrow = c(2, 2))
gam.check(diversity.gams$Fdiv_gam_model)
```



```
##
## Method: fREML   Optimizer: perf chol
## $grad
## [1]  1.273470e-11 -5.035972e-11  4.184344e-09  6.757546e-10
##
## $hess
##      [,1]     [,2]     [,3]     [,4]
## 2.28076291 -0.01154775 -0.4758353 -7.961066
## -0.01154775 3.01948064  0.1468824 -10.280164
## -0.47583532  0.14688241 36.0575020 -45.292855
## d -7.96106627 -10.28016443 -45.2928546 4074.500000
##
## Model rank = 198 / 198
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k'    edf k-index p-value
## s(Summer_SST) 49.0 16.9    0.97   0.01 **
## s(Winter_SST)  49.0 21.6    1.00   0.36
## s(Longitude, Latitude) 99.0 92.6    0.61 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Inspect the model summary

```
diversity.gams$Fdiv_gam_summary

## 
## Family: gaussian
## Link function: identity
##
## Formula:
## fdiv ~ s(Summer_SST, k = 50) + s(Winter_SST, k = 50) + s(Longitude,
##      Latitude, bs = "gp", k = 100)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.253328   0.003098   81.78 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df     F p-value
## s(Summer_SST)    16.92  21.48 8.09 <2e-16 ***
## s(Winter_SST)     21.56  26.97 25.65 <2e-16 ***
## s(Longitude,Latitude) 92.59 95.82 37.46 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.41 Deviance explained =  42%
## fREML = -14641 Scale est. = 0.001494 n = 8154
```

Inspect nugget:sill ratios

```
diversity.gams$fddiv_raw_autocorrelation

## [1] 0.640015

diversity.gams$fddiv_modelled_autocorrelation

## [1] 0.7660662
```

The model explains about 40% of the variance in the data, and much of this is captured by variation in winter SST values. Spatial effects are again important for capturing inherent autocorrelation

## Assessing variation in co-occurrence network structure

Assess how community composition changes along SST gradients by testing whether changing summer SST or changing winter SST plays a larger role in modulating species co-occurrence networks. The spatial CRF's model coefficients reveal which temperature effects have the strongest influence on network changes. Below is a simple function to summarise these effects by calculating the total number of co-occurrence associations that are predicted to change in response to each temperature effect, as well as the average magnitude of that change

```
indirect.effect.summary = function(x) {
  total.interactions <- length(CRF$graph[which(CRF$graph != 0)])
  pos.changes <- length(which(x[[1]] > 0))
  pos.magnitude <- mean(x[[1]][which(x[[1]] > 0)])
  neg.changes <- length(which(x[[1]] < 0))
  neg.magnitude <- mean(x[[1]][which(x[[1]] < 0)])

  list(total.interactions = total.interactions,
       pos.changes = pos.changes, pos.magnitude = pos.magnitude,
       neg.changes = neg.changes, neg.magnitude = neg.magnitude)
}
```

Use this function to assess how changing summer and winter SST influence network topology. First, the influence of summer SST

```
summ.temp.summary <- indirect.effect.summary(CRF$indirect_coefs$mean.summ.temp)
summ.temp.summary
```

```
## $total.interactions
## [1] 5266
##
## $pos.changes
## [1] 2188
##
## $pos.magnitude
## [1] 0.01971121
##
## $neg.changes
## [1] 52
##
## $neg.magnitude
## [1] -0.003705374
```

And for winter SST

```
wint.temp.summary <- indirect.effect.summary(CRF$indirect_coefs$mean.wint.temp)
wint.temp.summary
```

```
## $total.interactions
## [1] 5266
##
## $pos.changes
## [1] 1840
##
## $pos.magnitude
## [1] 0.04321521
##
## $neg.changes
## [1] 76
##
## $neg.magnitude
## [1] -0.01583439
```

Across both seasons, changing SST influences network topology. Nearly two thirds of the detected pairwise co-occurrences are predicted to become more positive in warmer sites regardless of the season. However, winter SST has a larger overall impact on network topology than Summer SST (this covariate has stronger overall effects). In particular, 76 pairs of species are much less likely to co-occur in sites with high winter SST than they are in sites with lower winter SST. Integrate predicted occurrences for species along with the model's estimates of pairwise association strengths to generate model-based co-occurrence adjacency matrices for each site in the data. Here, if a species is predicted to be absent from a site, then it (along with its network edges) is removed from the network

```
pred.networks <- MRFcov::predict_MRFnetworks(data = CRF$mrf_data,
                                               prep_covariates = F, MRF_mod = CRF, cached_predictions = preds,
                                               omit_zeros = T, n_cores = 4)
```

The returned object is a `list` (one entry for each site in the data) of co-occurrence adjacency matrices (using the `igraph` package's adjacency matrix class) allowing the use of many network calculations on each object of the `list`. To assess how modularity of the networks changes across SST gradients, create a `dataframe` that contains information on species' vertical distributions in the water column. This habitat category will be used as a grouping variable to assess modularity.

```
library(dplyr)
hab.use = Medfish.func.dat %>% filter(Species %in%
                                         rownames(CRF$graph)) %>%
  dplyr::select(Species, Vertical_Distribution) %>%
  dplyr::right_join(data.frame(Species = rownames(CRF$graph)))
```

Ensure that the `Species` column is identical in order to the rownames of the CRF model elements

```
identical(hab.use$Species, rownames(CRF$graph))
```

```
## [1] TRUE
```

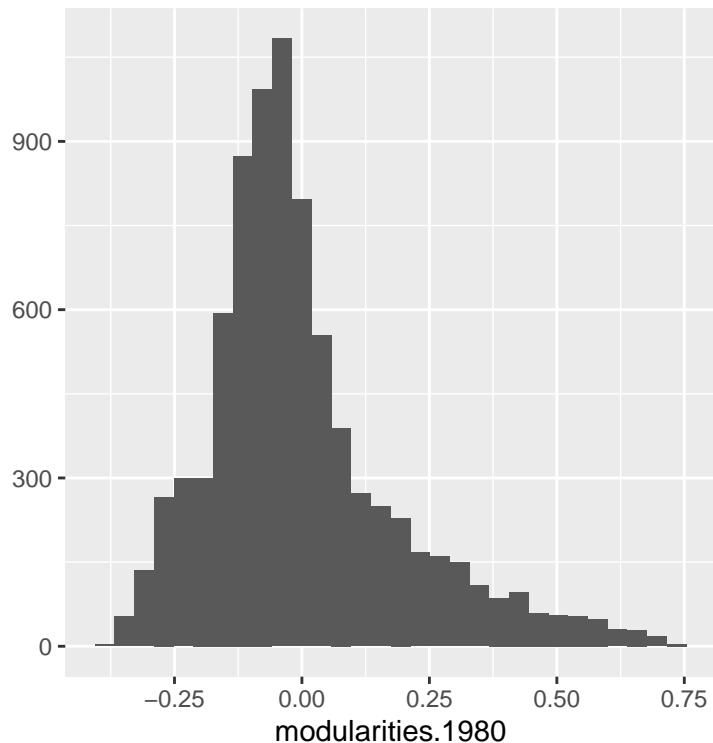
Calculate modularity using `igraph`'s `modularity` function by grouping species based on their vertical distributions. Extract the residuals from a linear regression between modularity and species richness, as the modularity metric is known to be biased by the number of nodes present in the network

```
modularities.1980 <- unlist(lapply(seq_along(pred.networks),
  function(x) {
    igraph::modularity(pred.networks[[x]],
      membership = as.factor(hab.use$Vertical_Distribution))
  }))
modularities.1980 <- resid(lm(modularities.1980 ~
  richness.1980))
```

Explore this vector's histogram to see the spread of the values and ensure there are no NAs

```
ggplot2::qplot(modularities.1980)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

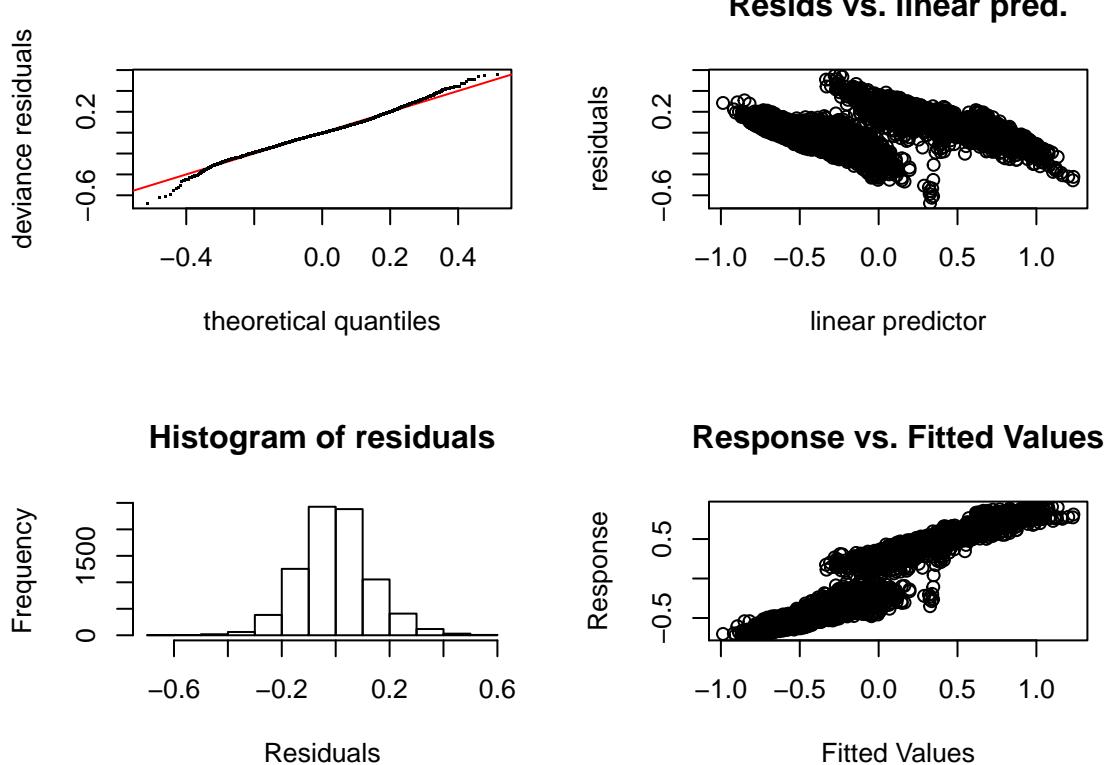


Fit GAMs using modularity as the response. The `fit_mod_gams` function is essentially identical to the process used for fitting diversity models above, except that initial model testing revealed that a cube root transformation of modularity was necessary to ensure Gaussian assumptions were met

```
source("Functions/fit_mod_gams.R")
modularity.gams <- fit_mod_gams(climate_data = Obs.1980.Climate.dat,
  modularity_data = modularities.1980,
  n_cores = 3)
```

Check the model's diagnostic plots to ensure assumptions have been met

```
par(mfrow = c(2, 2))
gam.check(modularity.gams$mod_gam_model)
```



```
##
## Method: fREML   Optimizer: perf chol
## $grad
## [1]  1.832632e-10 -4.041176e-10 -1.285436e-09 -7.496510e-09
##
## $hess
##           [,1]          [,2]          [,3]          [,4]
## 8.04793590  0.01582157 -0.1746352 -14.27617
## 0.01582157 18.69044262 -0.1243884 -20.03898
## -0.17463522 -0.12438838 42.2453162 -47.33229
## d -14.27616910 -20.03898285 -47.3322934 4074.50000
##
## Model rank = 198 / 198
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'    edf k-index p-value
## s(Summer_SST)      49.0  29.6    0.96  0.005 **
## s(Winter_SST)      49.0  41.1    1.00  0.425
## s(Longitude, Latitude) 99.0  96.7    0.32 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

modularity.gams$mod_gam_summary

##
## Family: gaussian
## Link function: identity
##
## Formula:
## modularity ~ s(Summer_SST, k = 50) + s(Winter_SST, k = 50) +
##           s(Longitude, Latitude, bs = "gp", k = 100)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.18187   0.01133 -16.05 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F p-value
## s(Summer_SST)    29.55  35.60 49.29 <2e-16 ***
## s(Winter_SST)     41.08  45.79 78.83 <2e-16 ***
## s(Longitude,Latitude) 96.66  97.57 539.12 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.925 Deviance explained = 92.7%
## fREML = -4298 Scale est. = 0.018011 n = 8154
modularity.gams$raw_autocorrelation

