

# Supporting information for ‘The Point Process Framework for Integrated Modelling of Biodiversity Data’

Kwaku Peprah Adjei, Philip Mostert, Jorge Sicacha Parada, Emma Skarstein, Robert B. O’Hara

Last updated: August 12 2024 at 13:40:22.

## Setup

These are the packages we will be using:

```
#devtools::install_github("PhilipMostert/PointedSDMs")
library(PointedSDMs) # model fitting
library(ggplot2)     # plotting
library(raster)      # plotting
library(mapproj)     # map options for plotting
library(INLA)        # functions for specifying mesh
library(dplyr)       # data handling
library(sf)          # spatial stuff
library(showtext)    # font for plot
library(giscoR)      # Polygon object
library(patchwork)   # combining figures
```

## Downloading data

Before you run this file, make sure you have the following files in the given locations:

- `data/environmental_covariates.rds`
  - should be in the data folder already.
- `data/Norwegian_lakes.rds`
  - can be downloaded from <https://bird.unit.no/resources/9b27e8f0-55dd-442c-be73-26781dad94c8/> content, click on “Innhold”-tab at the bottom of the page and select “Norwegian\_lakes.rds”.
- `data/artsobs_clean.rds`
  - created from running the script `R/data_preparation.R`, see further instructions there.
- `data/survey_clean.rds`
  - also created from running the script `R/data_preparation.R`, see further instructions there.

## Loading Norway map and making mesh

We begin by making a spatial mesh out of a map of Norway.

```

#proj <- '+proj=utm +zone=32 +datum=WGS84 +units=km +no_defs'#m
proj <- '+proj=tmerc +lat_0=58 +lon_0=6.05625 +k=1 +x_0=0 +y_0=0 +a=6377492.018 +units=km +no_defs +typ
norway.poly <- giscoR::gisco_get_countries(year = 2020, country = 'Norway', resolution = 60)
norway.poly <- st_transform(norway.poly, proj)
norway.poly <- st_cast(st_as_sf(norway.poly), 'POLYGON')
norway.poly <- norway.poly[which.max(st_area(norway.poly)),]

norway.poly.simp <- rmapshaper::ms_simplify(norway.poly, keep = 0.8)

```

Adjusting the mesh to be coarser is the easiest way to decrease the run-time for the model. With the following mesh, the model estimation takes a long time to complete, but feel free to change the `max.edge` or `cutoff` to get a coarser mesh.

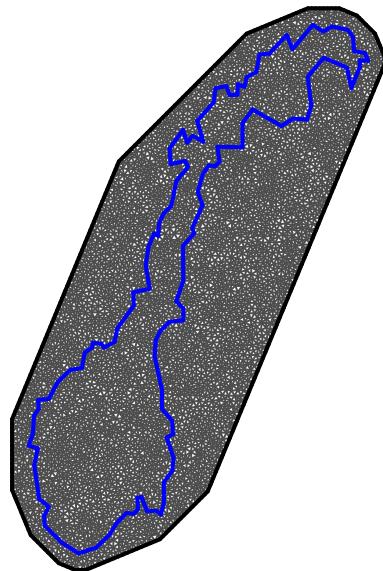
```

mesh <- inla.mesh.2d(boundary = inla.sp2segment(norway.poly.simp),
                      cutoff = 0.3 * 30, # 30smallest allowed distance between points
                      max.edge = c(6, 3) * 7.5, # 10decrease this for more int. points
                      offset = c(1, 1) * 50, #50
                      crs = st_crs(proj))

ipoints = fm_int(mesh, norway.poly.simp)

plot(mesh)

```



## Setting up covariate data

Next we load the environmental data, which will be used as covariates in our model.

```

covariates_raw <- readRDS("data/environmental_covariates.RDS")

covariates <- covariates_raw %>%
  # Log-transform area of lake
  dplyr::mutate(log_area = log(area_km2)) %>%
  # Log-transform catchment area of lake
  dplyr::mutate(log_catchment = log(catchment_area_km2)) %>%
  # Remove some uninformative variables

```

```

dplyr::select(-c(ebint, no_vatn_lnr, eb_waterregionID))

# Choose from
# "decimalLatitude", "decimalLongitude",
# "log_area", "perimeter_m", "distance_to_road",
# "eurolst_bio10", "catchment_area_km2", "SCI", "HFP"

Use <- c("SCI", "HFP")

cov_pixel <- SpatialPixelsDataFrame(
  points = covariates[,c("decimalLongitude", "decimalLatitude")],
  data = data.frame(covariates[,Use]),
  proj4string = CRS('+proj=lonlat +zone=32 +datum=WGS84 +units=m +no_defs'),
  tol = 0.99)
if (length(Use) == 1) names(cov_pixel@data) <- Use
# Scale covariates and convert to terra::rast
cov_raster <- scale(terra::project(terra::rast(cov_pixel), proj))

```

## Observation data

For this model, we have two observation sets, one which is downloaded from GBIF and one that is a survey dataset (see separate document for download instructions).

```

fishes <- c("Esox_lucius", "Perca_fluviatilis", "Salmo_trutta", "Salvelinus_alpinus")

survey <- readRDS("data/survey_clean.rds") %>%
  filter(species %in% fishes) %>%
  st_as_sf(coords = c("decimalLongitude", "decimalLatitude"),
            crs = '+proj=lonlat +zone=32 +datum=WGS84 +units=m +no_defs') %>%
  mutate(decimalLongitude = st_coordinates(.)[,1], decimalLatitude = st_coordinates(.)[,2]) %>%
  st_transform(proj) %>%
  st_intersects(., norway.poly.simp, sparse = FALSE)[,1]
artsobs <- readRDS("data/artsobs_clean.rds") %>%
  filter(species %in% fishes) %>%
  st_as_sf(coords = c("decimalLongitude", "decimalLatitude"),
            crs = '+proj=lonlat +zone=32 +datum=WGS84 +units=m +no_defs') %>%
  mutate(decimalLongitude = st_coordinates(.)[,1], decimalLatitude = st_coordinates(.)[,2]) %>%
  st_transform(proj) %>%
  st_intersects(., norway.poly.simp, sparse = FALSE)[,1]

```

We can plot the observed data points:

```

showtext_auto()
showtext_opts(dpi = 300)
f1 <- "Open sans"
font_add_google(f1, f1)

norway <- ggplot2::map_data("world", region = "Norway(?!:Svalbard)")
norway <- setdiff(norway, dplyr::filter(norway, subregion == "Jan Mayen"))

p_artsobs <- ggplot(artsobs, aes(x = decimalLongitude, y = decimalLatitude)) +
  geom_polygon(data = norway, aes(long, lat, group = group),
               color="grey80", fill = "grey95") +
  geom_point(color = "darkorange2", size = 0.5, alpha = 0.3) +

