

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: October 24 2023 at 10:10:11.

Setup

These are the packages we will be using:

```
# devtools::install_github("PhilipMostert/PointedSDMs")
library(PointedSDMs) # model fitting
library(ggplot2)      # plotting
library(raster)       # ??? Model fitting fails if I exclude this
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(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

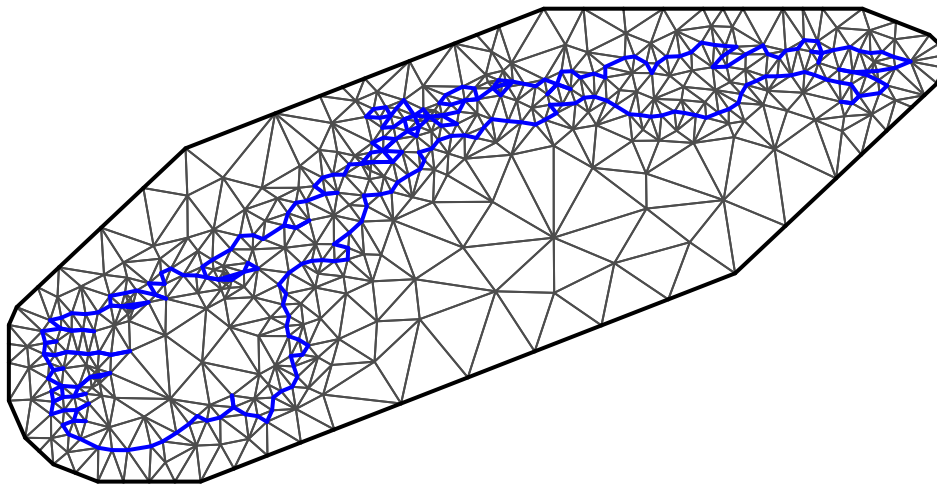
We begin by making a spatial mesh out of a map of Norway.

```
proj <- '+proj=utm +zone=32 +datum=WGS84 +units=m +no_defs'
norwayfill <- maps::map("world", "norway", fill=TRUE, plot=FALSE,
                        ylim=c(58,72), xlim=c(4,32))
IDs <- sapply(strsplit(norwayfill$names, "."), function(x) x[1])
norway.poly <- maptools::map2SpatialPolygons(norwayfill, IDs = IDs,
                                              proj4string = CRS(proj))
```

Adjusting the mesh to be coarser is the easiest way to decrease the run-time for the model. With the following mesh, the model fitting takes a little more than 1 hour for us, but feel free to change the `max.edge` or `cutoff` to get a coarser mesh.

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

Constrained refined Delaunay triangulation



Setting up covariate data

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

```
covariates_raw <- readRDS("data/environmental_covariates.RDS")
covariates <- covariates_raw %>%
  # Log-transform area of lake
```

```

dplyr::mutate(log_area = log(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("log_area", "eurolst_bio10", "SCI")

cov_pixel <- SpatialPixelsDataFrame(
  points = covariates[,c("decimalLongitude", "decimalLatitude")],
  data = covariates[,Use],
  proj4string = CRS(proj),
  tol = 0.340571)

# Scale covariates and convert to terra::rast
cov_raster <- scale(terra::rast(cov_pixel))

```

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)
artsobs <- readRDS("data/artsobs_clean.rds") %>%
  filter(species %in% fishes)

```

We can plot the observed data points:

```

showtext_auto()
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),