## [1] 0.06722272
modularity.gams$modelled_autocorrelation

## [1] 0.8198578

```

## Assessing how projected warming SST is expected to influence fish community structure

Next, draw inferences about how communities may change under future climate scenarios. First generate predictions of species' distributions using multivariate generalized boosted regression trees (also known as gradient boosted machines; GBM), which effectively simulate communities from CRFs posterior predictions. Climate and coordinate values will be used as predictors in the boosted regression trees

```
clim.dat <- Obs.1980.Climate.dat %>% dplyr::group_by(Latitude,
  Longitude) %>% dplyr::mutate(Summer_SST = mean(July,
  August, September), Winter_SST = mean(January,
  February, March)) %>% dplyr::ungroup() %>%
  dplyr::select(Latitude, Longitude, Summer_SST,
  Winter_SST)

future.clim.dat <- Proj.2040.2059.Climate.dat %>%
  dplyr::group_by(Latitude, Longitude) %>%
  dplyr::mutate(Summer_SST = mean(July,
  August, September), Winter_SST = mean(January,
  February, March)) %>% dplyr::ungroup() %>%
  dplyr::select(Latitude, Longitude, Summer_SST,
  Winter_SST)

clim_features <- as.matrix(clim.dat)
sp_outcomes <- as.matrix(preds$Probability_predictions)
```

Apply five-fold cross validation to estimate the multivariate GBM. Note, this step is very memory-heavy. Do not attempt on a personal laptop.

```
all_clim_dat <- rbind(clim_features, future.clim.dat)
scaled_clim_dat <- scale(all_clim_dat)
clim_features_sc <- scaled_clim_dat[1:nrow(clim_features),
  ]
future_clim_sc <- scaled_clim_dat[(nrow(clim_features) +
  1):nrow(all_clim_dat), ]

library(mvtnet)
out <- mvtnet(Y = sp_outcomes, X = clim_features_sc,
  n.trees = 500, shrinkage = 0.05, interaction.depth = 3,
  cv.folds = 5, compress = T, keep.data = FALSE)

hist_preds <- predict(out, newdata = clim_features_sc)
future_preds <- predict(out, newdata = as.matrix(future_clim_sc))
hist_occurs <- hist_preds
hist_occurs[hist_occurs < 0.5] <- 0
hist_occurs[hist_occurs >= 0.5] <- 1

future_occurs <- future_preds
future_occurs[future_occurs < 0.5] <- 0
future_occurs[future_occurs >= 0.5] <- 1

richness.1980_sim <- rowSums(hist_occurs)
richness.2040_sim <- rowSums(future_occurs)
```

Calculate predicted functional diversity and modularity metrics for historical and future scenarios

```
fdiv.1980_sim <- PhyloMeasures::mpd.query(Medfish.func.dendrogram,
    hist_occurs, standardize = F)
fdiv.2040_sim <- PhyloMeasures::mpd.query(Medfish.func.dendrogram,
    future_occurs, standardize = F)
hist_sim_prep <- MRFcov::prep_MRF_covariates(Obs.1980.Climate.dat %>%
    dplyr::group_by(Latitude, Longitude) %>%
    dplyr::mutate(mean.summ.temp = mean(July,
        August, September), mean.wint.temp = mean(January,
        February, March)) %>% dplyr::ungroup() %>%
    dplyr::select(mean.summ.temp, mean.wint.temp) %>%
    dplyr::bind_cols(Medfish.mrf.dat %>%
        dplyr::select(-mean.summ.temp, -mean.wint.temp)) %>%
    dplyr::select(-matches(paste(colnames(hist_occurs),
        collapse = "|")))) %>% dplyr::bind_cols(data.frame(hist_occurs)) %>%
    dplyr::select(matches(paste(colnames(Medfish.mrf.dat),
        collapse = "|")))) %>% dplyr::select(-mean.summ.temp,
    -mean.wint.temp, mean.summ.temp, mean.wint.temp),
    n_nodes = length(mrf.species)) %>% dplyr::bind_cols(data.frame(CRF$mrf_data) %>%
    dplyr::select(dplyr::contains("Spatial")))
hist_sim_prep <- hist_sim_prep[, colnames(CRF$mrf_data)]

hist_sim_networks <- MRFcov::predict_MRFnetworks(data = hist_sim_prep,
    prep_covariates = F, MRF_mod = CRF, omit_zeros = T,
    n_cores = 3)

modularities.1980_sim <- unlist(lapply(seq_along(hist_sim_networks),
    function(x) {
        igraph::modularity(hist_sim_networks[[x]],
            membership = as.factor(hab.use$Vertical_Distribution))
    }))
modularities.1980_sim <- resid(lm(modularities.1980_sim ~
richness.1980))

future_sim_prep <- MRFcov::prep_MRF_covariates(Proj.2040.2059.Climate.dat %>%
    dplyr::group_by(Latitude, Longitude) %>%
    dplyr::mutate(mean.summ.temp = mean(July,
        August, September), mean.wint.temp = mean(January,
        February, March)) %>% dplyr::ungroup() %>%
    dplyr::select(mean.summ.temp, mean.wint.temp) %>%
    dplyr::bind_cols(Medfish.mrf.dat %>%
        dplyr::select(-mean.summ.temp, -mean.wint.temp)) %>%
    dplyr::select(-matches(paste(colnames(future_occurs),
        collapse = "|")))) %>% dplyr::bind_cols(data.frame(future_occurs)) %>%
    dplyr::select(matches(paste(colnames(Medfish.mrf.dat),
        collapse = "|")))) %>% dplyr::select(-mean.summ.temp,
    -mean.wint.temp, mean.summ.temp, mean.wint.temp),
    n_nodes = length(mrf.species)) %>% dplyr::bind_cols(data.frame(CRF$mrf_data) %>%
    dplyr::select(dplyr::contains("Spatial")))
future_sim_prep <- future_sim_prep[, colnames(CRF$mrf_data)]
future_sim_networks <- MRFcov::predict_MRFnetworks(data = future_sim_prep,
    prep_covariates = F, MRF_mod = CRF, omit_zeros = T,
```

```
n_cores = 3)
modularities.2040_sim <- unlist(lapply(seq_along(future_sim_networks),
  function(x) {
    igraph::modularity(future_sim_networks[[x]],
      membership = as.factor(hab.use$Vertical_Distribution))
  }))
modularities.2040_sim <- resid(lm(modularities.2040_sim ~
  richness.2040_sim))
```

Calculating species' projected proportional change in geographical range

```
obs_range <- colSums(hist_occurs)
fut_range <- colSums(future_occurs)
prop_change <- (fut_range/(obs_range)) -
  1

prop_change <- data.frame(Species = colnames(preds$Probability_predictions),
  Change = prop_change) %>% dplyr::left_join(hab.use) %>%
  dplyr::mutate(Change = ifelse(is.na(Change),
    1, Change)) %>% dplyr::mutate(Change = ifelse(is.infinite(Change),
    1, Change)) %>% dplyr::arrange(Change) %>%
  dplyr::mutate(Species = factor(Species,
    levels = Species))
```

What proportion of species are predicted to lose at least some of their range?

```
(prop_change %>% dplyr::filter(Change < 0) %>%
  nrow())/nrow(prop_change)
```

```
## [1] 0.5674419
```

And of these losses in range, what proportional losses are predicted?

```
prop_change %>% dplyr::filter(Change < 0) %>%
  dplyr::summarise(min = min(Change), `0.25` = quantile(Change,
  0.25), mean = mean(Change), `0.75` = quantile(Change,
  0.75), max = max(Change))

##   min      .25      mean      0.75      max
## 1 -1 -0.5953987 -0.4458879 -0.3070529 -0.0003575898
```

Create a table to show projected range changes for species of economic and conservation importance, including non-indigenous species with the greatest potential impacts. This list of species was sourced from the General Fisheries Commission for the Mediterranean (GCGM; <http://www.fao.org/gfcm/data/dcrc/en/>)

```
key_species <- readxl::read_xlsx("./Fishing pressures/GFCM_species_of_importance.xlsx")
key_species_hists <- hist_occurs[, which(colnames(hist_occurs) %in%
  key_species$Species)]
key_species_futures <- future_occurs[, which(colnames(future_occurs) %in%
  key_species$Species)]
```

Function to summarise latitudinal ranges of key species in both time periods

```
summarise_lats = function(x) {
  quantile(coords$Latitude[which(x == 1)],
    probs = c(0, 0.25, 0.5, 0.75, 1))
}

hist_lats <- data.frame(t(apply(key_species_hists,
  2, summarise_lats)))
hist_lats$Species <- colnames(key_species_hists)
hist_lats$Time <- 1980

future_lats <- data.frame(t(apply(key_species_futures,
  2, summarise_lats)))
future_lats$Species <- colnames(key_species_futures)
future_lats$Time <- 2040
key_species_lats <- rbind(hist_lats, future_lats) %>%
  dplyr::group_by(Species) %>% dplyr::filter(!any(is.na(X0.))) %>%
  dplyr::ungroup() %>% dplyr::mutate(Species = gsub("_",
  " ", Species))
rm(hist_lats, future_lats)
```

Repeat by summarising longitudinal ranges of key species in both time periods

```
summarise_longs = function(x) {
  quantile(coords$Longitude[which(x ==
    1)], probs = c(0, 0.25, 0.5, 0.75,
    1))
}

hist_longs <- data.frame(t(apply(key_species_hists,
  2, summarise_longs)))
hist_longs$Species <- colnames(key_species_hists)
hist_longs$Time <- 1980

future_longs <- data.frame(t(apply(key_species_futures,
  2, summarise_longs)))
future_longs$Species <- colnames(key_species_futures)
future_longs$Time <- 2040
key_species_longs <- rbind(hist_longs, future_longs) %>%
  dplyr::group_by(Species) %>% dplyr::filter(!any(is.na(X0.))) %>%
  dplyr::ungroup() %>% dplyr::mutate(Species = gsub("_",
  " ", Species))
rm(hist_longs, future_longs)
```

Summarise proportional range size changes for key species