```

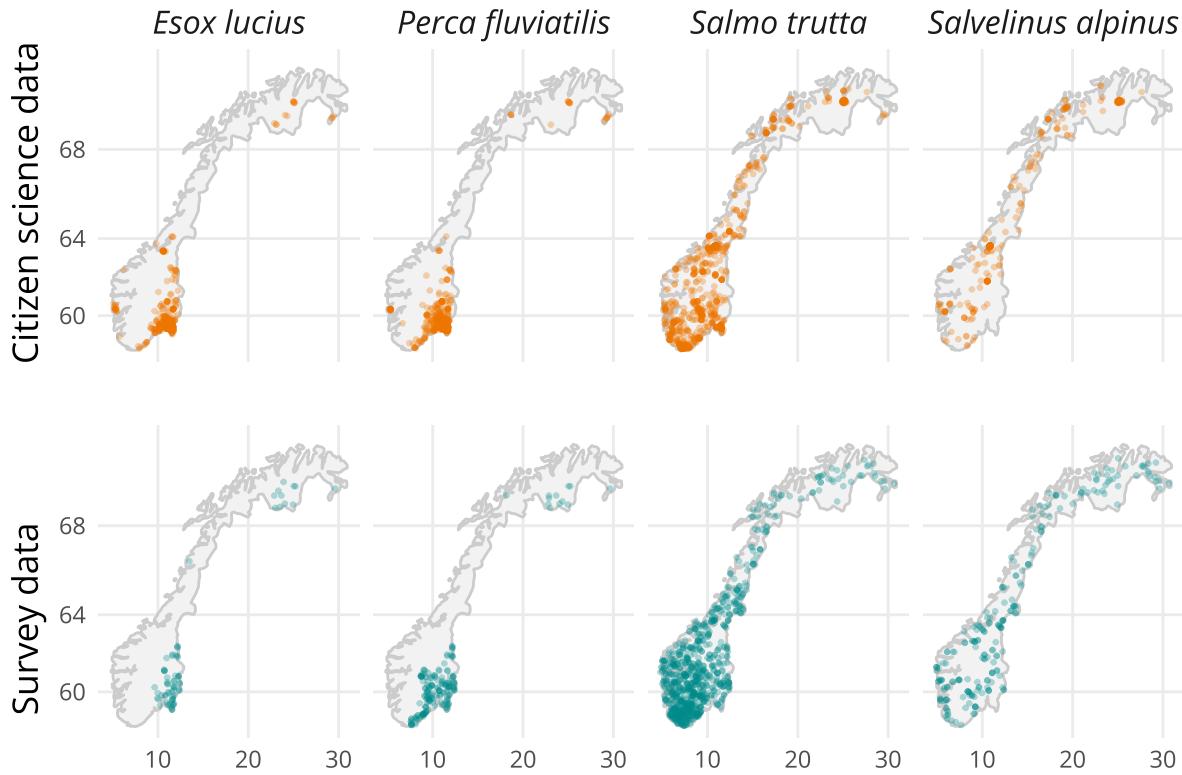
```

facet_wrap(~species, nrow = 1,
          labeller = labeller(species = function(string) sub("_", " ", string))) +
coord_map() +
labs(tag = "Citizen science data") +
theme_minimal() +
theme(text = element_text(family = f1),
      strip.text = element_text(family = f1, size = 12, face = "italic"),
      plot.tag = element_text(angle = 90, hjust = 0.5),
      plot.tag.position = c(-0.03, 0.45),
      legend.position = "none",
      axis.title = element_blank(),
      axis.text.x = element_blank())

p_survey <- ggplot(survey %>% filter(occurrenceStatus == 1),
                    aes(x = decimalLongitude, y = decimalLatitude)) +
geom_polygon(data = norway, aes(long, lat, group = group),
             color="grey80", fill = "grey95") +
geom_jitter(color = "darkcyan", size = 0.5, alpha = 0.3) +
facet_wrap(~species, nrow = 1) +
coord_map() +
xlab("Longitude") +
ylab("Latitude") +
labs(tag = "Survey data") +
theme_minimal() +
theme(text = element_text(family = f1),
      plot.tag = element_text(angle = 90, hjust = 0.5),
      plot.tag.position = c(-0.03, 0.45),
      legend.position = "none",
      axis.title = element_blank(),
      strip.text = element_blank(),
      plot.margin = margin(l = 30))

p_artsobs / p_survey

```



```
ggsave("figures/presence_points.pdf", height = 5, width = 8)
ggsave("figures/presence_points.png", height = 5, width = 8)
```

## Separate models for the two datasets

For the presence/absence survey data, we use a Bernoulli distribution, where the presence probability for species  $j \in \{Salmo trutta, Perca fluviatilis, Esox lucius, Salvelinus alpinus\}$  depends on some covariates  $x(s)$ , along with a spatial field  $\xi_j(s)$ :

$$Y_{PA,j}(s_i) \sim \text{Bernoulli}(p_{PA,j}(s_i))$$

$$\text{cloglog}(p_{PA,j}(s_i)) = \alpha_{PA,j} + x(s_i)^T \beta_j + \xi_j(s_i).$$

We first prepare the model using the `startSpecies` function.

```
surveyModel <- startSpecies(
  survey,
  Boundary = norway.poly,
  IPS = ipoints,
  spatialCovariates = cov_raster,
  speciesName = "species",
  speciesSpatial = 'replicate',
  speciesIntercept = FALSE,
  pointsIntercept = FALSE,
  responsePA = "occurrenceStatus",
  pointsSpatial = NULL,
  Mesh = mesh,
  Projection = proj
)
```

We then specify priors for the spatial effects using `.$specifySpatial` and for the intercepts using `.$priorsFixed`. We also set the precision of the group model for the spatial effect to fixed to stabilize the model using `.$specifyRandom`.

```
for (fish in fishes) {

  surveyModel$specifySpatial(Species = fish,
                               prior.range = c(50, 0.01),
                               prior.sigma = c(0.5, 0.01),
                               constr = FALSE)

  surveyModel$priorsFixed(Effect = 'intercept',
                         Species = fish,
                         mean.linear = 0,
                         prec.linear = 10)
}

surveyModel$specifyRandom(speciesGroup = list(model = "iid",
                                              hyper = list(prec = list(initial = 7.5, fixed = TRUE)))))

surveyModel$priorsFixed(Effect = 'HFP', mean.linear = 0, prec.linear = 0.1)
surveyModel$priorsFixed(Effect = 'SCI', mean.linear = 0, prec.linear = 0.1)
```

We first define some model options, and then fit the model using `fitISDM`.

```
modelOptions <- list(num.threads = 4,
                     control.inla = list(int.strategy = 'ccd',
                                         cmin = 0,
                                         control.vb=list(enable=FALSE),
                                         diagonal = 1e-3,
                                         strategy = 'adaptive'),
                     safe = TRUE,
                     inla.mode = 'experimental')

surveyFit <- fitISDM(surveyModel,
                      options = modelOptions)

summary(surveyFit)

## Summary of 'modSpecies' object:
##
## inlabru version: 2.11.1
## INLA version: 24.06.27
##
## Types of data modelled:
##
## survey           Present absence
##
## Summary of the fixed effects for the species:
##
## Summary for Perca_fluviatilis:
##               mean          sd 0.025quant   0.5quant
## Perca_fluviatilis_intercept -1.26815772 0.08565720 -1.4346815 -1.26873201
## Perca_fluviatilis_HFP      0.32794385 0.05695822  0.2162542  0.32794392
## Perca_fluviatilis_SCI     -0.08935951 0.08417959 -0.2544275 -0.08935947
```

```

##          0.975quant      mode      kld
## Perca_fluviatilis_intercept -1.0982250 -1.26869656 1.644011e-08
## Perca_fluviatilis_HFP      0.4396331  0.32794392 5.504681e-11
## Perca_fluviatilis_SCI     0.0757082 -0.08935947 5.526557e-11
##
## Summary for Salmo_trutta:
##          mean      sd 0.025quant 0.5quant 0.975quant
## Salmo_trutta_intercept 0.88504578 0.06218702 0.7591126 0.88609783 1.00438828
## Salmo_trutta_HFP       -0.04253053 0.04493712 -0.1306451 -0.04253146 0.04558934
## Salmo_trutta_SCI      -0.01743668 0.05017972 -0.1158345 -0.01743657 0.08096044
##          mode      kld
## Salmo_trutta_intercept 0.88596111 9.749787e-08
## Salmo_trutta_HFP       -0.04253147 5.553370e-11
## Salmo_trutta_SCI      -0.01743656 5.522390e-11
##
## Summary for Salvelinus_alpinus:
##          mean      sd 0.025quant 0.5quant
## Salvelinus_alpinus_intercept -0.95531437 0.07865170 -1.10812707 -0.9558931
## Salvelinus_alpinus_HFP      0.04871821 0.06612344 -0.08094329 0.0487182
## Salvelinus_alpinus_SCI     0.11754186 0.06918795 -0.01812888 0.1175419
##          0.975quant      mode      kld
## Salvelinus_alpinus_intercept -0.7990322 -0.9558487 2.251264e-08
## Salvelinus_alpinus_HFP      0.1783797 0.0487182 5.526713e-11
## Salvelinus_alpinus_SCI     0.2532126 0.1175419 5.525050e-11
##
## Summary for Esox_lucius:
##          mean      sd 0.025quant 0.5quant 0.975quant
## Esox_lucius_intercept -2.08748837 0.11566251 -2.3128513 -2.0880349 -1.8589570
## Esox_lucius_HFP         0.43182260 0.06968665 0.2951741 0.4318226 0.5684713
## Esox_lucius_SCI        -0.07635387 0.12156465 -0.3147318 -0.0763533 0.1620208
##          mode      kld
## Esox_lucius_intercept -2.0880219 6.351033e-09
## Esox_lucius_HFP         0.4318226 5.526522e-11
## Esox_lucius_SCI        -0.0763533 5.491568e-11

## Time used:
##      Pre = 1.05, Running = 8.39, Post = 1.31, Total = 10.7
## Random effects:
##      Name    Model
##      speciesShared SPDE2 model
##
## Model hyperparameters:
##          mean      sd 0.025quant 0.5quant 0.975quant      mode
## Range for speciesShared 1896.94 3219.25     132.095   981.19   9468.57 331.135
## Stdev for speciesShared 2.33      3.48      0.144      1.29      10.95  0.377
##
## Deviance Information Criterion (DIC) .....: 2600.71
## Deviance Information Criterion (DIC, saturated) ....: 2597.99
## Effective number of parameters .....: 12.67
##
## Watanabe-Akaike information criterion (WAIC) ....: 2600.68
## Effective number of parameters .....: 12.32
##
## Marginal log-Likelihood: -1397.56