```

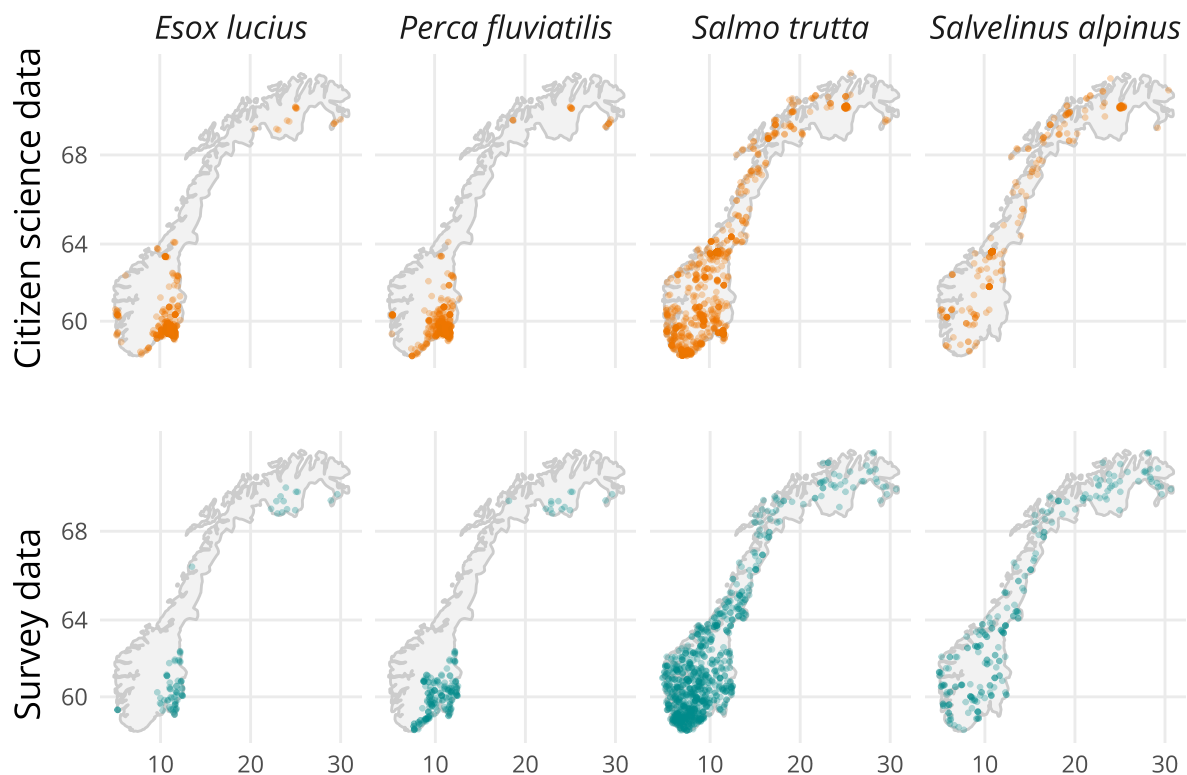
```

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)
```

Joint model for four fish species

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 $\xi_j(s)$:

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

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

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 $\xi_j(s)$, plus an additional spatial field $\xi_{\text{bias}}(s)$ that is unique to the citizen science data, but shared across all fish species:

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

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

So in summary, for this model we have:

- one bias field, based on CS data and common across all species
- 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.

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

```
fish_model_setup <- intModel(
  survey,                # Survey data
  artsobs,               # Citizen science data
  spatialCovariates = cov_raster, # Covariates
  speciesName = "species", # The column containing species name
  speciesSpatial = "copy",  # Copy species fields across data
  Coordinates = c("decimalLongitude", "decimalLatitude"), # Name of coords
  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
```

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`.

```
fish_model_setup$addBias("artsobs")
```

For the species specific fields, the default in PointedSDMs is that these are allowed to be different up to a scaling factor (named beta in INLA) across the data sets. That means that for instance the trout-specific spatial field for the citizen science data set is equal to beta times the trout-specific spatial field for the survey data. In practice, this has to do with the copy-option in INLA. But in our model we want them to be the same, not to vary by a factor. So we manually change this using the `$changeComponents` function, by setting `hyper = list(beta = list(fixed = TRUE))` for each of the four citizen science fields.

```

fish_model_setup$changeComponents(
  'Perca_fluviatilis_artsobs_spatial(
    main = geometry,
    copy = "Perca_fluviatilis_survey_spatial",
    hyper = list(beta = list(fixed = TRUE)))',
  print = FALSE)

fish_model_setup$changeComponents(
  'Salmo_trutta_artsobs_spatial(
    main = geometry,
    copy = "Salmo_trutta_survey_spatial",
    hyper = list(beta = list(fixed = TRUE)))',
  print = FALSE)

fish_model_setup$changeComponents(
  'Salvelinus_alpinus_artsobs_spatial(
    main = geometry,
    copy = "Salvelinus_alpinus_survey_spatial",
    hyper = list(beta = list(fixed = TRUE)))',
  print = FALSE)

fish_model_setup$changeComponents(
  'Esox_lucius_artsobs_spatial(
    main = geometry,
    copy = "Esox_lucius_survey_spatial",
    hyper = list(beta = list(fixed = TRUE)))')