```
summarise_ranges = function(x) {
  sum(x)/nrow(key_species_hists)
}

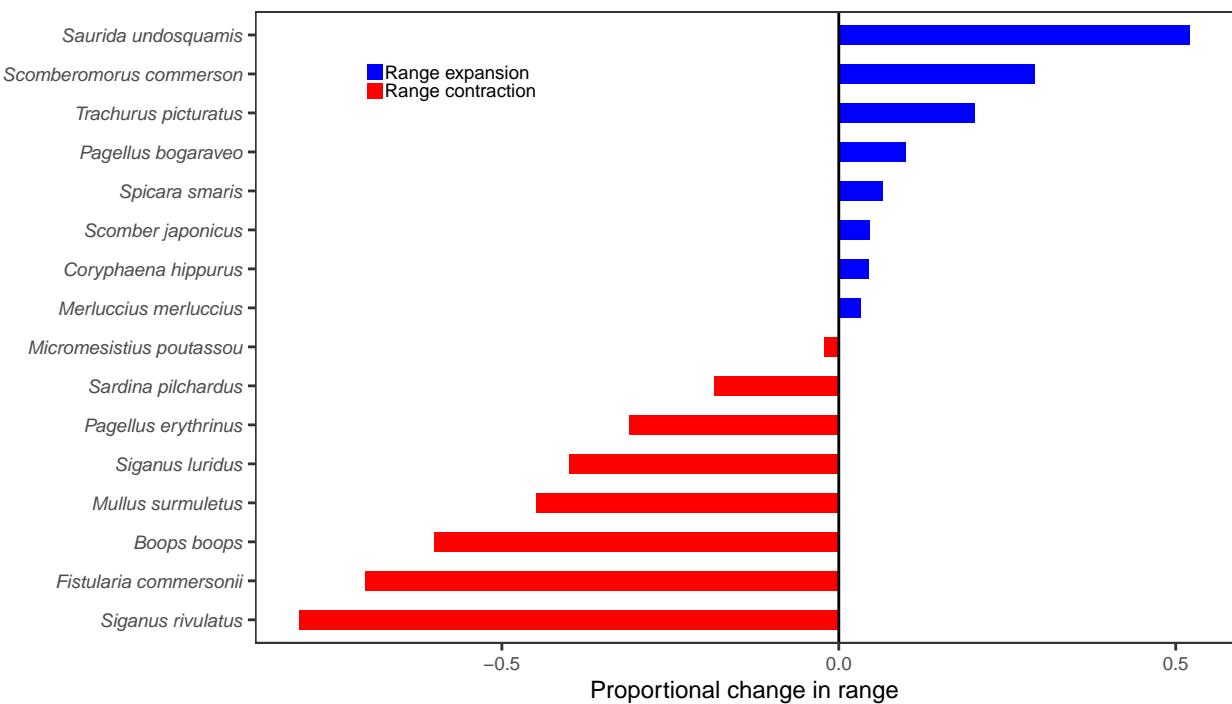
hist_ranges <- data.frame(apply(key_species_hists,
  2, summarise_ranges))
colnames(hist_ranges) <- "Range_hist"
hist_ranges$Species <- colnames(key_species_hists)

future_ranges <- data.frame(apply(key_species_futures,
  2, summarise_ranges))
colnames(future_ranges) <- "Range_future"
future_ranges$Species <- colnames(key_species_futures)
key_species_ranges <- hist_ranges %>% dplyr::left_join(future_ranges) %>%
  dplyr::filter(Range_hist != 0 & Range_future !=
    0) %>% dplyr::rowwise() %>% dplyr::mutate(Range_change = (Range_future/Range_hist) -
  1) %>% dplyr::mutate(Species = gsub("_",
  " ", Species)) %>% dplyr::mutate(Direction = ifelse(Range_change <
  0, "loss", "gain"))

## Joining, by = "Species"
rm(hist_ranges, future_ranges)
```

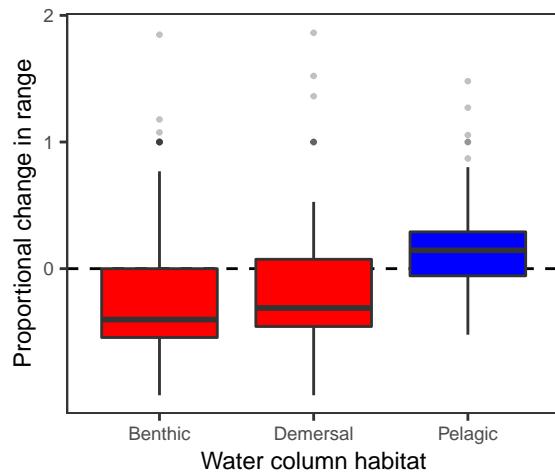
Plot of key species' proportional changes in geographical ranges (part a of **Figure 3**)

```
key_species_ranges <- key_species_ranges[order(key_species_ranges$Range_change),
  ]
key_species_ranges$Species <- factor(key_species_ranges$Species,
  levels = key_species_ranges$Species)
ggplot(key_species_ranges, aes(x = Species,
  y = Range_change)) + theme_bw() + geom_bar(stat = "identity",
  aes(fill = Direction), width = 0.5) +
  scale_fill_manual(name = "", labels = c("Range expansion",
  "Range contraction"), values = c(gain = "blue",
  loss = "red")) + geom_hline(yintercept = 0) +
  labs(y = "Proportional change in range",
  x = "") + theme(axis.text.y = element_text(face = "italic")) +
  coord_flip() + theme(legend.position = c(0.2,
  0.89), legend.text = element_text(size = 7),
  legend.title = element_text(size = 0.1,
  hjust = 0.5), legend.key.size = unit(0.25,
  "cm"), legend.background = element_rect(fill = "white",
  colour = "white", size = 0.01), legend.box.background = element_blank(),
  panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
  axis.text = element_text(size = 7), axis.title = element_text(size = 9),
  plot.margin = margin(l = -9, r = 4, t = 10,
  b = 0.5))
```



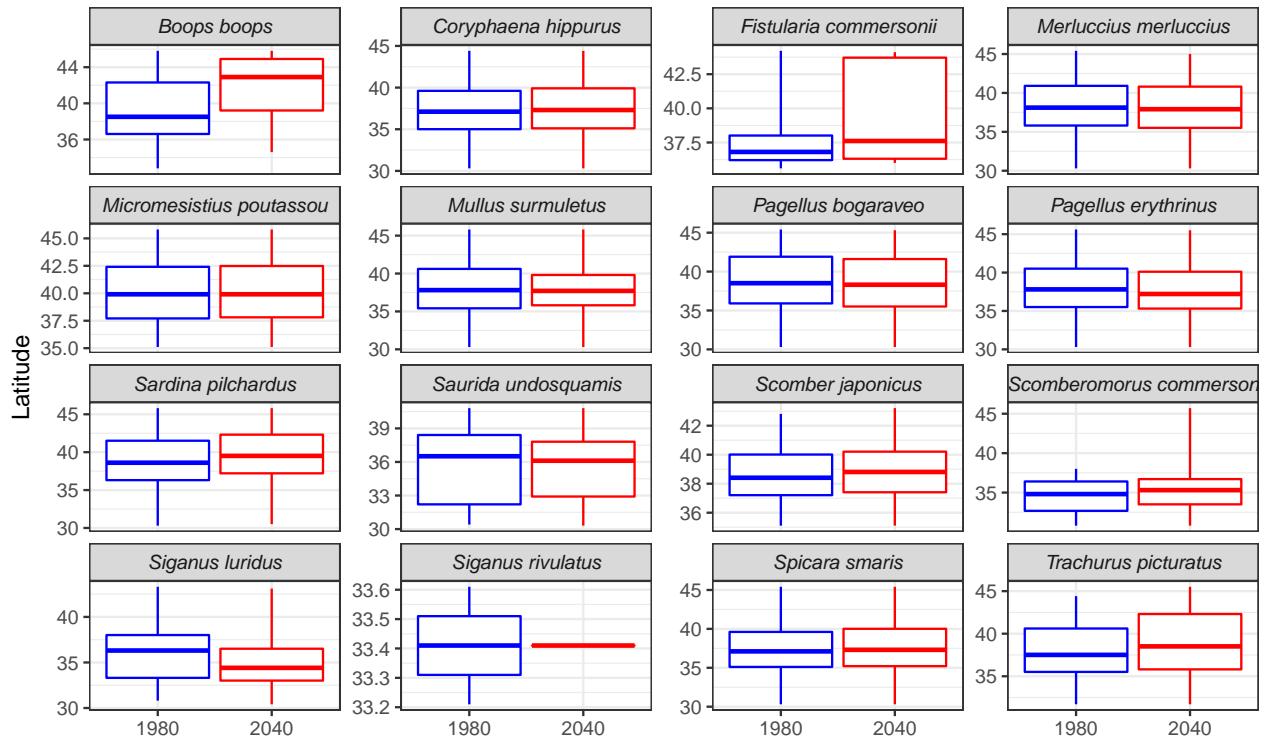
Plot projected range changes by habitat for all species (part b of **Figure 3**)

```
ggplot(prop_change, aes(Vertical_Distribution,
  Change)) + theme_bw() + geom_hline(yintercept = 0,
  size = 0.5, col = "black", linetype = "dashed") +
  geom_boxplot(fill = c("red", "red", "blue"),
  outlier.alpha = 0.3, outlier.size = 0.5) +
  scale_y_continuous(limits = quantile(prop_change$Change,
  c(0, 0.95)), breaks = c(0, 1, 2,
  3)) + theme(legend.justification = c(0,
  1), legend.position = c(0.01, 0.99),
  legend.key.size = unit(0.1, "cm"), legend.title = element_text(size = 9),
  legend.text = element_text(size = 9),
  legend.key = element_rect(fill = NA),
  legend.background = element_rect(fill = NA,
  colour = NA, size = 0.01), legend.box.background = element_rect(colour = NA,
  size = 0.01), panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), axis.text = element_text(size = 7),
  axis.title = element_text(size = 9)) +
  ylab("Proportional change in range") +
  xlab("Water column habitat")
```



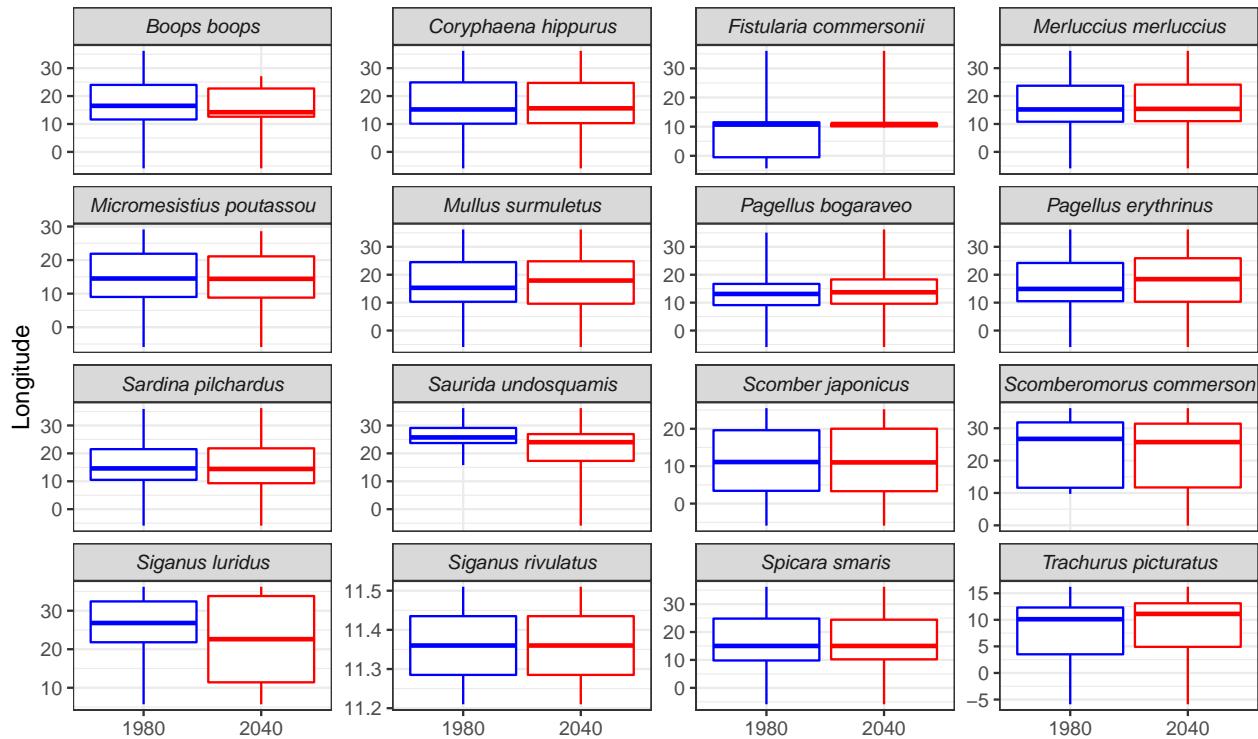
Boxplots of key species' latitudinal ranges for inclusion in **Supplementary Material**

```
ggplot(data = key_species_lats, aes(x = as.factor(Time),
  colour = as.factor(Time), group = Time)) +
  theme_bw() + geom_boxplot(aes(ymin = X0.,
  lower = X25., middle = X50., upper = X75.,
  ymax = X100.), stat = "identity") + scale_colour_manual(values = c("blue",
  "red")) + facet_wrap(~Species, scales = "free_y") +
  labs(x = "", y = "Latitude") + theme(legend.position = "none",
  strip.text = element_text(face = "italic"))
```



Boxplots of key species' longitudinal ranges for inclusion in **Supplementary Material**

```
ggplot(data = key_species_longs, aes(x = as.factor(Time),
  colour = as.factor(Time), group = Time)) +
  theme_bw() + geom_boxplot(aes(ymin = X0.,
  lower = X25., middle = X50., upper = X75.,
  ymax = X100.), stat = "identity") + scale_colour_manual(values = c("blue",
  "red")) + facet_wrap(~Species, scales = "free_y") +
  labs(x = "", y = "Longitude") + theme(legend.position = "none",
  strip.text = element_text(face = "italic"))
```



Summary of the full distribution of projected changes in species richness for each site

```
summary(richness.2040_sim - richness.1980_sim)

##      Min. 1st Qu. Median    Mean 3rd Qu.    Max.
## -49.000 -10.000 -3.000 -2.256   5.000  38.000
```

What proportion of sites will be less species rich by 2040?