```

```

##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
saveRDS(surveyFit, "results/surveyModel.rds")

```

The presence-only data is fitted with a Poisson point process model, where the intensity depends on the same covariates  $x(s)$  and the same spatial field  $\xi_j(s)$ . In the ISDM, we will add an additional spatial field  $\xi_{\text{bias}}(s)$  that is unique to the citizen science data, but shared across all fish species:

$$Y_{PO,j}(s_i) \sim \text{Poisson}(e^{\eta_{PO,j}(s_i)})$$

$$\eta_{PO,j}(s_i) = \alpha_{PO,j} + x(s)^T \beta_j + \xi_j(s_i) + \xi_{\text{bias}}(s_i).$$

```

artsobsModel <- startSpecies(
  artsobs,
  Boundary = norway.poly,                               # Citizen science data
  IPS = ipoints,                                       # Boundary
  spatialCovariates = cov_raster,                      # integration points
  speciesName = "species",                            # Covariates
  speciesSpatial = 'replicate',                       # The column containing species name
  speciesIntercept = FALSE,                           # Creates field per species
  pointsIntercept = FALSE,                           # Turn random intercept off for species
  pointsSpatial = NULL,                             # Turn dataset intercept off
  Mesh = mesh,                                     # NULL since we use speciesSpatial
  Projection = proj                                # NULL since we use speciesSpatial
)

for (fish in fishes) {

  artsobsModel$specifySpatial(Species = fish,
    prior.range = c(50, 0.01),
    prior.sigma = c(0.5, 0.01),
    constr = FALSE)

  artsobsModel$priorsFixed(Effect = 'intercept',
    Species = fish,
    mean.linear = 0,
    prec.linear = 10)
}

artsobsModel$specifyRandom(speciesGroup = list(model = "iid",
  hyper = list(prec = list(initial = 7.5, fixed = TRUE)))))

artsobsModel$priorsFixed(Effect = 'HFP', mean.linear = 0, prec.linear = 0.1)
artsobsModel$priorsFixed(Effect = 'SCI', mean.linear = 0, prec.linear = 0.1)

artsobsFit <- fitISDM(artsobsModel,
  options = modelOptions)

summary(artsobsFit)

## Summary of 'modSpecies' object:
##
## inlabru version: 2.11.1
## INLA version: 24.06.27

```

```

## 
## Types of data modelled:
## 
## artsobs Present only
## 
## Summary of the fixed effects for the species:
## 
## Summary for Salvelinus_alpinus:
##               mean      sd 0.025quant  0.5quant
## Salvelinus_alpinus_intercept -3.4423048 0.24612801 -3.9238101 -3.4426799
## Salvelinus_alpinus_HFP       -0.3685681 0.08336226 -0.5326316 -0.3683609
## Salvelinus_alpinus_SCI      0.2204964 0.05841759  0.1059548  0.2204928
##               0.975quant     mode      kld
## Salvelinus_alpinus_intercept -2.9586956 -3.4426905 7.874424e-10
## Salvelinus_alpinus_HFP       -0.2056780 -0.3683579 1.567216e-09
## Salvelinus_alpinus_SCI      0.3350586  0.2204927 5.097464e-11
## 
## Summary for Esox_lucius:
##               mean      sd 0.025quant  0.5quant  0.975quant
## Esox_lucius_intercept -3.6668304 0.24864144 -4.1532544 -3.6672042 -3.1783115
## Esox_lucius_HFP        0.5435777 0.04484355  0.4557658  0.5435338  0.63163889
## Esox_lucius_SCI        -0.1624930 0.06582774 -0.2916247 -0.1624765 -0.03345534
##               mode      kld
## Esox_lucius_intercept -3.6672154 8.086312e-10
## Esox_lucius_HFP        0.5435334 2.791628e-10
## Esox_lucius_SCI        -0.1624764 5.534394e-11
## 
## Summary for Salmo_trutta:
##               mean      sd 0.025quant  0.5quant  0.975quant
## Salmo_trutta_intercept -3.06070834 0.24200972 -3.53421506 -3.0610657 -2.5851912
## Salmo_trutta_HFP        0.16000795 0.03696410  0.08750630  0.1600139  0.2324759
## Salmo_trutta_SCI        0.04446896 0.03987571 -0.03372543  0.0444694  0.1226609
##               mode      kld
## Salmo_trutta_intercept -3.0610742 6.829807e-10
## Salmo_trutta_HFP        0.1600139 4.404436e-11
## Salmo_trutta_SCI        0.0444694 4.738889e-11
## 
## Summary for Perca_fluviatilis:
##               mean      sd 0.025quant  0.5quant
## Perca_fluviatilis_intercept -3.65333059 0.24849232 -4.13946470 -3.65370435
## Perca_fluviatilis_HFP        0.54626838 0.04072355  0.46655471  0.54621771
## Perca_fluviatilis_SCI        0.01270829 0.05269066 -0.09062196  0.01271064
##               0.975quant     mode      kld
## Perca_fluviatilis_intercept -3.1651013 -3.65371545 8.022881e-10
## Perca_fluviatilis_HFP        0.6262696  0.54621719 4.280362e-10
## Perca_fluviatilis_SCI        0.1160251  0.01271065 3.980341e-11
## 
## Time used:
##   Pre = 1.18, Running = 53.4, Post = 1.63, Total = 56.2
## Random effects:
##   Name      Model
##   speciesShared SPDE2 model
## 
## Model hyperparameters:

```

```

##               mean    sd 0.025quant 0.5quant 0.975quant   mode
## Range for speciesShared 183.52 6.99      170.15  183.39      197.65 183.14
## Stdev for speciesShared  69.09 1.84      65.54   69.07      72.77  69.04
##
## Deviance Information Criterion (DIC) .....: -38108.84
## Deviance Information Criterion (DIC, saturated) ....: -38126.59
## Effective number of parameters .....: -42857.34
##
## Watanabe-Akaike information criterion (WAIC) ....: 9641.85
## Effective number of parameters .....: 2396.07
##
## Marginal log-Likelihood: -25841.05
## is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
saveRDS(artsobsFit, "results/arts0bsModel.rds")