```

Components:

```

## --1 + Perca_fluviatilis_survey_spatial(main = geometry, model = Perca_fluviatilis_survey_field) +
##   Salmo_trutta_survey_spatial(main = geometry, model = Salmo_trutta_survey_field) +
##   Salvelinus_alpinus_survey_spatial(main = geometry, model = Salvelinus_alpinus_survey_field) +
##   Esox_lucius_survey_spatial(main = geometry, model = Esox_lucius_survey_field) +
##   Perca_fluviatilis_log_area(main = Perca_fluviatilis_log_area,
##     model = "linear") + Salmo_trutta_log_area(main = Salmo_trutta_log_area,
##     model = "linear") + Salvelinus_alpinus_log_area(main = Salvelinus_alpinus_log_area,
##     model = "linear") + Esox_lucius_log_area(main = Esox_lucius_log_area,
##     model = "linear") + Perca_fluviatilis_eurolst_bio10(main = Perca_fluviatilis_eurolst_bio10,
##     model = "linear") + Salmo_trutta_eurolst_bio10(main = Salmo_trutta_eurolst_bio10,
##     model = "linear") + Salvelinus_alpinus_eurolst_bio10(main = Salvelinus_alpinus_eurolst_bio10,
##     model = "linear") + Esox_lucius_eurolst_bio10(main = Esox_lucius_eurolst_bio10,
##     model = "linear") + Perca_fluviatilis_SCI(main = Perca_fluviatilis_SCI,
##     model = "linear") + Salmo_trutta_SCI(main = Salmo_trutta_SCI,
##     model = "linear") + Salvelinus_alpinus_SCI(main = Salvelinus_alpinus_SCI,
##     model = "linear") + Esox_lucius_SCI(main = Esox_lucius_SCI,
##     model = "linear") + Perca_fluviatilis_intercept(1) + Salmo_trutta_intercept(1) +
##     Salvelinus_alpinus_intercept(1) + Esox_lucius_intercept(1) +
##     artsobs_biasField(main = geometry, model = artsobs_bias_field) +
##     Perca_fluviatilis_artsobs_spatial(main = geometry, copy = "Perca_fluviatilis_survey_spatial",
##       hyper = list(beta = list(fixed = TRUE))) + Salmo_trutta_artsobs_spatial(main = geometry,
##       copy = "Salmo_trutta_survey_spatial", hyper = list(beta = list(fixed = TRUE))) +
##     Salvelinus_alpinus_artsobs_spatial(main = geometry, copy = "Salvelinus_alpinus_survey_spatial",
##       hyper = list(beta = list(fixed = TRUE))) + Esox_lucius_artsobs_spatial(main = geometry,
##       copy = "Esox_lucius_survey_spatial", hyper = list(beta = list(fixed = TRUE)))