```
length(which((richness.2040_sim - richness.1980_sim) <
             0))/length(richness.1980_sim)
```

```
## [1] 0.5909983
```

Maps of projected SST changes to create parts **a** and **b** of **Figure 1**.

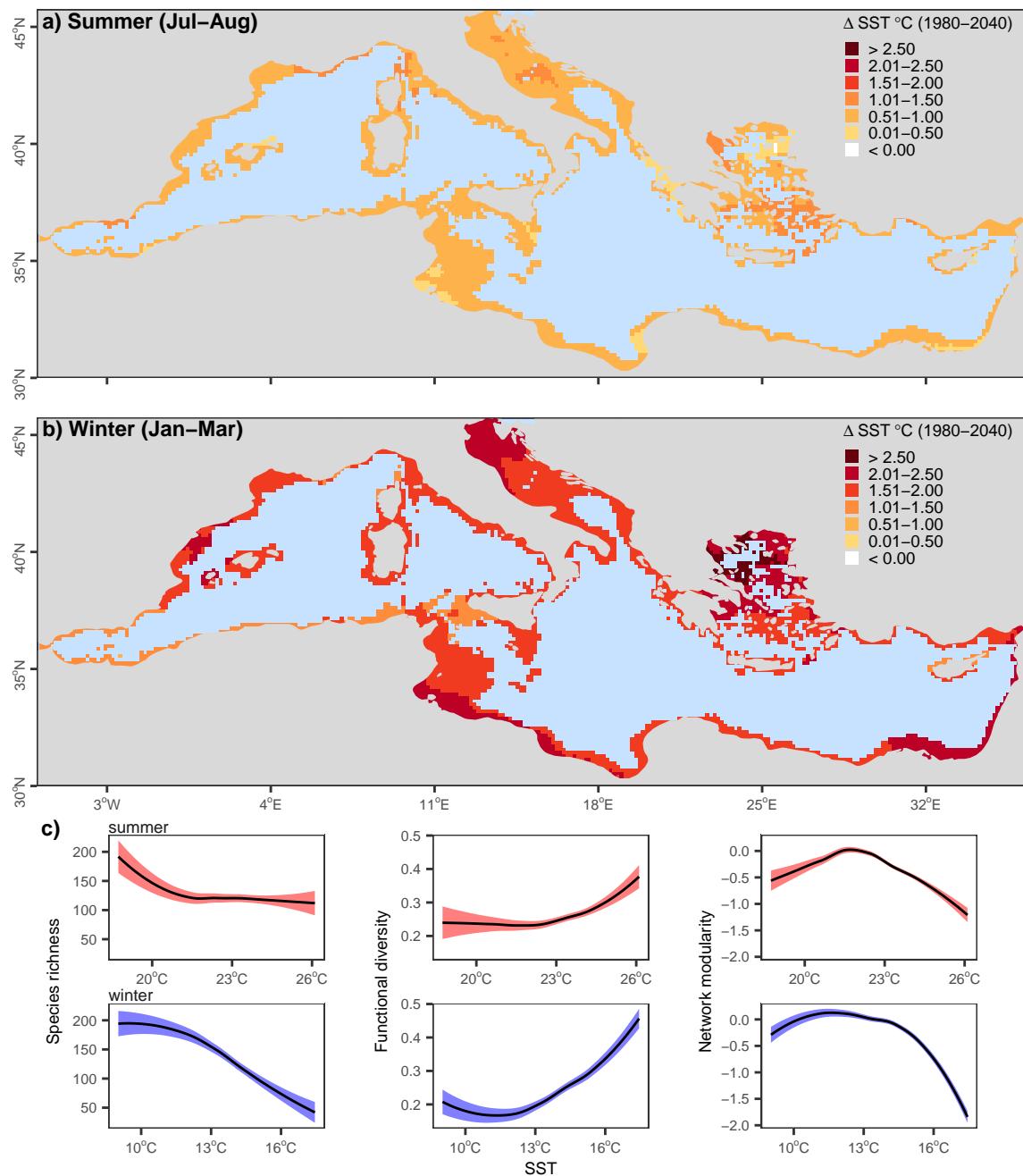
```
source("Functions/map_sst.R")
summer_map <- map_sst(obs_climate_dat = Obs.1980.Climate.dat,
                      proj_climate_dat = Proj.2040.2059.Climate.dat,
                      summer = TRUE)
winter_map <- map_sst(obs_climate_dat = Obs.1980.Climate.dat,
                      proj_climate_dat = Proj.2040.2059.Climate.dat,
                      summer = FALSE)
```

Plot the gam model results for each outcome variable to complete **Figure 1**

```
source("Functions/generate_gam_plot.R")
rich_plot <- generate_gam_plot(smooths = richness.gams$gam_effects$smooths,
                                ylabel = "Species richness", y_decimal = FALSE) +
  xlab("") + theme(plot.margin = margin(0,
  0, 0, 0.5, "cm"))
div_plot <- generate_gam_plot(smooths = diversity.gams$Fdiv_gam_effects$smooths,
                               ylabel = "Functional diversity", y_decimal = TRUE,
                               season_labels = F) + theme(plot.margin = margin(0,
  0.4, 0, 0.4, "cm"))
mod_plot <- generate_gam_plot(smooths = modularity.gams$mod_gam_effects$smooths,
                               ylabel = "Network modularity", y_decimal = TRUE,
                               season_labels = F) + xlab("") + theme(plot.margin = margin(0,
  0.5, 0, 0, "cm"))
```

**Figure 1**

```
cowplot::plot_grid(cowplot::plot_grid(summer_map +
  scale_x_continuous(labels = rep("", 6),
  breaks = seq(-3, 35, 7)), winter_map,
  ncol = 1, align = "hv", rel_heights = c(1,
  1), rel_widths = c(1, 1), labels = c("a) Summer (Jul-Aug)",
  "b) Winter (Jan-Mar)"), label_size = 9,
  hjust = c(-0.21, -0.225), vjust = c(1.65,
  1.65)), cowplot::plot_grid(rich_plot,
  div_plot, mod_plot, ncol = 3, align = "h"),
  ncol = 1, labels = c("", "c")), rel_heights = c(1,
  0.44), label_size = 9, hjust = -2.3,
  vjust = 1.1, align = "h", axis = "r")
```



Mapping projected changes for the simulated communities produces part of **Figure 2**

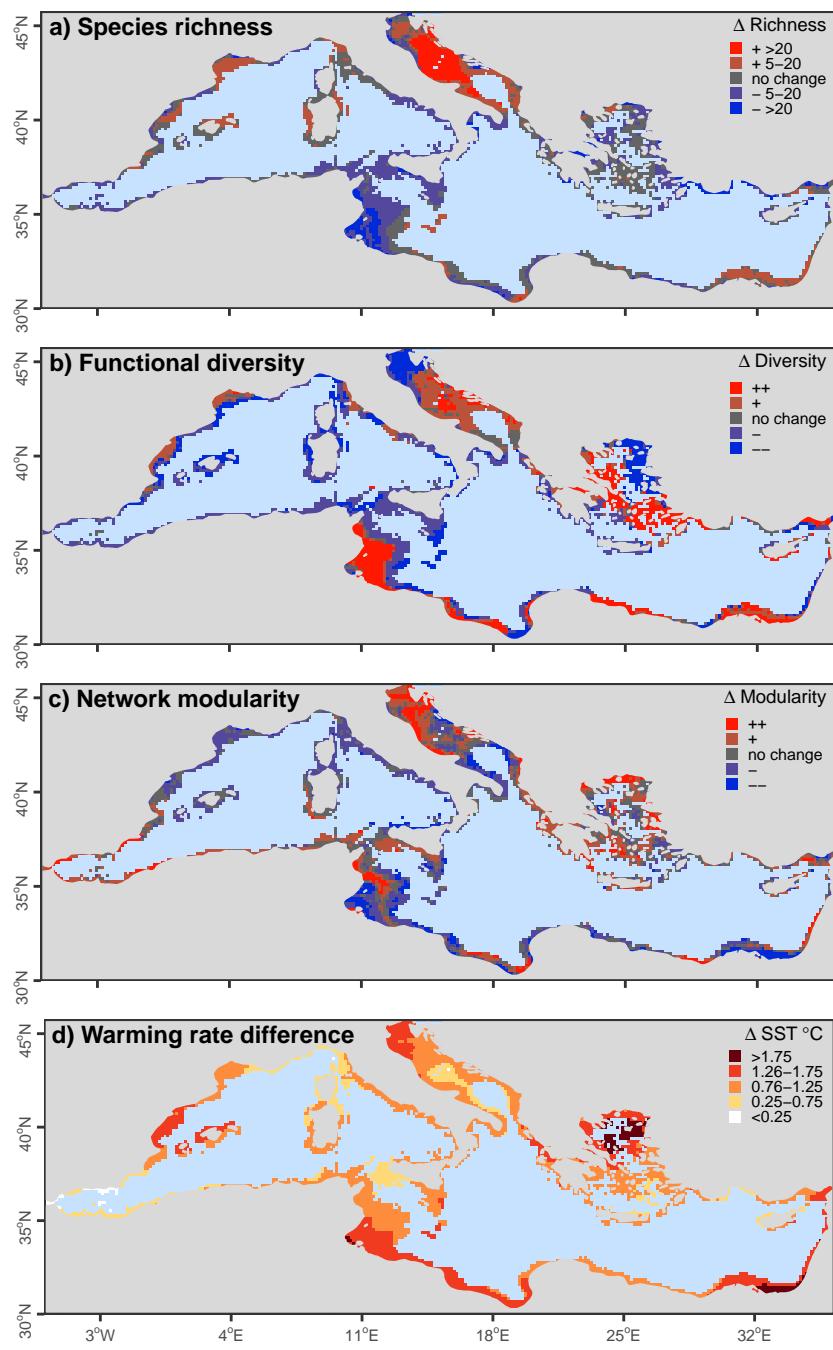
```
source("Functions/plot_composite_map.R")
div_map = plot_composite_map(coords, future_obs = fdiv.2040_sim,
    past_obs = fdiv.1980_sim, metric = "diversity")
rich_map <- plot_composite_map(coords, future_obs = richness.2040_sim,
    past_obs = richness.1980_sim, metric = "richness")
mod_map <- plot_composite_map(coords, future_obs = modularities.2040_sim,
    past_obs = modularities.1980_sim, metric = "modularity")
```

The other part of **Figure 2** shows differences in projected warming for Winter vs Summer

```
SST_changes <- Obs.1980.Climate.dat %>% dplyr::group_by(Latitude,
Longitude) %>% dplyr::mutate(obs_Summer_SST = mean(July,
August, September), obs_Winter_SST = mean(January,
February, March)) %>% dplyr::ungroup() %>%
dplyr::select(Latitude, Longitude, obs_Summer_SST,
obs_Winter_SST) %>% dplyr::left_join(Proj.2040.2059.Climate.dat %>%
dplyr::group_by(Latitude, Longitude) %>%
dplyr::mutate(proj_Summer_SST = mean(July,
August, September), proj_Winter_SST = mean(January,
February, March)) %>% dplyr::ungroup() %>%
dplyr::select(Latitude, Longitude, proj_Summer_SST,
proj_Winter_SST)) %>% dplyr::mutate(change_Summer_SST = proj_Summer_SST -
obs_Summer_SST, change_Winter_SST = proj_Winter_SST -
obs_Winter_SST)
temp_map <- plot_composite_map(coords, future_obs = SST_changes$proj_Winter_SST -
SST_changes$obs_Winter_SST, past_obs = SST_changes$proj_Summer_SST -
SST_changes$obs_Summer_SST, metric = "temp")
```

**Figure 2**