```

## Joint model for four fish species

Now we will fit two IDMS, one with the two datasets combined, and another with an additional spatial field for the citizen science data. Both these models will have four shared fields, shared across the data sets (survey/citizen science), but separate for each fish species. Since we have two data sets and four species, that means that we in total have eight sub-models.

```

ISDMsetup <- startSpecies(
  survey,                               # Survey data
  artsobs,                               # Citizen science data
  Boundary = norway.poly,                # Boundary
  IPS = ipoints,                         # integration points
  spatialCovariates = cov_raster,        # Covariates
  speciesName = "species",               # The column containing species name
  speciesSpatial = 'replicate',          # Creates field per species
  speciesIntercept = FALSE,              # Turn random intercept off for species
  pointsIntercept = FALSE,               # Turn dataset intercept off
  responsePA = "occurrenceStatus",       # Name of response column
  pointsSpatial = NULL,                  # NULL since we use speciesSpatial
  Mesh = mesh,                           # inla mesh object
  Projection = proj                     # CRS for points and covariates
)

for (fish in fishes) {

  ISDMsetup$specifySpatial(Species = fish,
    prior.range = c(50, 0.01),
    prior.sigma = c(0.5, 0.01),
    constr = FALSE)

  ISDMsetup$priorsFixed(Effect = 'intercept',
    Species = fish,
    mean.linear = 0,
    prec.linear = 10)

}

```

```

ISDMsetup$specifyRandom(speciesGroup = list(model = "iid",
                                             hyper = list(prec = list(initial = 7.5, fixed = TRUE)))))

ISDMsetup$priorsFixed(Effect = 'HFP', mean.linear = 0, prec.linear = 0.1)
ISDMsetup$priorsFixed(Effect = 'SCI', mean.linear = 0, prec.linear = 0.1)

We may look at which terms are included in each of the eight sub-models by calling $updateFormula with
the data sets as the arguments.

ISDMsetup$updateFormula(datasetName = "survey")

## $Perca_fluviatilis
## occurrenceStatus ~ Perca_fluviatilis_SCI + Perca_fluviatilis_HFP +
##   speciesShared + Perca_fluviatilis_intercept
## <environment: 0x556c817ba070>
##
## $Salmo_trutta
## occurrenceStatus ~ Salmo_trutta_SCI + Salmo_trutta_HFP + speciesShared +
##   Salmo_trutta_intercept
## <environment: 0x556c817ba070>
##
## $Salvelinus_alpinus
## occurrenceStatus ~ Salvelinus_alpinus_SCI + Salvelinus_alpinus_HFP +
##   speciesShared + Salvelinus_alpinus_intercept
## <environment: 0x556c817ba070>
##
## $Esox_lucius
## occurrenceStatus ~ Esox_lucius_SCI + Esox_lucius_HFP + speciesShared +
##   Esox_lucius_intercept
## <environment: 0x556c817ba070>

ISDMsetup$updateFormula(datasetName = "artsobs")

## $Salvelinus_alpinus
## geometry ~ Salvelinus_alpinus_SCI + Salvelinus_alpinus_HFP +
##   speciesShared + Salvelinus_alpinus_intercept
## <environment: 0x556c817ba070>
##
## $Esox_lucius
## geometry ~ Esox_lucius_SCI + Esox_lucius_HFP + speciesShared +
##   Esox_lucius_intercept
## <environment: 0x556c817ba070>
##
## $Salmo_trutta
## geometry ~ Salmo_trutta_SCI + Salmo_trutta_HFP + speciesShared +
##   Salmo_trutta_intercept
## <environment: 0x556c817ba070>
##
## $Perca_fluviatilis
## geometry ~ Perca_fluviatilis_SCI + Perca_fluviatilis_HFP + speciesShared +
##   Perca_fluviatilis_intercept
## <environment: 0x556c817ba070>

ISDM <- fitISDM(ISDMsetup,
                  options = modelOptions)

```

We may then examine the model summary and save the model for future use.

```
summary(ISDM)

## Summary of 'modSpecies' object:
##
## inlabru version: 2.11.1
## INLA version: 24.06.27
##
## Types of data modelled:
##
## survey           Present absence
## artsobs         Present only
##
## Summary of the fixed effects for the species:
##
## Summary for Perca_fluviatilis:
##               mean        sd 0.025quant 0.5quant
## Perca_fluviatilis_intercept -3.63614002 0.24464951 -4.1144044 -3.63661988
## Perca_fluviatilis_HFP      0.42966914 0.03553123  0.3600757  0.42964035
## Perca_fluviatilis_SCI     -0.02556405 0.04409297 -0.1120373 -0.02556062
##                           0.975quant   mode      kld
## Perca_fluviatilis_intercept -3.15519542 -3.6366368 1.322877e-09
## Perca_fluviatilis_HFP      0.49942626  0.4296401 2.038013e-10
## Perca_fluviatilis_SCI     0.06088955 -0.0255606 4.477600e-11
##
## Summary for Salmo_trutta:
##               mean        sd 0.025quant 0.5quant
## Salmo_trutta_intercept -2.80056210 0.23593110 -3.26193875 -2.80099436
## Salmo_trutta_HFP       -0.01024675 0.03068344 -0.07044996 -0.01023452
## Salmo_trutta_SCI      0.01733201 0.02733480 -0.03627072  0.01733253
##                           0.975quant   mode      kld
## Salmo_trutta_intercept -2.33675512 -2.80100515 9.767316e-10
## Salmo_trutta_HFP       0.04988685 -0.01023445 8.207174e-11
## Salmo_trutta_SCI      0.07093177  0.01733254 4.965363e-11
##
## Summary for Salvelinus_alpinus:
##               mean        sd 0.025quant 0.5quant
## Salvelinus_alpinus_intercept -3.2292040 0.23972451 -3.69785519 -3.2296798
## Salvelinus_alpinus_HFP      -0.2307107 0.05961685 -0.34781059 -0.2306428
## Salvelinus_alpinus_SCI     0.1510587 0.04334595  0.06606403  0.1510577
##                           0.975quant   mode      kld
## Salvelinus_alpinus_intercept -2.7578889 -3.2296948 1.237773e-09
## Salvelinus_alpinus_HFP      -0.1139964 -0.2306422 3.621611e-10
## Salvelinus_alpinus_SCI     0.2360588  0.1510577 5.267180e-11
##
## Summary for Esox_lucius:
##               mean        sd 0.025quant 0.5quant 0.975quant
## Esox_lucius_intercept -3.7321674 0.24564729 -4.2123655 -3.7326501 -3.2492759
## Esox_lucius_HFP        0.5221459 0.04080591  0.4422217  0.5221128  0.6022584
## Esox_lucius_SCI       -0.1535054 0.05850085 -0.2682498 -0.1534956 -0.0388172
##                           mode      kld
## Esox_lucius_intercept -3.7326679 1.381858e-09
## Esox_lucius_HFP        0.5221125 2.070918e-10
## Esox_lucius_SCI       -0.1534955 5.062795e-11
```

```

## Time used:
##      Pre = 1.16, Running = 54.3, Post = 1.72, Total = 57.2
## Random effects:
##   Name      Model
##   speciesShared SPDE2 model
##
## Model hyperparameters:
##               mean     sd 0.025quant 0.5quant 0.975quant    mode
## Range for speciesShared 192.00 7.49      177.66   191.86      207.14 191.60
## Stdev for speciesShared 62.51 1.71      59.20    62.49      65.93  62.46
##
## Deviance Information Criterion (DIC) .....: -20071.47
## Deviance Information Criterion (DIC, saturated) ....: -20091.93
## Effective number of parameters .....: -41648.20
##
## Watanabe-Akaike information criterion (WAIC) ....: 26934.36
## Effective number of parameters .....: 2855.65
##
## Marginal log-Likelihood: -33686.56
## is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
saveRDS(ISDM, "results/ISDMModel.rds")

```

This model has the species specific spatial fields, but we also want a bias field that is shared across the species. We add this using `$addBias`.

```

ISDMsetupBias <- startSpecies(
  survey,
  artobs,
  Boundary = norway.poly,
  IPS = ipoints,
  spatialCovariates = cov_raster,
  speciesName = "species",
  speciesSpatial = 'replicate',
  speciesIntercept = FALSE,
  pointsIntercept = FALSE,
  responsePA = "occurrenceStatus",
  pointsSpatial = NULL,
  Mesh = mesh,
  Projection = proj
)

for (fish in fishes) {

  ISDMsetupBias$specifySpatial(Species = fish,
                                prior.range = c(50, 0.01),
                                prior.sigma = c(0.5, 0.01),
                                constr = FALSE)

  ISDMsetupBias$priorsFixed(Effect = 'intercept',
                            Species = fish,
                            mean.linear = 0,
                            prec.linear = 10)
}