```

```
## <environment: 0x5558cb804730>
```

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.

```
fish_model_setup$updateFormula(datasetName = "survey")
```

```
## $Perca_fluviatilis
## occurrenceStatus ~ Perca_fluviatilis_log_area + Perca_fluviatilis_eurolst_bio10 +
##   Perca_fluviatilis_SCI + Perca_fluviatilis_intercept + Perca_fluviatilis_survey_spatial
## <environment: 0x5558d16be740>
##
## $Salmo_trutta
## occurrenceStatus ~ Salmo_trutta_log_area + Salmo_trutta_eurolst_bio10 +
##   Salmo_trutta_SCI + Salmo_trutta_intercept + Salmo_trutta_survey_spatial
## <environment: 0x5558d16be740>
##
## $Salvelinus_alpinus
## occurrenceStatus ~ Salvelinus_alpinus_log_area + Salvelinus_alpinus_eurolst_bio10 +
##   Salvelinus_alpinus_SCI + Salvelinus_alpinus_intercept + Salvelinus_alpinus_survey_spatial
## <environment: 0x5558d16be740>
##
## $Esox_lucius
## occurrenceStatus ~ Esox_lucius_log_area + Esox_lucius_eurolst_bio10 +
##   Esox_lucius_SCI + Esox_lucius_intercept + Esox_lucius_survey_spatial
## <environment: 0x5558d16be740>
```

```
fish_model_setup$updateFormula(datasetName = "artsobs")
```

```
## $Salvelinus_alpinus
## geometry ~ Salvelinus_alpinus_log_area + Salvelinus_alpinus_eurolst_bio10 +
##   Salvelinus_alpinus_SCI + Salvelinus_alpinus_intercept + Salvelinus_alpinus_artsobs_spatial +
##   artsobs_biasField
## <environment: 0x5558d16be740>
##
## $Esox_lucius
## geometry ~ Esox_lucius_log_area + Esox_lucius_eurolst_bio10 +
##   Esox_lucius_SCI + Esox_lucius_intercept + Esox_lucius_artsobs_spatial +
##   artsobs_biasField
## <environment: 0x5558d16be740>
##
## $Salmo_trutta
## geometry ~ Salmo_trutta_log_area + Salmo_trutta_eurolst_bio10 +
##   Salmo_trutta_SCI + Salmo_trutta_intercept + Salmo_trutta_artsobs_spatial +
##   artsobs_biasField
## <environment: 0x5558d16be740>
##
## $Perca_fluviatilis
## geometry ~ Perca_fluviatilis_log_area + Perca_fluviatilis_eurolst_bio10 +
##   Perca_fluviatilis_SCI + Perca_fluviatilis_intercept + Perca_fluviatilis_artsobs_spatial +
##   artsobs_biasField
## <environment: 0x5558d16be740>
```

Finally, we actually fit the model using `fitISDM`.

```
fish_model <- fitISDM(fish_model_setup,
  options = list(
    control.inla = list(int.strategy = 'eb', cmin = 0),
    safe = TRUE,
    inla.mode = 'experimental'))
```