```
cowplot::plot_grid(rich_map + theme(plot.margin = margin(t = 1,
r = 0.5, -10, -15.5)) + scale_x_continuous(labels = rep("", 6),
breaks = seq(-3, 35, 7)), div_map +
theme(plot.margin = margin(1, 0.5, -10,
-15.5)) + scale_x_continuous(labels = rep("", 6),
breaks = seq(-3, 35, 7)), mod_map +
theme(plot.margin = margin(1, 0.5, -10,
-15.5)) + scale_x_continuous(labels = rep("", 6),
breaks = seq(-3, 35, 7)), temp_map +
theme(plot.margin = margin(1, 0.5, -10,
-15.5)), ncol = 1, labels = c("a) Species richness",
"b) Functional diversity", "c) Network modularity",
"d) Warming rate difference"), label_size = 9,
hjust = c(-0.24, -0.21, -0.21, -0.185),
vjust = 1.5)
```

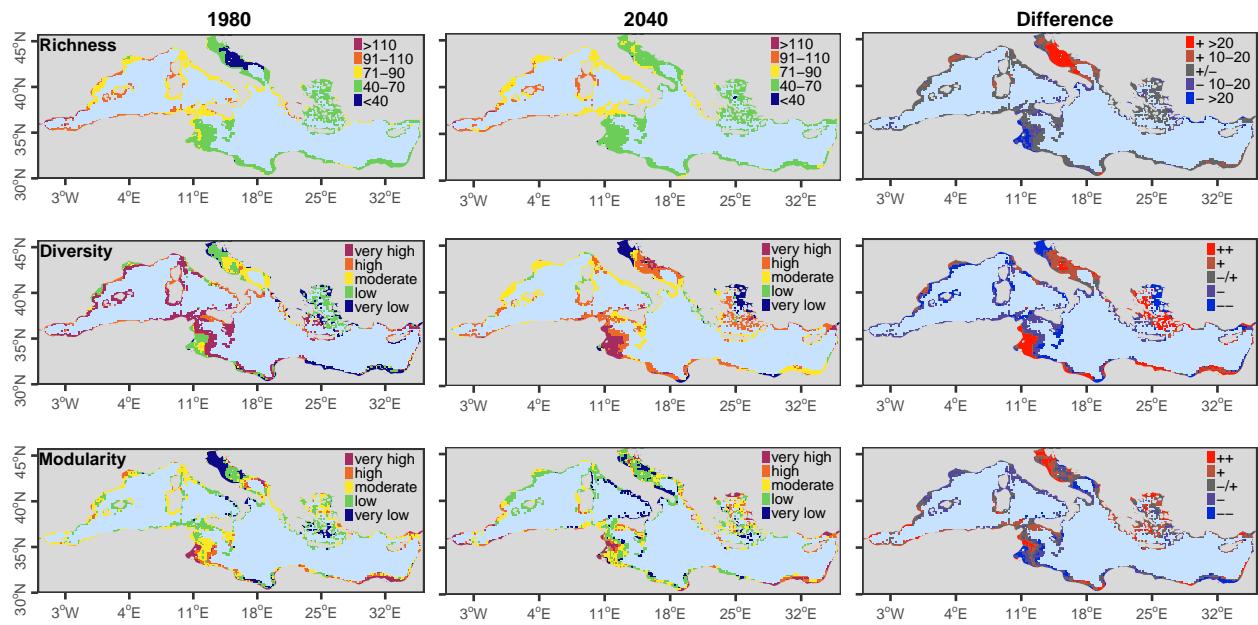


Plot the historical and projected metrics, as well as projected differences, to include in **Supplementary Material**

```
source("Functions/plot_composite_map_supp.R")
div_map = plot_composite_map_supp(coords,
  future_obs = fdiv.2040_sim, past_obs = fdiv.1980_sim,
  metric = "diversity")
rich_map <- plot_composite_map_supp(coords,
  future_obs = richness.2040_sim, past_obs = richness.1980_sim,
  metric = "richness")
mod_map <- plot_composite_map_supp(coords,
  future_obs = modularities.2040_sim, past_obs = modularities.1980_sim,
  metric = "modularity")
```

Visualise the maps

```
cowplot::plot_grid(rich_map[[1]] + theme(plot.margin = margin(3.5,
  0.5, -16, -12.75)) + annotate("text",
  x = -1.6, y = 44.65, label = "Richness",
  size = 2.5, fontface = 2), rich_map[[2]] +
  theme(plot.margin = margin(3.5, 0.5,
  -16.5, -15.5)) + scale_y_continuous(labels = rep("", 4),
  breaks = seq(30, 49, 5)), rich_map[[3]] +
  theme(plot.margin = margin(2.5, 0.75,
  -16.5, -15.5)) + scale_y_continuous(labels = rep("", 4),
  breaks = seq(30, 49, 5)), div_map[[1]] +
  annotate("text", x = -1.8, y = 44.65,
  label = "Diversity", size = 2.5,
  fontface = 2) + theme(plot.margin = margin(1,
  0.5, -16.5, -12.75)), div_map[[2]] +
  scale_y_continuous(labels = rep("", 4),
  breaks = seq(30, 49, 5)) + theme(plot.margin = margin(1,
  0.5, -16.5, -15.5)), div_map[[3]] + scale_y_continuous(labels = rep("", 4),
  breaks = seq(30, 49, 5)) + theme(plot.margin = margin(1,
  0.75, -16.5, -15.5)), mod_map[[1]] +
  annotate("text", x = -1.1, y = 44.65,
  label = "Modularity", size = 2.5,
  fontface = 2) + theme(plot.margin = margin(1,
  0.5, -16.5, -12.75)), mod_map[[2]] +
  scale_y_continuous(labels = rep("", 4),
  breaks = seq(30, 49, 5)) + theme(plot.margin = margin(1,
  0.5, -16.5, -15.5)), mod_map[[3]] + scale_y_continuous(labels = rep("", 4),
  breaks = seq(30, 49, 5)) + theme(plot.margin = margin(1,
  0.75, -16.5, -15.5)), ncol = 3, labels = c("1980",
  "2040", "Difference", "", "", "", "",
  "", ""), label_size = 8, hjust = c(-4.5,
  -4.5, -1.8, -4, -4, -4, -4, -4),
  vjust = 1.25)
```



Characterise sites into three categories: crosses the 12 - 13 degree threshold threshold, stays above the threshold and stays below

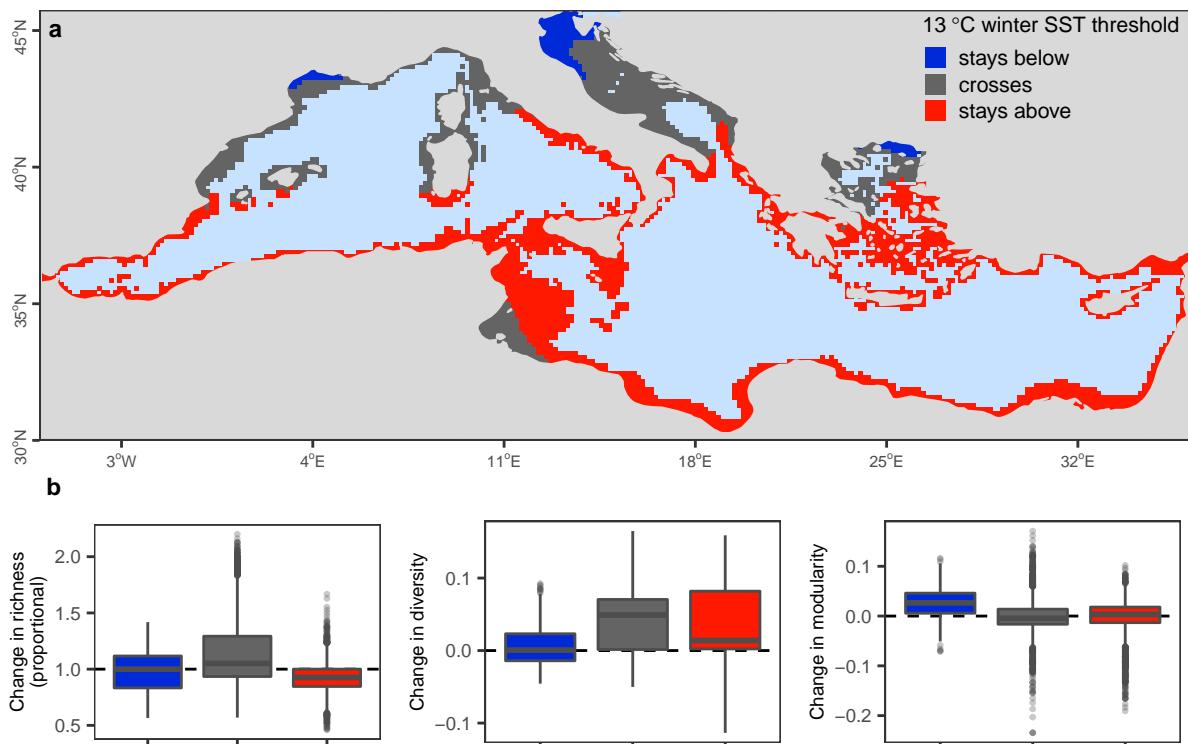
```
SST_changes <- Obs.1980.Climate.dat %>% dplyr::group_by(Latitude,
Longitude) %>% dplyr::mutate(obs_Summer_SST = mean(July,
August, September), obs_Winter_SST = mean(January,
February, March)) %>% dplyr::ungroup() %>%
dplyr::select(Latitude, Longitude, obs_Summer_SST,
obs_Winter_SST) %>% dplyr::left_join(Proj.2040.2059.Climate.dat %>%
dplyr::group_by(Latitude, Longitude) %>%
dplyr::mutate(proj_Summer_SST = mean(July,
August, September), proj_Winter_SST = mean(January,
February, March)) %>% dplyr::ungroup() %>%
dplyr::select(Latitude, Longitude, proj_Summer_SST,
proj_Winter_SST)) %>% dplyr::mutate(change_Summer_SST = proj_Summer_SST -
obs_Summer_SST, change_Winter_SST = proj_Winter_SST -
obs_Winter_SST) %>% dplyr::mutate(cross_threshold = dplyr::case_when(obs_Winter_SST >
13.5 & proj_Winter_SST > 13.5 ~ "stays above",
obs_Winter_SST <= 13.5 & proj_Winter_SST <=
13.5 ~ "stays below", TRUE ~ "crosses")) %>%
dplyr::bind_cols(data.frame(richness.1980_sim = richness.1980_sim,
richness.2040_sim = richness.2040_sim,
fdiv.1980_sim = fdiv.1980_sim, fdiv.2040_sim = fdiv.2040_sim,
modularities.1980_sim = modularities.1980_sim,
modularities.2040_sim = modularities.2040_sim))
```

```
## Joining, by = c("Latitude", "Longitude")
```

Produce a map of these categories to include in **Supplementary Material**. The bottom panel of this figure shows boxplots for each variable

```
source("Functions/plot_threshold_map.r")
threshold_map <- plot_threshold_map(SST_changes)

threshold_map
```



## Investigating patterns for two important invasive species

Investigate how co-occurrences for two widespread and increasingly captured invasive species are linked with winter SST variation. First, identify which species show strong and climatically variable associations with other species

```
wint_temp_coefs <- lapply(seq_along(CRF$key_coefs),
  function(x) {
    coefs <- CRF$key_coefs[[x]] %>% dplyr::filter(grepl("mean.wint.temp_",
      Variable)) %>% dplyr::filter(Rel_importance >
      0.015)
    list(n_coefs = nrow(coefs), coefs = coefs)
  })
names(wint_temp_coefs) <- names(CRF$key_coefs)
which(purrr::map_dbl(wint_temp_coefs, "n_coefs") >
  0)
```

```
##      Psenes_pellucidus Symphodus_melanocercus          Alepes_djedaba
##                      1                         18                     21
##      Pomadasys_incisus                         158
```

A key species is `Alepes_djedaba` (the shrimp scad; species 21). Which species is it tightly correlated with?