```

```

}

ISDMsetupBias$specifyRandom(speciesGroup = list(model = "iid",
    hyper = list(prec = list(initial = 7.5, fixed = TRUE)))))

ISDMsetupBias$priorsFixed(Effect = 'HFP', mean.linear = 0, prec.linear = 0.1)
ISDMsetupBias$priorsFixed(Effect = 'SCI', mean.linear = 0, prec.linear = 0.1)

ISDMsetupBias$addBias("artsobs", copyModel = FALSE)
ISDMsetupBias$specifySpatial(Bias = 'artsobs',
    prior.range = c(50, 0.01),
    prior.sigma = c(0.5, 0.01))

ISDMbias <- fitISDM(ISDMsetupBias,
    options = modelOptions)

summary(ISDMbias)

## Summary of 'modSpecies' object:
##
## inlabru version: 2.11.1
## INLA version: 24.06.27
##
## Types of data modelled:
##
## survey           Present absence
## artsobs          Present only
##
## Summary of the fixed effects for the species:
##
## Summary for Perca_fluviatilis:
##               mean        sd 0.025quant   0.5quant
## Perca_fluviatilis_intercept -1.3129022 0.17516037 -1.6555362 -1.31321525
## Perca_fluviatilis_HFP      0.5985722 0.03783207  0.5244817  0.59853842
## Perca_fluviatilis_SCI     -0.0340006 0.04998037 -0.1320282 -0.03399357
##                           0.975quant      mode      kld
## Perca_fluviatilis_intercept -0.96848783 -1.31321737 8.103272e-10
## Perca_fluviatilis_HFP      0.67285466  0.59853813 2.448363e-10
## Perca_fluviatilis_SCI     0.06398697 -0.03399354 5.272311e-11
##
## Summary for Salmo_trutta:
##               mean        sd 0.025quant   0.5quant 0.975quant
## Salmo_trutta_intercept 0.46227893 0.16144748 0.14502900 0.46248926 0.77833201
## Salmo_trutta_HFP       0.11594464 0.03406912 0.04916634 0.11593450 0.18278067
## Salmo_trutta_SCI       0.01515909 0.03589991 -0.05524437 0.01516146 0.08554903
##                           mode      kld
## Salmo_trutta_intercept 0.46249039 4.410009e-10
## Salmo_trutta_HFP       0.11593445 6.745799e-11
## Salmo_trutta_SCI       0.01516147 5.066810e-11
##
## Summary for Salvelinus_alpinus:
##               mean        sd 0.025quant   0.5quant
## Salvelinus_alpinus_intercept -0.54326762 0.16302590 -0.86240526 -0.54348021

```

```

## Salvelinus_alpinus_HFP      -0.07409401 0.05395860 -0.18005436 -0.07404141
## Salvelinus_alpinus_SCI      0.14635085 0.04954849  0.04919879  0.14634805
##
##                               0.975quant      mode      kld
## Salvelinus_alpinus_intercept -0.22292029 -0.54348138 4.377793e-10
## Salvelinus_alpinus_HFP       0.03156738 -0.07404101 2.772529e-10
## Salvelinus_alpinus_SCI       0.24351883  0.14634804 5.338874e-11
##
## Summary for Esox_lucius:
##                               mean      sd 0.025quant 0.5quant 0.975quant
## Esox_lucius_intercept -1.5376400 0.17784934 -1.8852775 -1.5380455 -1.187699965
## Esox_lucius_HFP        0.6807987 0.04072957  0.6009771  0.6807821  0.760714118
## Esox_lucius_SCI        -0.1254082 0.06183941 -0.2467023 -0.1253969 -0.004178569
##                               mode      kld
## Esox_lucius_intercept -1.5380492 1.315819e-09
## Esox_lucius_HFP        0.6807820 8.148945e-11
## Esox_lucius_SCI        -0.1253969 5.846177e-11
##
## Time used:
##     Pre = 1.53, Running = 1138, Post = 5.59, Total = 1145
## Random effects:
##   Name      Model
## speciesShared SPDE2 model
## artsobs_biasField SPDE2 model
##
## Model hyperparameters:
##                               mean      sd 0.025quant 0.5quant 0.975quant      mode
## Range for speciesShared    224.89  21.742     184.82   223.96    270.34 222.32
## Stdev for speciesShared    24.51   1.233      22.17   24.48     27.03 24.43
## Range for artsobs_biasField 230.15  15.972     200.91   229.40    263.77 227.46
## Stdev for artsobs_biasField  3.66   0.222      3.25    3.65     4.13  3.63
##
## Deviance Information Criterion (DIC) .....: -36392.50
## Deviance Information Criterion (DIC, saturated) ....: -36412.97
## Effective number of parameters .....: -42600.01
##
## Watanabe-Akaike information criterion (WAIC) ....: 12046.15
## Effective number of parameters .....: 2887.54
##
## Marginal log-Likelihood: -25798.06
## is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
saveRDS(ISDMbias, "results/ISDMBiasModel.rds")

```

## Predictions and plots

Once the model has been fit, we can look at the predictions from the species-specific shared fields and the bias field.

First we create a plot to show the coefficients and the associated credibility intervals for the four different models considered. The estimates and standard errors for the four models are relatively similar.

```
makeData <- function(data, name) {
```

```

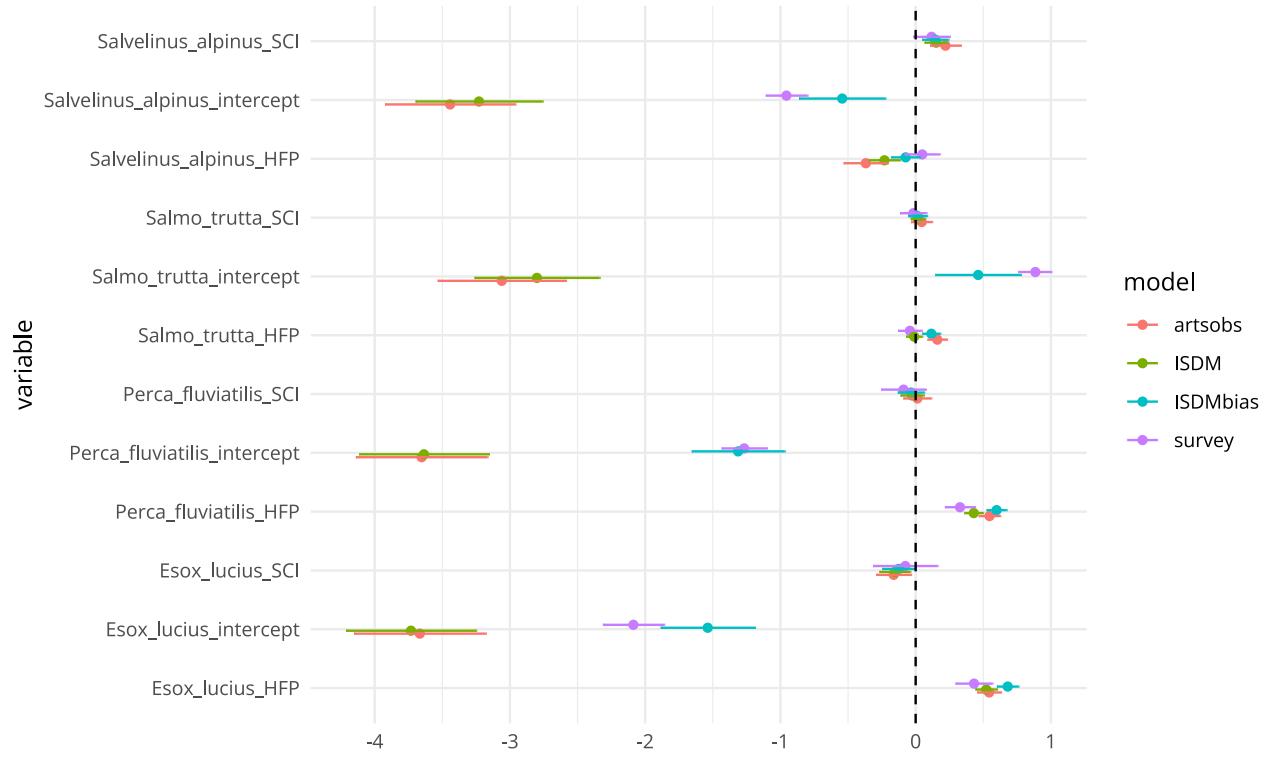
data$summary.fixed$variable <- rownames(data$summary.fixed)
data$summary.fixed$min <- data$summary.fixed$`0.025quant`
data$summary.fixed$max <- data$summary.fixed$`0.975quant`
data$summary.fixed$model <- name
data$summary.fixed

}

plotData <- rbind(makeData(surveyFit, 'survey'),
                  makeData(artsobsFit, 'artsobs'),
                  makeData(ISDM, 'ISDM'),
                  makeData(ISDMbias, 'ISDMbias'))

ggplot(plotData, aes(x = mean, y = variable, col = model)) +
  geom_point(position=position_dodge(width=0.2)) +
  geom_errorbar(aes(x = mean, y = variable, xmin = min, xmax = max),
                position=position_dodge(width=0.2), width = 0) +
  geom_vline(xintercept = 0, lty = 2) +
  theme_minimal() +
  theme(text = element_text(family = f1),
        strip.text = element_text(family = f1, size = 12, face = "italic"),
        plot.tag = element_text(angle = 90, hjust = 0.5),
        plot.tag.position = c(-0.03, 0.45))