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

```
summary(fish_model)
```

```
## Summary of 'bruSDM' object:
##
## inlabru version: 2.9.0
## INLA version: 23.04.24
##
## 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_log_area -0.04524608 0.1035409 -0.24818259 -0.04524608
## Perca_fluviatilis_eurolst_bio10 0.28464161 0.1567597 -0.02260185 0.28464161
## Perca_fluviatilis_SCI -0.22540404 0.1681564 -0.55498446 -0.22540404
## Perca_fluviatilis_intercept -3.15345933 5.3067567 -13.55451132 -3.15345933
##
##              0.975quant      mode kld
## Perca_fluviatilis_log_area 0.1576904 -0.04524608 0
## Perca_fluviatilis_eurolst_bio10 0.5918851 0.28464161 0
## Perca_fluviatilis_SCI 0.1041764 -0.22540404 0
## Perca_fluviatilis_intercept 7.2475927 -3.15345933 0
##
## Summary for Salmo_trutta:
##
##              mean      sd  0.025quant  0.5quant
## Salmo_trutta_log_area 0.10578962 0.06349870 -0.01866555 0.10578962
## Salmo_trutta_eurolst_bio10 -0.05242897 0.07977155 -0.20877835 -0.05242897
## Salmo_trutta_SCI -0.09169998 0.06929637 -0.22751837 -0.09169998
## Salmo_trutta_intercept 0.57235066 0.68072409 -0.76184405 0.57235066
##
##              0.975quant      mode kld
## Salmo_trutta_log_area 0.23024478 0.10578962 0
## Salmo_trutta_eurolst_bio10 0.10392040 -0.05242897 0
## Salmo_trutta_SCI 0.04411841 -0.09169998 0
## Salmo_trutta_intercept 1.90654537 0.57235066 0
##
## Summary for Salvelinus_alpinus:
##
##              mean      sd  0.025quant
## Salvelinus_alpinus_log_area 0.381397264 0.07022379 0.2437612
## Salvelinus_alpinus_eurolst_bio10 -0.185761269 0.10308470 -0.3878036
## Salvelinus_alpinus_SCI 0.003861832 0.09880188 -0.1897863
```



```

## Salvelinus_alpinus_intercept      -0.927193806 0.30243462 -1.5199548
##                                0.5quant  0.975quant      mode kld
## Salvelinus_alpinus_log_area       0.381397264 0.51903337  0.381397264  0
## Salvelinus_alpinus_eurolst_bio10 -0.185761269 0.01628103 -0.185761269  0
## Salvelinus_alpinus_SCI            0.003861832 0.19750996  0.003861832  0
## Salvelinus_alpinus_intercept      -0.927193806 -0.33443284 -0.927193806  0
##
## Summary for Esox_lucius:
##                                mean      sd 0.025quant  0.5quant
## Esox_lucius_log_area              0.1585652 0.1211022 -0.07879078  0.1585652
## Esox_lucius_eurolst_bio10        -0.2701033 0.1728488 -0.60888066 -0.2701033
## Esox_lucius_SCI                  -0.1465265 0.1559041 -0.45209296 -0.1465265
## Esox_lucius_intercept            -3.2682549 2.3870826 -7.94685083 -3.2682549
##                                0.975quant      mode kld
## Esox_lucius_log_area              0.39592114 0.1585652  0
## Esox_lucius_eurolst_bio10         0.06867412 -0.2701033  0
## Esox_lucius_SCI                   0.15903989 -0.1465265  0
## Esox_lucius_intercept             1.41034097 -3.2682549  0

## Time used:
##   Pre = 5.4, Running = 9808, Post = 0.35, Total = 9813
## Random effects:
##   Name      Model
##   Perca_fluviatilis_survey_spatial SPDE2 model
##   Salmo_trutta_survey_spatial SPDE2 model
##   Salvelinus_alpinus_survey_spatial SPDE2 model
##   Esox_lucius_survey_spatial SPDE2 model
##   artsobs_biasField SPDE2 model
##   Salvelinus_alpinus_artsobs_spatial Copy
##   Esox_lucius_artsobs_spatial Copy
##   Salmo_trutta_artsobs_spatial Copy
##   Perca_fluviatilis_artsobs_spatial Copy
##
## Model hyperparameters:
##                                mean      sd 0.025quant 0.5quant
## Theta1 for Perca_fluviatilis_survey_spatial -2.403 0.137    -2.880  -2.490
## Theta2 for Perca_fluviatilis_survey_spatial -1.208 0.141    -1.578  -1.230
## Theta1 for Salmo_trutta_survey_spatial      -0.445 0.158    -0.889  -0.489
## Theta2 for Salmo_trutta_survey_spatial      -0.885 0.122    -1.016  -0.857
## Theta1 for Salvelinus_alpinus_survey_spatial -2.368 0.069    -2.543  -2.381
## Theta2 for Salvelinus_alpinus_survey_spatial  0.680 0.083     0.582   0.716
## Theta1 for Esox_lucius_survey_spatial        -1.685 0.108    -1.767  -1.650
## Theta2 for Esox_lucius_survey_spatial        -0.810 0.054    -0.837  -0.783
## Theta1 for artsobs_biasField                  -1.120 0.200    -1.638  -1.146
## Theta2 for artsobs_biasField                  -3.041 0.104    -3.307  -3.053
##                                0.975quant      mode
## Theta1 for Perca_fluviatilis_survey_spatial -2.369 -2.318
## Theta2 for Perca_fluviatilis_survey_spatial -1.079 -1.078
## Theta1 for Salmo_trutta_survey_spatial      -0.347 -0.313
## Theta2 for Salmo_trutta_survey_spatial      -0.561 -0.982
## Theta1 for Salvelinus_alpinus_survey_spatial -2.239 -2.336
## Theta2 for Salvelinus_alpinus_survey_spatial  0.912  0.631
## Theta1 for Esox_lucius_survey_spatial        -1.386 -1.768
## Theta2 for Esox_lucius_survey_spatial        -0.643 -0.844