```
CRF$key_coefs[[21]][, 1:3]
```

```
##                               Variable Rel_importance Standardised_coef
## 11                      mean.wint.temp      0.41352199     2.4781665
## 1           Scomberomorus_commerson      0.20108351     1.7281022
## 2             Upeneus_moluccensis      0.07593946     1.0619766
## 6       Halobatrachus_didactylus      0.05562045    -0.9088630
## 5            Gonichthys_cocco      0.05527705    -0.9060531
## 8            Lappanella_fasciata      0.04999014    -0.8616350
## 12 mean.wint.temp_Scomberomorus_commerson      0.03902868     0.7613303
## 7            Hymenocephalus_italicus      0.02874630     0.6533895
## 9            Lepidotrigla_cavillone      0.01946096    -0.5376050
## 4            Coryphaena_ippurus      0.01371576     0.4513269
## 10           Nettastoma_melanurum      0.01325844    -0.4437387
## 3              Belone_belone      0.01011000     0.3874866
```

The shrimp scad's occurrence probability is strongly related to occurrence of `Scomberomorus_commerson` (the Spanish mackerel), and this relationship varies across winter SST gradients. These relationships produce **Figure 4**.

Use historical 1980 data to create subsets for when species co-occur and do not co-occur

```
# For the mackerel
pres_plot_data <- Medfish.mrf.dat %>% dplyr::bind_cols(coords) %>%
  dplyr::filter(Alepes_djedaba == 1) %>%
  dplyr::select(Scomberomorus_commerson,
    mean.wint.temp, mean.summ.temp, Latitude,
    Longitude)
pres_plot_data$presence <- "scad present"

abs_plot_data <- Medfish.mrf.dat %>% dplyr::bind_cols(coords) %>%
  dplyr::filter(Alepes_djedaba == 0) %>%
  dplyr::select(Scomberomorus_commerson,
    mean.wint.temp, mean.summ.temp, Latitude,
    Longitude)
abs_plot_data$presence <- "scad absent"
full_plot_data <- rbind(pres_plot_data, abs_plot_data)

# For the scad
mack_pres_plot_data <- Medfish.mrf.dat %>%
  dplyr::bind_cols(coords) %>% dplyr::filter(Scomberomorus_commerson ==
  1) %>% dplyr::select(Alepes_djedaba,
    mean.wint.temp, mean.summ.temp, Latitude,
    Longitude)
mack_pres_plot_data$presence <- "mackerel present"

mack_abs_plot_data <- Medfish.mrf.dat %>%
  dplyr::bind_cols(coords) %>% dplyr::filter(Scomberomorus_commerson ==
  0) %>% dplyr::select(Alepes_djedaba,
    mean.wint.temp, mean.summ.temp, Latitude,
    Longitude)
mack_abs_plot_data$presence <- "mackerel absent"
mack_full_plot_data <- rbind(mack_pres_plot_data,
  mack_abs_plot_data)
```

The proportion of observed SST range that the species occur together in during winter shows that co-occurrence increases in warmer winter waters

```
round((pres_plot_data %>% dplyr::filter(Scomberomorus_commerson ==
  1) %>% dplyr::select(mean.wint.temp) %>%
  max() - pres_plot_data %>% dplyr::filter(Scomberomorus_commerson ==
  1) %>% dplyr::select(mean.wint.temp) %>%
  min())/(max(full_plot_data$mean.wint.temp) -
  min(full_plot_data$mean.wint.temp)),
  2)
```

```
## [1] 0.33
```

Co-occurrence takes place throughout a broader proportion of observed SST range during Summer, which is why the model does not support an effect of summer SST on co-occurrence probability

```
round((pres_plot_data %>% dplyr::filter(Scomberomorus_commerson ==
  1) %>% dplyr::select(mean.summ.temp) %>%
  max() - pres_plot_data %>% dplyr::filter(Scomberomorus_commerson ==
  1) %>% dplyr::select(mean.summ.temp) %>%
  min())/(max(full_plot_data$mean.summ.temp) -
  min(full_plot_data$mean.summ.temp)),
  2)
```

Generate species co-occurrence plots from the observed historical data

```
summer_col <- "red"
winter_col <- "blue"
axis_degree = function(x) parse(text = paste0(x,
  "^\circ", "*C"))
full_plot_data <- rbind(pres_plot_data, abs_plot_data)

binomial_smooth <- function(...) {
  suppressMessages(geom_smooth(method = "glm",
    method.args = list(family = "binomial",
      control = glm.control(maxit = 5)),
    ...))
}

mack_plot <- ggplot(full_plot_data, aes(x = mean.wint.temp,
  y = Scomberomorus_commerson)) + theme_bw() +
  binomial_smooth(data = abs_plot_data,
    fill = "black", colour = "black",
    size = 0.5, level = 0.999) + binomial_smooth(data = pres_plot_data,
    size = 0.5, fill = winter_col, colour = "black",
    level = 0.999) + geom_rug(data = abs_plot_data %>%
  dplyr::filter(Scomberomorus_commerson ==
  1), colour = "black", size = 0.03,
  alpha = 0.1, sides = "t") + geom_rug(data = pres_plot_data %>%
  dplyr::filter(Scomberomorus_commerson ==
  0), colour = "black", size = 0.03,
  alpha = 0.1, sides = "b") + geom_rug(data = pres_plot_data %>%
  dplyr::filter(Scomberomorus_commerson ==
  1), size = 0.03, colour = winter_col,
  alpha = 0.15, sides = "t") + geom_rug(data = pres_plot_data %>%
  dplyr::filter(Scomberomorus_commerson ==
  0), size = 0.03, colour = winter_col,
  alpha = 0.15, sides = "b") + ylab("Mackerel occurrence probability") +
  xlab("winter SST") + scale_x_continuous(label = axis_degree) +
  scale_y_continuous(breaks = c(0, 0.25,
  0.5, 0.75, 1)) + theme(panel.grid.major = element_blank(),
  legend.position = "none", panel.grid.minor = element_blank(),
  axis.text = element_text(size = 5.5),
  axis.title = element_text(size = 7)) +
  annotate("text", x = 10.4, y = 0.18,
  label = "Shrimp scad\nabsent", size = 2.75,
  colour = "black") + annotate("text",
  x = 14.2, y = 0.9, label = "Shrimp scad\npresent",
```

```

size = 2.75, colour = winter_col)

mack_summ_plot <- ggplot(full_plot_data,
  aes(x = mean.summ.temp, y = Scomberomorus_commerson)) +
  theme_bw() + binomial_smooth(data = abs_plot_data,
  fill = "black", colour = "black", size = 0.5,
  level = 0.999) + binomial_smooth(data = pres_plot_data,
  size = 0.5, fill = summer_col, colour = "black",
  level = 0.999) + geom_rug(data = abs_plot_data %>%
  dplyr::filter(Scomberomorus_commerson ==
    1), colour = "black", size = 0.03,
  alpha = 0.1, sides = "t") + geom_rug(data = abs_plot_data %>%
  dplyr::filter(Scomberomorus_commerson ==
    0), colour = "black", size = 0.03,
  alpha = 0.1, sides = "b") + geom_rug(data = pres_plot_data %>%
  dplyr::filter(Scomberomorus_commerson ==
    1), size = 0.03, colour = summer_col,
  alpha = 0.15, sides = "t") + geom_rug(data = pres_plot_data %>%
  dplyr::filter(Scomberomorus_commerson ==
    0), size = 0.03, colour = summer_col,
  alpha = 0.15, sides = "b") + ylab("Mackerel occurrence probability") +
  xlab("summer SST") + scale_x_continuous(label = axis_degree) +
  scale_y_continuous(breaks = c(0, 0.25,
  0.5, 0.75, 1)) + theme(panel.grid.major = element_blank(),
  legend.position = "none", panel.grid.minor = element_blank(),
  axis.text = element_text(size = 5.5),
  axis.title = element_text(size = 7)) +
  annotate("text", x = 23.15, y = 0.35,
  label = "Shrimp scad\nabsent", size = 2.75,
  colour = "black") + annotate("text",
  x = 20.9, y = 0.9, label = "Shrimp scad\npresent",
  size = 2.75, colour = summer_col)

scad_plot <- ggplot(mack_full_plot_data,
  aes(x = mean.wint.temp, y = Alepes_djedaba)) +
  theme_bw() + binomial_smooth(data = mack_abs_plot_data,
  fill = "black", colour = "black", size = 0.5,
  level = 0.999) + binomial_smooth(data = mack_pres_plot_data,
  size = 0.5, fill = winter_col, colour = "black",
  level = 0.999) + geom_rug(data = mack_abs_plot_data %>%
  dplyr::filter(Alepes_djedaba == 1), colour = "black",
  size = 0.03, alpha = 0.1, sides = "t") +
  geom_rug(data = mack_abs_plot_data %>%
  dplyr::filter(Alepes_djedaba == 0),
  colour = "black", size = 0.03, alpha = 0.1,
  sides = "b") + geom_rug(data = mack_pres_plot_data %>%
  dplyr::filter(Alepes_djedaba == 1), size = 0.03,
  colour = winter_col, alpha = 0.15, sides = "t") +
  geom_rug(data = mack_pres_plot_data %>%
  dplyr::filter(Alepes_djedaba == 0),
  size = 0.03, colour = winter_col,
  alpha = 0.15, sides = "b") + ylab("Scad occurrence probability") +
  xlab("winter SST") + scale_x_continuous(label = axis_degree) +

```

```

scale_y_continuous(breaks = c(0, 0.25,
  0.5, 0.75, 1)) + theme(panel.grid.major = element_blank(),
legend.position = "none", panel.grid.minor = element_blank(),
axis.text = element_text(size = 5.5),
axis.title = element_text(size = 7)) +
annotate("text", x = 10.5, y = 0.11,
  label = "Spanish mackerel\nabsent",
  size = 2.75, colour = "black") +
annotate("text", x = 14.2, y = 0.9, label = "Spanish mackerel\npresent",
  size = 2.75, colour = winter_col)

scad_summ_plot <- ggplot(mack_full_plot_data,
aes(x = mean.summ.temp, y = Alepes_djedaba)) +
theme_bw() + binomial_smooth(data = mack_abs_plot_data,
fill = "black", colour = "black", size = 0.5,
level = 0.999) + binomial_smooth(data = mack_pres_plot_data,
size = 0.5, fill = summer_col, colour = "black",
level = 0.999) + geom_rug(data = mack_abs_plot_data %>%
dplyr::filter(Alepes_djedaba == 1), colour = "black",
size = 0.03, alpha = 0.1, sides = "t") +
geom_rug(data = mack_abs_plot_data %>%
dplyr::filter(Alepes_djedaba == 0),
colour = "black", size = 0.03, alpha = 0.1,
sides = "b") + geom_rug(data = mack_pres_plot_data %>%
dplyr::filter(Alepes_djedaba == 1), size = 0.03,
colour = summer_col, alpha = 0.15, sides = "t") +
geom_rug(data = mack_pres_plot_data %>%
dplyr::filter(Alepes_djedaba == 0),
size = 0.03, colour = summer_col,
alpha = 0.15, sides = "b") + ylab("Scad occurrence probability") +
xlab("summer SST") + scale_x_continuous(label = axis_degree) +
scale_y_continuous(breaks = c(0, 0.25,
  0.5, 0.75, 1)) + theme(panel.grid.major = element_blank(),
legend.position = "none", panel.grid.minor = element_blank(),
axis.text = element_text(size = 5.5),
axis.title = element_text(size = 7)) +
annotate("text", x = 20.1, y = 0.33,
  label = "Spanish mackerel\nabsent",
  size = 2.75, colour = "black") +
annotate("text", x = 24, y = 0.8, label = "Spanish mackerel\npresent",
  size = 2.75, colour = summer_col)

```

Add images of the two species and generate **Figure 4**