```



```
ggsave("figures/CIplots.png")
```

```
## Saving 8 x 5 in image
```

```

ggsave("figures/CIplots.pdf")

## Saving 8 x 5 in image

We define a function that will do species-specific predictions for the ISDM with a bias field, and save the species predictions, since these take a little time to compute.

predict_species <- function(model, predict_data){
  sharedfield <- predict(model,
                         data = predict_data,
                         spatial = TRUE,
                         n.samples = 1000)
  file_name <- paste0("results/sharedfield", ".rds")
  saveRDS(sharedfield, file_name)
  return(sharedfield)
}

predData <- fm_pixels(mesh = mesh, mask = norway.poly, dims = c(450, 450))

predictionFish <- predict_species(
  model = ISDMbias,
  predict_data = predData
)

saveRDS(predictionFish, "results/Predictionlist.rds")

```

Once we have the predictions, we can make some plots. We similarly define a function that makes a plot for one species, and then run this for all four species.

```

NO <- st_boundary(st_as_sf(norway.poly))

plot_preferences <- list(scale_color_distiller(palette = "BrBG", direction = 1),
                          coord_sf(),
                          xlab(""),
                          ylab(""),
                          scale_x_continuous(breaks = c(5, 25)),
                          scale_y_continuous(breaks = c(60, 68)),
                          theme_minimal(),
                          theme(text = element_text(family = f1),
                                axis.text=element_text(size=12),
                                legend.text = element_text(size = 12),
                                title = element_text(family = f1, size = 15, face = "italic"),
                                legend.key.height = unit(0.3, "cm"),
                                legend.title = element_blank(),
                                legend.position = "bottom"))
)

plot_species <- function(predictions, species_to_plot, plot_preferences){

  p <- ggplot() +
    gg(predictions,
       aes(col = mean)) +
    gg(NO, lwd = 0.8) +
    labs(title = sub("_", " ", species_to_plot)) +
    plot_preferences
  return(p)
}

```

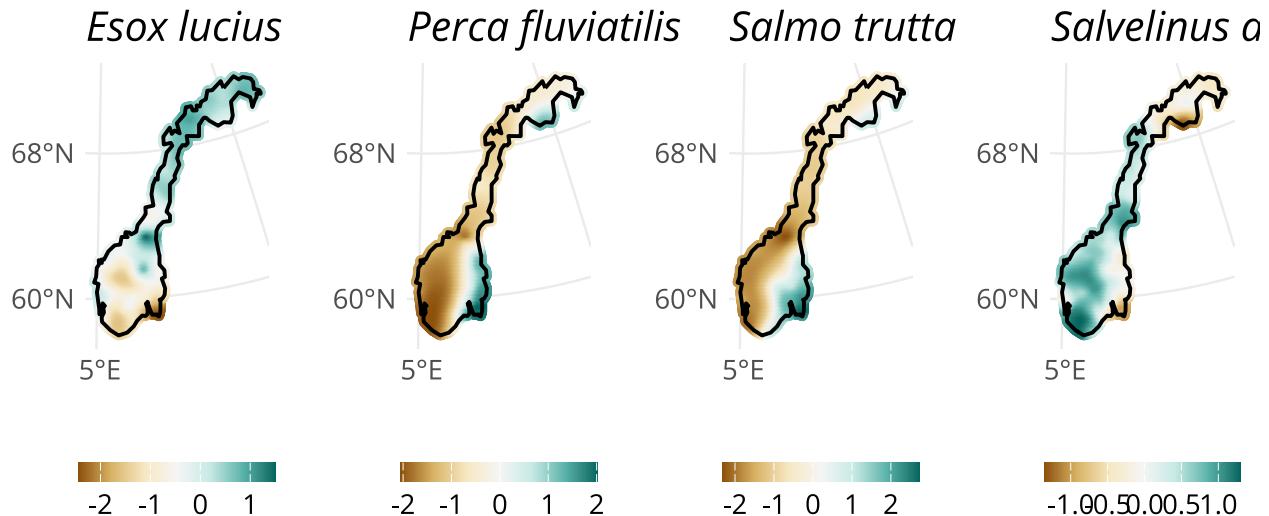
```

}

plot_list <- list()
for(fish in fishes) {
  plot_list[[fish]] <- plot_species(
    predictions = predictionFish$speciesPredictions[[fish]],
    species = fish,
    plot_preferences = plot_preferences)
}

patchwork::wrap_plots(plot_list, nrow = 1)

```



```

ggsave("figures/fishplot_four_species.pdf", height = 3*2, width = 8*2)
ggsave("figures/fishplot_four_species.png", height = 3*2, width = 8*2)

```

And finally we predict and plot the bias field, which is shared between all the fish, as it describes the human sampling more than the distribution of the fish.

```

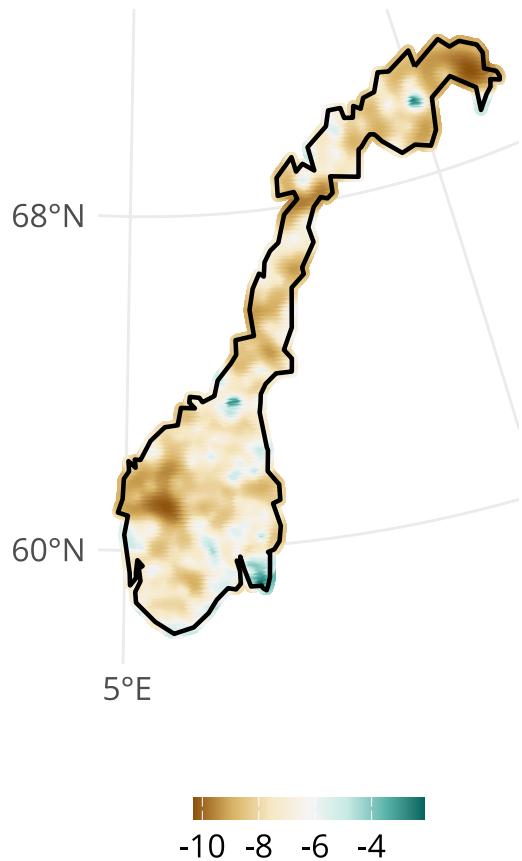
fish_biasfield <- predict(ISDMbias,
                           data = predData,
                           bias = TRUE,
                           n.samples = 1000)

saveRDS(fish_biasfield, "results/ISDMbiasfield.rds")
fish_biasfield <- readRDS("results/ISDMbiasfield.rds")

ggplot() +
  gg(fish_biasfield$biasFields$artsobs,
     aes(col = mean)) +
  gg(NO, lwd = 0.8) +
  labs(title = "Bias field") +
  plot_preferences