```

```
## Theta1 for artsobs_biasField          -0.927 -0.934
## Theta2 for artsobs_biasField          -2.917 -2.950
##
## Deviance Information Criterion (DIC) ..... -79306.75
## Deviance Information Criterion (DIC, saturated) .... NA
## Effective number of parameters ..... -82064.55
##
## Watanabe-Akaike information criterion (WAIC) ...: -Inf
## Effective number of parameters .....: 4321.20
##
## Marginal log-Likelihood: -13939.46
## 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(fish_model, "results/fish_model.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.

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

```
predict_species <- function(model, species, mask, mesh){
  sharedfield <- predict(model,
    mesh = mesh,
    mask = mask,
    format = 'sp',
    spatial = TRUE,
    fun = 'linear',
    species = species,
    n.samples = 1000)
  file_name <- paste0("results/sharedfield_", species, ".rds")
  saveRDS(sharedfield, file_name)
  return(sharedfield)
}
```

```
prediction_list <- list()
for(fish in fishes) {
  prediction_list[[fish]] <- predict_species(
    model = fish_model,
    mesh = mesh,
    mask = norway.poly,
    species = fish
  )
}
```

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.

```

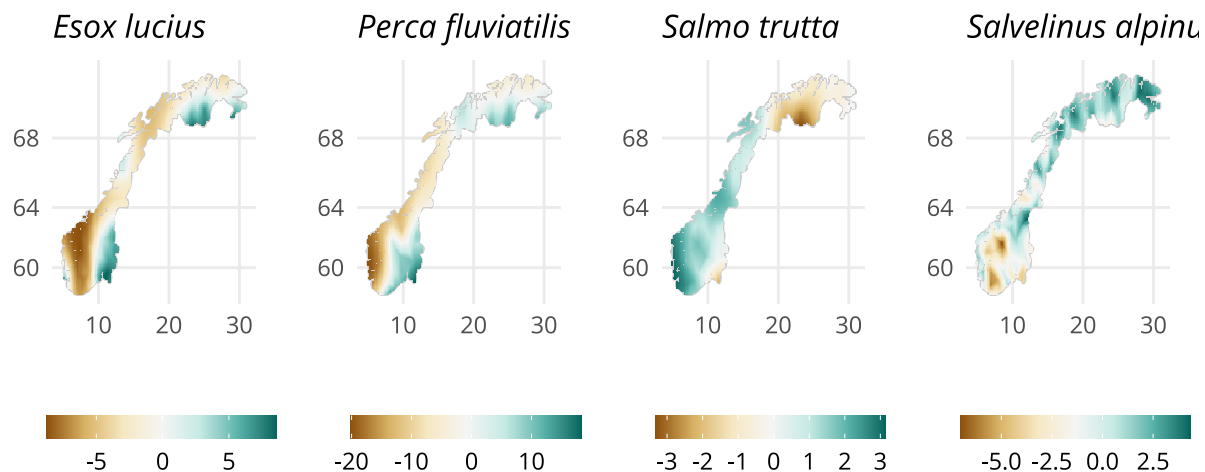
plot_preferences <- list(scale_fill_distiller(palette = "BrBG", direction = 1),
  coord_map(),
  xlab(""), ylab(""),
  theme_minimal(),
  theme(text = element_text(family = f1),
    title = element_text(family = f1, size = 10, face = "italic"),
    legend.key.height = unit(0.3, "cm"),
    legend.title = element_blank(),
    legend.position = "bottom")
)

plot_species <- function(predictions, species_to_plot, plot_preferences){
  p <- ggplot() +
    geom_polygon(data = norway, aes(long, lat, group = group),
      color="grey80", fill = "grey95") +
    gg(predictions$speciesPredictions[[species_to_plot]]) +
    labs(title = sub("_", " ", species_to_plot)) +
    plot_preferences
  return(p)
}

plot_list <- list()
for(fish in fishes) {
  plot_list[[fish]] <- plot_species(
    predictions = readRDS(paste0("results/sharedfield_", fish, ".rds")),
    species = fish,
    plot_preferences = plot_preferences)
}