```

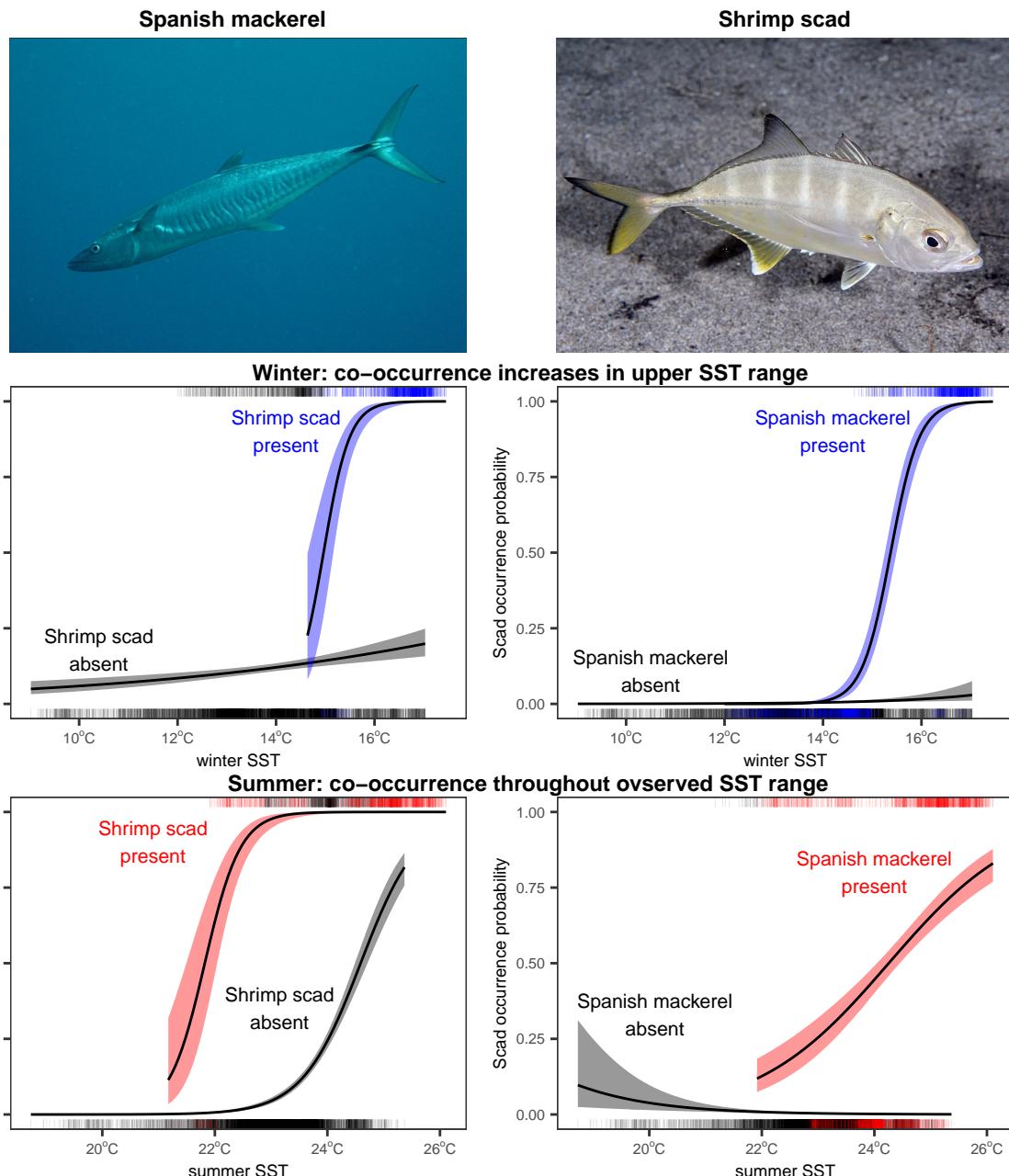
img1 <- jpeg::readJPEG("Images/mackerel.jpg")
img2 <- jpeg::readJPEG("Images/shrimp_scad.jpg")
im_A <- ggplot() + ggpubr::background_image(img1) +
  theme(plot.margin = margin(t = 0.5, l = 1.1,
    r = 0.2, b = 0.3, unit = "cm"))
im_B <- ggplot() + ggpubr::background_image(img2) +
  theme(plot.margin = margin(t = 0.5, l = 1.1,
    r = 0.2, b = 0.3, unit = "cm"))
ggpubr::ggarrange(im_A, im_B, mack_plot,
  scad_plot, mack_summ_plot, scad_summ_plot,

```

```

labels = c("Spanish mackerel", "Shrimp scad",
      "", "Winter: co-occurrence increases in upper SST range",
      "", "Summer: co-occurrence throughout observed SST range"),
font.label = list(size = 9), ncol = 2,
nrow = 3, heights = c(0.9, 1, 1), hjust = c(-1.1,
-1.8, -1, 0.41, -1, 0.42), vjust = c(1.5,
1.5, 1.5, 0.5, 1.5, 0.5))

```



Create the remaining figures for **Supplementary Material**. Read in the GSA shapefile and manipulate using `sf` functions

```
library(viridis)
library(ggplot2)
library(dplyr)
library(sf)
gsa.shp <- sf::st_read("./Med_shapefiles/GSAs_simplified.shp")
```

Combine the two GSA 11s into one polygon

```
gsa_geoms <- gsa.shp %>% dplyr::mutate(District = dplyr::case_when(F_GSA_LIB %in%
  c("GSA 11.1", "GSA 11.2") ~ "GSA 11",
  TRUE ~ as.character(F_GSA_LIB))) %>%
  group_by(District) %>% summarize()
st_geometry(gsa.shp) <- NULL
gsa_correct.shp <- gsa.shp %>% dplyr::mutate(District = dplyr::case_when(F_GSA_LIB %in%
  c("GSA 11.1", "GSA 11.2") ~ "GSA 11",
  TRUE ~ as.character(F_GSA_LIB))) %>%
  dplyr::filter(F_GSA_LIB != "GSA 11.1") %>%
  dplyr::filter(!F_GSA_LIB %in% c("GSA 28",
  "GSA 29", "GSA 30")) %>% dplyr::left_join(gsa_geoms)
gsa_correct.shp <- st_as_sf(gsa_correct.shp)
```

Set labels for different seas based on GSAs

```
sea_labels <- data.frame(District = unique(gsa_correct.shp$District),
  Sea = c("Alboran", "Alboran", "Alboran",
  "", "", "", "", "", "Tyrrhenian",
  "Tyrrhenian", "", "Central Gulfs",
  "Central Gulfs", "Central Gulfs",
  "", "", "Adriatic", "Adriatic", "",
  "", "", "Aegean", "", "", "", "", ""),
  ""))
gsa_correct.shp = gsa_correct.shp %>% dplyr::left_join(sea_labels)
```

```
## Warning: Column `District` joining character vector and factor, coercing
## into character vector
```

Read in the fishing pressure data for each GSA

```
fish_dat <- readxl::read_xlsx("Fishing_pressures/Med_fishing_pressure.xlsx")
fish_dat.shp <- fish_dat %>% dplyr::select(-Country) %>%
  dplyr::mutate(F_GSA_LIB = paste("GSA",
  "GSA")) %>% dplyr::mutate(District = dplyr::case_when(F_GSA_LIB %in%
  c("GSA 11.1", "GSA 11.2") ~ "GSA 11",
  TRUE ~ as.character(F_GSA_LIB))) %>%
  dplyr::select(-F_GSA_LIB, -GSA) %>% dplyr::group_by(District) %>%
  dplyr::summarise_all(funs(mean)) %>%
  dplyr::ungroup() %>% dplyr::right_join(gsa_correct.shp) %>%
  dplyr::mutate(Fishing_pressure = Total_landing/Shelf_surface) %>%
  dplyr::mutate(Fishing_pressure = as.vector(scale(Fishing_pressure))) %>%
  dplyr::mutate(Demersal_pressure = Demersal_landing/Shelf_surface) %>%
  dplyr::mutate(Demersal_pressure = as.vector(scale(Demersal_pressure))) %>%
  dplyr::mutate(Pelagic_pressure = Pelagic_landing/Shelf_surface) %>%
  dplyr::mutate(Pelagic_pressure = as.vector(scale(Pelagic_pressure)))
```

```

## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.

```

Set plotting parameters

```

mapping_theme = function() {
  theme(legend.title = element_text(size = 6.5,
    hjust = 0.5), legend.text = element_text(size = 6),
    legend.key.size = unit(0.3, "cm"),
    legend.justification = c(1, 1), legend.position = c(0.99,
      0.99), legend.spacing.x = unit(0.1,
      "cm"), legend.box = "horizontal",
    legend.margin = margin(-0.02, 0.04,
      -0.02, 0.04, unit = "cm"), legend.key = element_rect(fill = "gray85"),
    legend.background = element_rect(fill = "gray85",
      colour = "gray85", size = 0.05),
    legend.box.background = element_rect(colour = "gray85",
      fill = "gray85", size = 0.05),
    plot.margin = margin(1, 0.5, -12.5,
      -8), axis.text.x = element_text(size = 6),
    axis.text.y = element_text(size = 6,
      angle = 90, vjust = 0, hjust = 0.7),
    panel.grid.major = element_line(colour = "gray85"),
    panel.background = element_rect(fill = "gray85",
      colour = "black"))
}

scale_x_longitude <- function(xmin = -180,
  xmax = 180, step = 1, ...) {
  xbreaks <- seq(xmin, xmax, step)
  xlabel <- unlist(lapply(xbreaks, function(x) ifelse(x <
    0, parse(text = paste0(-x, "^\circ",
      "*W")), ifelse(x > 0, parse(text = paste0(x,
        "^\circ", "*E")), x))))
  return(scale_x_continuous("", breaks = xbreaks,
    labels = xlabel, expand = c(0, 0),
    ...))
}
scale_y_latitude <- function(ymin = -90,
  ymax = 90, step = 0.5, ...) {
  ybreaks <- seq(ymin, ymax, step)
  ylabel <- unlist(lapply(ybreaks, function(x) ifelse(x <
    0, parse(text = paste0(-x, "^\circ",
      "*S")), ifelse(x > 0, parse(text = paste0(x,
        "^\circ", "*N")), x))))

```

```

    return(scale_y_continuous("", breaks = ybreaks,
      labels = ylabels, expand = c(0, 0),
      ...))
}

```

Get Mediterranean city locations and population sizes from Wikipedia

```

library(rvest)
library(tidyverse)
library(stringr)

html_population <- read_html("https://en.wikipedia.org/wiki/
                               List_of_coastal_settlements_of_the_Mediterranean_Sea")
med_cities <- html_population %>% html_nodes("table") %>%
  .[[1]] %>% html_table(fill = TRUE) %>%
  dplyr::select(City, Country, Population) %>%
  dplyr::filter(!Population == "N/A")

```

Geocode each city's coordinates using OpenStreetMap

```

nominatim_osm = function(address = NULL) {
  if (suppressWarnings(is.null(address))) {
    return(data.frame(ADDRESS = address,
                      LONGITUDE = NA, LATITUDE = NA))
  }
  tryCatch(d <- jsonlite::fromJSON(gsub("\\@addr\\@",
                                         gsub("\\s+", "\%20", address), "http://nominatim.openstreetmap.org/
                                         search/@addr?format=json&addressdetails=0&limit=1")),
           error = function(c) return(data.frame()))
  if (!exists("d")) {
    return(data.frame(ADDRESS = address,
                      LONGITUDE = NA, LATITUDE = NA))
  } else if (length(d) == 0) {
    return(data.frame(ADDRESS = address,
                      LONGITUDE = NA, LATITUDE = NA))
  } else {
    return(data.frame(ADDRESS = address,
                      LONGITUDE = as.numeric(d$lon),
                      LATITUDE = as.numeric(d$lat)))
  }
}

med_city_coords <- lapply(seq_len(nrow(med_cities)),
  function(x) {
    cat("Geocoding city", x, "of", nrow(med_cities),
        "...\\n")
    Sys.sleep(2)
    coords <- nominatim_osm(paste(tolower(med_cities$City[x]),
                                    tolower(med_cities$Country[x]),
                                    sep = ", "))
  })

med_city_coords <- do.call(rbind, med_city_coords) %>%
  dplyr::bind_cols(med_cities) %>% dplyr::filter(!is.na(LATITUDE)) %>%
  dplyr::select(-ADDRESS) %>% janitor::clean_names()

```