```

## Bias field



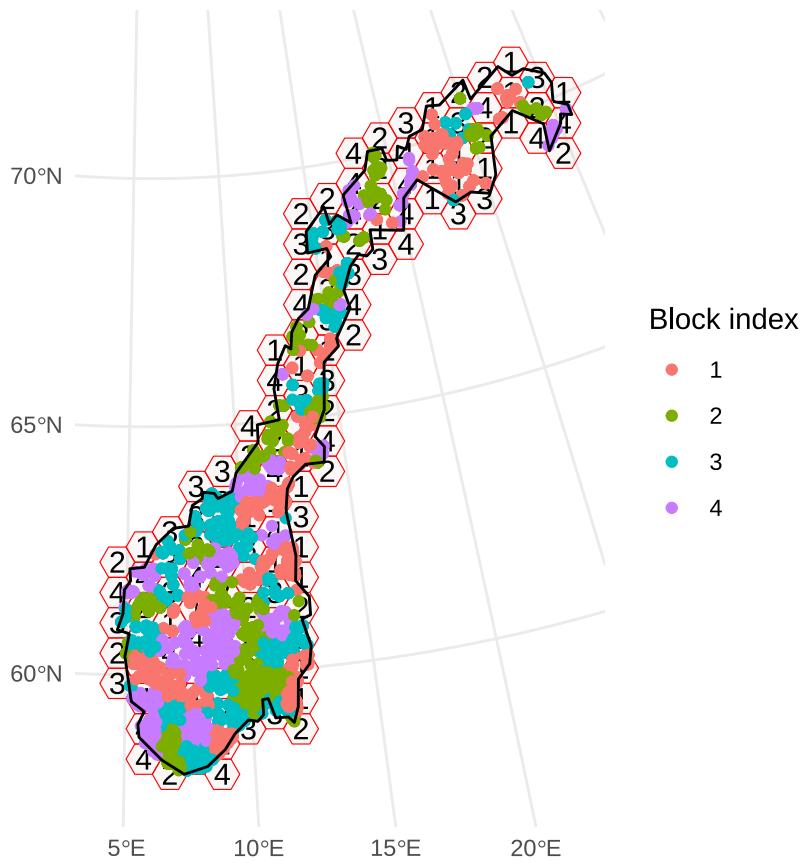
```
ggsave("figures/ISDMbiasfield.pdf", width = 3*2, height = 3*2)
```

## Cross-validation

We will use a spatial-block cross-validation technique to show that the ISDM with a second spatial effect performs better than the model with only one. Here, we grid our map into 4 separate spatial blocks, which is illustrated below. We then fit a model with all blocks except one, and calculate the DIC score. The model with the lowest DIC is deemed best. To do this, we first set up the spatial block using `.$spatialBlock`.

```
ISDMsetup$spatialBlock(k = 4, rows_cols = c(15,15),  
                        flat_top = TRUE, plot = TRUE)
```

## Plot of the blocked data



```
ISDMsetupBias$spatialBlock(k = 4, rows_cols = c(15,15),
                             flat_top = TRUE, plot = FALSE)
```

And then fit the models using `blockedCV`:

```
ISDMBlock <- blockedCV(ISDMsetup,
                         options = modelOptions)

ISDMBiasBlock <- blockedCV(ISDMsetupBias,
                            options = modelOptions)
```

The results are then given as:

```
print('ISDM results:')

## [1] "ISDM results:"
```

ISDMBlock

```
## Spatial block cross-validation score:
##
```

```
## Formula: ~speciesShared + Esox_lucius_intercept + Perca_fluviatilis_intercept + Salmo_trutta_intercept
```

```
##
```

	mean.deviance	p.eff	dic
## fold 1	16352.60	-31699.12	-15346.52
## fold 2	16777.48	-28571.97	-11794.49
## fold 3	16770.06	-32628.49	-15858.43
## fold 4	16497.39	-28834.23	-12336.84

```

##  

## mean DIC score: -13834.07  

print('ISDM bias results:')  

## [1] "ISDM bias results:"  

ISDMBiasBlock  

## Spatial block cross-validation score:  

##  

## Formula: ~speciesShared + Esox_lucius_intercept + Perca_fluviatilis_intercept + Salmo_trutta_intercept  

##  

##      mean.deviance    p.eff      dic  

## fold 1      4872.785 -32470.49 -27597.70  

## fold 2      4626.105 -29283.33 -24657.22  

## fold 3      5099.961 -32942.88 -27842.92  

## fold 4      4887.071 -29342.72 -24455.65  

##  

## mean DIC score: -26138.37

```

We next perform a spatial-block cross-validation to assess whether the ISDM models fit the data better than the single dataset models. To do this, we block our study area (as before), and fit a model (training model) using every combination of datasets we have in all block combinations, with one block left out. We then use the training model to predict onto the survey data in the block that was left out, and use these predictions as an offset in a new model (testing model) with the left out data. After fitting the testing data, we obtain the marginal likelihood as the measure of fit. The model with the highest marginal likelihood values provides the best fit.

To do this, we use the `blockedCV` function again, but specify `method = 'Predict'`. We then specify the name of the prediction dataset using `predictName`. To include the model with the bias field, we have to set up a second call of `blockedCV`, but this time specifying `datasetCombs = list(c('survey', 'artsobs'))`, meaning that we are only fitting a model using that combination of datasets (in this case, the ISDM with a bias field).

```

ISDMPredBlock <- blockedCV(data = ISDMsetup,
                           options = modelOptions,
                           method = 'Predict',
                           predictName = 'survey')

ISDMPredBiasBlock <- blockedCV(data = ISDMsetupBias,
                                 options = modelOptions,
                                 datasetCombs = list(c('survey', 'artsobs')),
                                 method = 'Predict',
                                 predictName = 'survey')

```

The results are given as follows. The first column represents the name of the datasets used in the training model, and the folds represent the marginal likelihood obtained from the test model.

```

ISDMPredBlock  

## Spatial block cross-validation score:  

##  

## Log marginal likelihoods obtained by using the predictions from the model as an offset:  

##  

##      fold 1      fold 2      fold 3      fold 4      mean  

## survey       -370.7335 -308.9310 -305.2366 -483.4931 -367.0986  

## artsobs      -342.1359 -340.5830 -265.3669 -383.4564 -332.8855  

## survey and artsobs -311.5939 -288.0785 -257.1512 -363.8219 -305.1614

```

## ISDMPredBiasBlock

```
## Spatial block cross-validation score:  
##  
## Log marginal likelihoods obtained by using the predictions from the model as an offset:  
##  
##          fold 1    fold 2    fold 3    fold 4      mean  
## survey and artsobs -281.3648 -206.1506 -246.3698 -319.1175 -263.2507
```

## Session info and runtime

This document took 3.9 hours to compile.

Current session info

```
- Session info -----  
setting  value  
version R version 4.4.1 (2024-06-14)  
os       Ubuntu 20.04.1 LTS  
system   x86_64, linux-gnu  
ui       X11  
language (EN)  
collate  C.UTF-8  
ctype    C.UTF-8  
tz       Europe/Oslo  
date     2024-08-12  
pandoc   2.5 @ /usr/bin/ (via rmarkdown)
```

```
- Packages -----  
package      * version    date (UTC) lib source  
base64enc     0.1-3      2015-07-28 [1] CRAN (R 4.3.2)  
blockCV        3.1-3      2023-06-04 [1] CRAN (R 4.3.1)  
class         7.3-21     2023-01-23 [1] CRAN (R 4.2.2)  
classInt       0.4-7      2022-06-10 [1] CRAN (R 4.2.1)  
cli            3.6.1      2023-03-23 [1] CRAN (R 4.3.1)  
clipr           0.8.0      2022-02-22 [1] CRAN (R 4.3.2)  
codetools      0.2-20     2024-03-31 [1] CRAN (R 4.4.0)  
colorspace      2.0-3      2022-02-21 [1] CRAN (R 4.1.2)  
countrycode    1.6.0      2024-03-22 [1] CRAN (R 4.4.0)  
curl            4.3.3      2022-10-06 [1] CRAN (R 4.2.2)  
DBI             1.2.3      2024-06-02 [1] CRAN (R 4.4.0)  
desc            1.4.3      2023-12-10 [1] CRAN (R 4.3.2)  
details         0.3.0      2022-03-27 [1] CRAN (R 4.3.2)  
digest          0.6.31     2022-12-11 [1] CRAN (R 4.2.2)  
dplyr          * 1.1.3     2023-09-03 [1] CRAN (R 4.3.1)  
e1071           1.7-11     2022-06-07 [1] CRAN (R 4.2.1)  
evaluate        0.24.0     2024-06-10 [1] CRAN (R 4.4.0)  
fansi            1.0.2      2022-01-14 [1] CRAN (R 4.1.2)  
farver           2.1.1      2022-07-06 [1] CRAN (R 4.2.1)  
fastmap          1.1.1      2023-02-24 [1] CRAN (R 4.3.1)  
fmesher          * 0.1.5     2023-12-20 [1] CRAN (R 4.3.2)  
generics         0.1.3      2022-07-05 [1] CRAN (R 4.3.2)  
geojsonsf        2.0.3      2022-05-30 [1] CRAN (R 4.3.2)  
ggplot2          * 3.5.1     2024-04-23 [1] CRAN (R 4.4.0)
```

giscoR	* 0.5.0	2024-05-29	[1]	CRAN	(R 4.4.0)
glue	1.6.2	2022-02-24	[1]	CRAN	(R 4.2.1)
gttable	0.3.5	2024-04-22	[1]	CRAN	(R 4.4.0)
highr	0.11	2024-05-26	[1]	CRAN	(R 4.4.0)
htmltools	0.5.4	2022-12-07	[1]	CRAN	(R 4.2.2)
httr	1.4.7	2023-08-15	[1]	CRAN	(R 4.3.1)
INLA	* 24.06.27	2024-06-27	[1]	local	
inlabru	* 2.11.1	2024-07-01	[1]	CRAN	(R 4.4.1)
jsonlite	1.7.3	2022-01-17	[1]	CRAN	(R 4.1.2)
KernSmooth	2.23-22	2023-07-10	[1]	CRAN	(R 4.3.1)
knitr	1.45	2023-10-30	[1]	CRAN	(R 4.3.2)
labeling	0.4.3	2023-08-29	[1]	CRAN	(R 4.3.1)
lattice	0.20-45	2021-09-22	[1]	CRAN	(R 4.2.1)
lifecycle	1.0.4	2023-11-07	[1]	CRAN	(R 4.3.2)
magrittr	2.0.3	2022-03-30	[1]	CRAN	(R 4.4.0)
mapproj	* 1.2.8	2022-01-12	[1]	CRAN	(R 4.1.2)
maps	* 3.4.0	2021-09-25	[1]	CRAN	(R 4.1.2)
Matrix	* 1.7-0	2024-04-26	[1]	CRAN	(R 4.4.0)
MatrixModels	0.5-3	2023-11-06	[1]	CRAN	(R 4.4.0)
mnormt	2.1.0	2022-06-07	[1]	CRAN	(R 4.2.1)
munsell	0.5.1	2024-04-01	[1]	CRAN	(R 4.3.3)
numDeriv	2016.8-1.1	2019-06-06	[1]	CRAN	(R 4.3.2)
patchwork	* 1.2.0	2024-01-08	[1]	CRAN	(R 4.3.2)
pillar	1.9.0	2023-03-22	[1]	CRAN	(R 4.3.1)
pkgconfig	2.0.3	2019-09-22	[1]	CRAN	(R 4.3.2)
plyr	1.8.8	2022-11-11	[1]	CRAN	(R 4.2.2)
png	0.1-8	2022-11-29	[1]	CRAN	(R 4.2.2)
PointedSDMs	* 2.1.1	2024-08-12	[1]	Github	(PhilipMostert/PointedSDMs@34bd34b)
proxy	0.4-27	2022-06-09	[1]	CRAN	(R 4.2.1)
R.devices	* 2.17.2	2024-01-29	[1]	CRAN	(R 4.3.2)
R.methodsS3	1.8.2	2022-06-13	[1]	CRAN	(R 4.3.2)
R.oo	1.26.0	2024-01-24	[1]	CRAN	(R 4.3.2)
R.utils	2.12.3	2023-11-18	[1]	CRAN	(R 4.3.2)
R6	* 2.5.1	2021-08-19	[1]	CRAN	(R 4.3.2)
ragg	1.2.7	2023-12-11	[1]	CRAN	(R 4.3.2)
rappdirs	0.3.3	2021-01-31	[1]	CRAN	(R 4.3.2)
raster	* 3.5-11	2021-12-23	[1]	CRAN	(R 4.1.2)
RColorBrewer	1.1-3	2022-04-03	[1]	CRAN	(R 4.3.2)
Rcpp	1.0.8	2022-01-13	[1]	CRAN	(R 4.1.2)
rlang	1.1.1	2023-04-28	[1]	CRAN	(R 4.3.1)
rmarkdown	2.27	2024-05-17	[1]	CRAN	(R 4.4.0)
s2	1.1.6	2023-12-19	[1]	CRAN	(R 4.3.2)
scales	1.3.0	2023-11-28	[1]	CRAN	(R 4.3.2)
sessioninfo	1.2.2	2021-12-06	[1]	CRAN	(R 4.3.2)
sf	* 1.0-9	2022-11-08	[1]	CRAN	(R 4.2.2)
showtext	* 0.9-7	2024-03-02	[1]	CRAN	(R 4.4.0)
showtextdb	* 3.0	2020-06-04	[1]	CRAN	(R 4.3.1)
sn	2.1.1	2023-04-04	[1]	CRAN	(R 4.3.1)
sp	* 2.0-0	2023-06-22	[1]	CRAN	(R 4.3.1)
sysfonts	* 0.8.9	2024-03-02	[1]	CRAN	(R 4.4.0)
systemfonts	1.1.0	2024-05-15	[1]	CRAN	(R 4.4.0)
terra	1.6-47	2022-12-02	[1]	CRAN	(R 4.2.2)
textshaping	0.4.0	2024-05-24	[1]	CRAN	(R 4.4.0)

```
tibble      3.2.1    2023-03-20 [1] CRAN (R 4.3.1)
tidyselect   1.2.1    2024-03-11 [1] CRAN (R 4.3.3)
tinytex      0.51     2024-05-06 [1] CRAN (R 4.4.0)
units        0.8-0    2022-02-05 [1] CRAN (R 4.1.2)
utf8         1.2.2    2021-07-24 [1] CRAN (R 4.2.1)
vctrs        0.6.4    2023-10-12 [1] CRAN (R 4.3.1)
withr        3.0.1    2024-07-31 [1] CRAN (R 4.4.1)
wk           0.6.0    2022-01-03 [1] CRAN (R 4.1.2)
xfun         0.40     2023-08-09 [1] CRAN (R 4.3.1)
xml2        1.3.3    2021-11-30 [1] CRAN (R 4.1.2)
yaml         2.3.7    2023-01-23 [1] CRAN (R 4.2.2)

[1] /home/ahomec/p/philism/R/x86_64-pc-linux-gnu-library/4.1
[2] /usr/local/lib/R/site-library
[3] /usr/lib/R/site-library
[4] /usr/lib/R/library
```

---