```

med_city_coords$population <- as.numeric(gsub(",",
      "", med_city_coords$population))
save(med_city_coords, file = "Processed data/med_city_coords.rda")

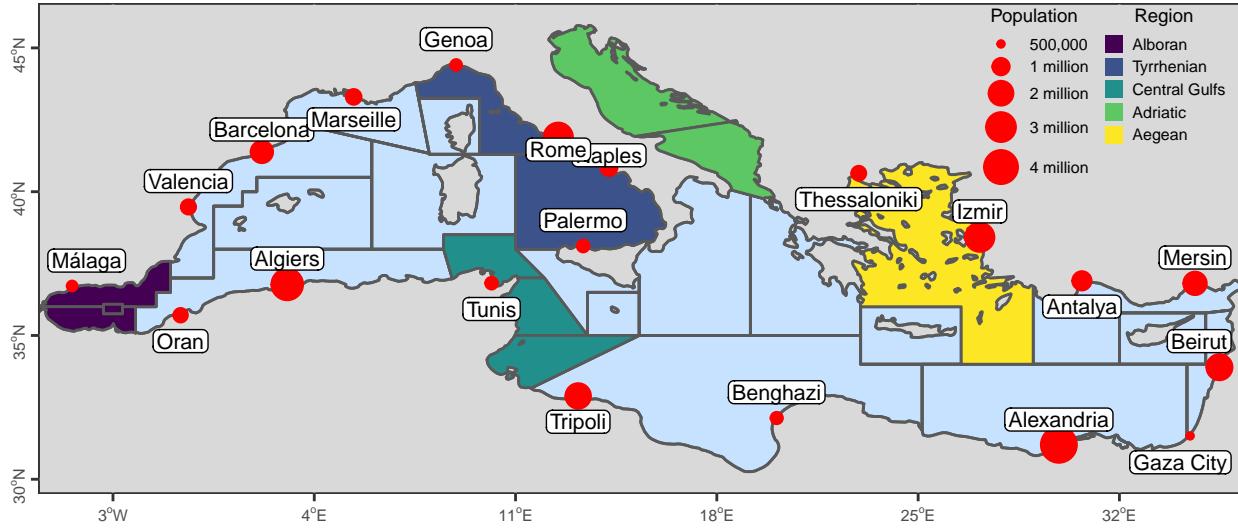
```

Primary Mediterranean plot with region labels

```

load("Processed data/med_city_coords.rda")
library(ggrepel)
supp1_map <- ggplot() + coord_fixed() + theme_bw() +
  geom_sf(data = fish_dat.shp, fill = "slategray1",
    aes(geometry = geometry)) + geom_sf(data = fish_dat.shp,
    aes(fill = Sea, geometry = geometry)) +
  geom_point(data = med_city_coords %>%
    dplyr::filter(population > 5e+05) %>%
    dplyr::filter(!city %in% c("Tangier",
      "Port Said")), aes(x = longitude,
    y = latitude, size = population),
    colour = "red") + geom_label_repel(data = med_city_coords %>%
    dplyr::filter(population > 5e+05) %>%
    dplyr::filter(!city %in% c("Tangier",
      "Port Said")), aes(x = longitude,
    y = latitude, label = city), type = "closed",
    direction = "y", size = 2.85, point.padding = 0.1,
    label.padding = 0.1, force = 5, max.iter = 5000) +
  mapping_theme() + scale_fill_manual(name = "Region",
    values = viridis(5, option = "viridis"),
    limits = c("Alboran", "Tyrrhenian", "Central Gulfs",
      "Adriatic", "Aegean")) + scale_size(name = "Population",
    range = c(1, 6), breaks = c(515556, 1e+06,
      2e+06, 3e+06, 4e+06), labels = c("500,000",
      "1 million", "2 million", "3 million",
      "4 million")) + scale_x_longitude(xmin = -3,
    xmax = 35, step = 7) + scale_y_continuous(labels = function(label) {
      parse(text = paste0(label, "°", "*N")))
    }) + labs(x = "", y = "") + guides(fill = guide_legend(override.aes = list(size = 1.5,
      alpha = 1, color = NA)))
## Warning: Ignoring unknown parameters: type
supp1_map

```



Determine which GSA each point in the data belongs to

```

plot_points = Obs.1980.Climate.dat %>% dplyr::ungroup() %>%
  dplyr::select(Latitude, Longitude) %>%
  dplyr::mutate(LATITUDE = Latitude, LONGITUDE = Longitude)
plot_points <- sf::st_as_sf(plot_points,
  coords = c("LONGITUDE", "LATITUDE"))
sf::st_crs(plot_points) <- sf::st_crs(gsa_correct.shp)

# Intersect to find those that directly
# fall within a GSA polygon
point_gsas <- sf::st_intersection(plot_points,
  gsa_correct.shp)
points_located <- point_gsas %>% dplyr::select(Latitude,
  Longitude, District) %>% right_join(Obs.1980.Climate.dat %>%
  dplyr::ungroup() %>% dplyr::select(Latitude,
  Longitude))

# Find nearest polygon for remaining
# points
points_notlocated <- points_located %>% dplyr::filter(is.na(District))

# Convert to a list of st_point objects
pnts_geoms <- purrr::map2(points_notlocated$Longitude,
  points_notlocated$Latitude, ~sf::st_sfc(sf::st_point(c(.x,
  .y)), crs = sf::st_crs(gsa_correct.shp)))

# Define a distance function to return
# closest polygon
dist_fun <- function(x, y) {
  which.min(st_distance(x, y))
}

# Get closest polygons
closest_gsas <- gsa_correct.shp$District[purrr::map_dbl(pnts_geoms,
  dist_fun, y = st_centroid(gsa_correct.shp))]
points_notlocated$District <- closest_gsas

```

```

point_gsas <- rbind(points_located %>% dplyr::filter(!is.na(District)),
  points_notlocated)

save(point_gsas, file = "Processed data/point_gsas.rda")

```

Plot of fishing pressures

```

load("Processed data/point_gsas.rda")

tot_pressure <- ggplot() + theme_bw() + geom_sf(data = fish_dat.shp,
  aes(fill = Fishing_pressure, geometry = geometry)) +
  scale_fill_viridis(name = "Total\npressure",
    option = "magma", breaks = c(0, 3),
    labels = c("low", "high")) + mapping_theme() +
  scale_x_continuous(labels = rep("", 6),
    breaks = seq(-3, 35, 7)) + coord_sf(xlim = c(-6,
  36.25), ylim = c(29.99, 45.75), expand = F) +
  scale_y_latitude(ymin = 30, ymax = 45,
    step = 5) + labs(x = "", y = "")

dem_pressure <- ggplot() + theme_bw() + geom_sf(data = fish_dat.shp,
  aes(fill = Demersal_pressure, geometry = geometry)) +
  scale_fill_viridis(name = "Demersal\npressure",
    option = "magma", breaks = c(0, 3),
    labels = c("low", "high")) + mapping_theme() +
  coord_sf(xlim = c(-6, 36.25), ylim = c(29.99,
  45.75), expand = F) + scale_x_continuous(labels = rep("", 6),
  breaks = seq(-3, 35, 7)) + scale_y_latitude(ymin = 30,
  ymax = 45, step = 5) + labs(x = "", y = "")

pel_pressure <- ggplot() + theme_bw() + geom_sf(data = fish_dat.shp,
  aes(fill = Pelagic_pressure, geometry = geometry)) +
  scale_fill_viridis(name = "Pelagic\npressure",
    option = "magma", breaks = c(0, 3),
    labels = c("low", "high")) + mapping_theme() +
  coord_sf(xlim = c(-6, 36.25), ylim = c(29.99,
  45.75), expand = F) + scale_x_longitude(xmin = -3,
  xmax = 35, step = 7) + scale_y_latitude(ymin = 30,
  ymax = 45, step = 5) + labs(x = "", y = "")

```

Plot of predicted community metric changes

```

gsa_change_dat <- point_gsas %>% dplyr::bind_cols(data.frame(richness.1980_sim = richness.1980_sim,
  richness.2040_sim = richness.2040_sim,
  fdiv.1980_sim = fdiv.1980_sim, fdiv.2040_sim = fdiv.2040_sim,
  modularities.1980_sim = modularities.1980_sim,
  modularities.2040_sim = modularities.2040_sim)) %>%
  dplyr::group_by(District) %>% dplyr::summarise(Mean_rich_change = mean(richness.2040_sim -
  richness.1980_sim), Mean_div_change = mean(fdiv.2040_sim -
  fdiv.1980_sim), Mean_mod_change = mean(modularities.2040_sim -
  modularities.1980_sim), N_samples = n())
st_geometry(gsa_change_dat) <- NULL

gsa_change.shp <- gsa_change_dat %>% dplyr::left_join(fish_dat.shp)

```

```

rich_change <- ggplot() + geom_sf(data = gsa_change.shp,
aes(fill = Mean_rich_change, geometry = geometry)) +
mapping_theme() + coord_sf(xlim = c(-6,
36.25), ylim = c(29.99, 45.75), expand = F) +
scale_fill_gradient2(name = expression(Delta ~
richness), low = "blue", mid = "darkgrey",
high = "red", midpoint = 0, breaks = c(-12,
0, 4), labels = c("-12", "0",
"+4")) + scale_x_continuous(labels = rep("", 6),
breaks = seq(-3, 35, 7)) + scale_y_continuous(labels = c(rep("", 4)),
breaks = seq(30, 45, 5)) + labs(x = "",
y = "")

div_change <- ggplot() + geom_sf(data = gsa_change.shp,
aes(fill = Mean_div_change, geometry = geometry)) +
mapping_theme() + coord_sf(xlim = c(-6,
36.25), ylim = c(29.99, 45.75), expand = F) +
scale_fill_gradient2(name = expression(Delta ~
diversity), low = "blue", mid = "darkgrey",
high = "red", midpoint = quantile(gsa_change.shp$Mean_div_change,
probs = c(0.5)), breaks = quantile(gsa_change.shp$Mean_div_change,
probs = c(0.1, 0.9)), labels = c("-", "+")) + scale_x_continuous(labels = rep("", 6),
breaks = seq(-3, 35, 7)) + scale_y_continuous(labels = rep("", 4),
breaks = seq(30, 45, 5)) + labs(x = "",
y = "")

mod_change <- ggplot() + geom_sf(data = gsa_change.shp,
aes(fill = Mean_mod_change, geometry = geometry)) +
mapping_theme() + coord_sf(xlim = c(-6,
36.25), ylim = c(29.99, 45.75), expand = F) +
scale_fill_gradient2(name = expression(Delta ~
modularity), low = "blue", mid = "darkgrey",
high = "red", midpoint = quantile(gsa_change.shp$Mean_mod_change,
probs = c(0.5)), breaks = quantile(gsa_change.shp$Mean_mod_change,
probs = c(0.1, 0.9)), labels = c("-", "+")) + scale_x_longitude(xmin = -3,
xmax = 35, step = 7) + scale_y_continuous(labels = rep("", 4),
breaks = seq(30, 45, 5)) + labs(x = "",
y = "")

```

Put the two sets of plots together for inclusion in **Supplementary Material**

```
supp3_map <- cowplot:::plot_grid(cowplot:::plot_grid(tot_pressure,
  dem_pressure, pel_pressure, ncol = 1,
  rel_widths = c(1, 1, 1), labels = c("a",
  "b", "c"), label_size = 9, align = "h",
  hjust = c(-4, -3.7, -4), vjust = 2.5),
cowplot:::plot_grid(rich_change, div_change,
  mod_change, ncol = 1, rel_widths = c(1,
  1, 1), labels = c("d", "e", "f"),
  label_size = 9, align = "h", hjust = c(-3.6,
  -4, -6.5), vjust = 2.5), ncol = 2,
  rel_widths = c(1, 1))
```

```
supp3_map
```