patchwork::wrap_plots(plot_list, nrow = 1)

```



```

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

```

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.

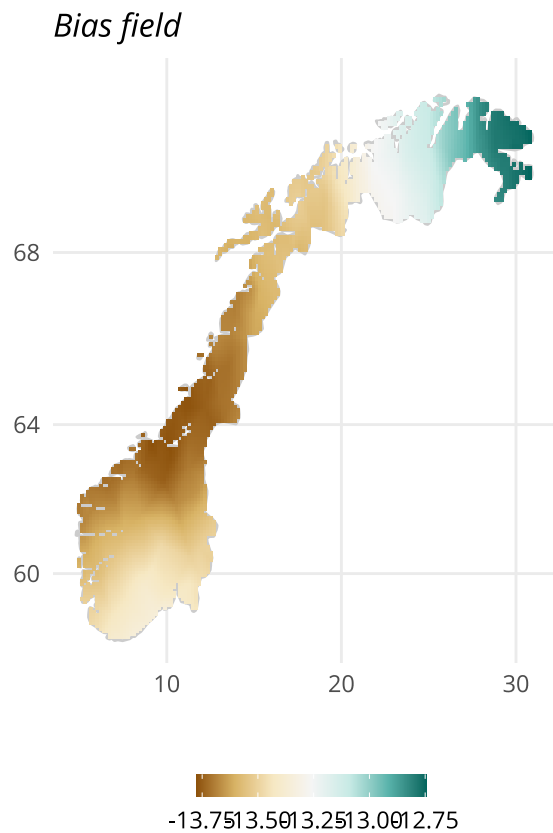
```

fish_biasfield <- predict(fish_model,
                        mesh = mesh,
                        mask = norway.poly,
                        format = 'sp',
                        biasfield = TRUE,
                        fun = 'linear',
                        n.samples = 1000)

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

ggplot() +
  geom_polygon(data = norway, aes(long, lat, group = group),
              color="grey80", fill = "grey95") +
  gg(fish_biasfield$biasFields$artsobs) +
  labs(title = "Bias field") +
  plot_preferences

```



```

ggsave("figures/fishplot_biasfield.pdf", width = 3, height = 3)

```

Session info and runtime

This document took 2.89 hours to compile.

Current session info

```

- Session info -----
setting  value
version  R version 4.3.1 (2023-06-16)
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     2023-10-24
pandoc   2.5 @ /usr/bin/ (via rmarkdown)

- Packages -----
package      * version      date (UTC) lib source
base64enc    0.1-3        2015-07-28 [1] CRAN (R 4.3.0)
blockCV      3.1-3        2023-06-04 [1] CRAN (R 4.3.0)
class        7.3-17       2020-04-26 [4] CRAN (R 4.0.0)
classInt     0.4-10       2023-09-05 [1] CRAN (R 4.3.1)
cli          3.6.1        2023-03-23 [1] CRAN (R 4.3.0)
clipr        0.8.0        2022-02-22 [1] CRAN (R 4.3.0)
codetools    0.2-18       2020-11-04 [4] CRAN (R 4.0.3)
colorspace   2.1-0        2023-01-23 [1] CRAN (R 4.3.0)
curl         5.0.1        2023-06-07 [1] CRAN (R 4.3.0)
DBI          1.1.3        2022-06-18 [1] CRAN (R 4.3.0)
desc         1.4.2        2022-09-08 [1] CRAN (R 4.3.0)
details      0.3.0        2022-03-27 [1] CRAN (R 4.3.1)
digest       0.6.31       2022-12-11 [1] CRAN (R 4.3.0)
dplyr        * 1.1.3       2023-09-03 [1] CRAN (R 4.3.1)
e1071        1.7-13       2023-02-01 [1] CRAN (R 4.3.0)
evaluate     0.21         2023-05-05 [1] CRAN (R 4.3.0)
fanshi       1.0.5        2023-10-08 [1] CRAN (R 4.3.1)
farver       2.1.1        2022-07-06 [1] CRAN (R 4.3.0)
fastmap      1.1.1        2023-02-24 [1] CRAN (R 4.3.0)
fmesher      * 0.1.2       2023-08-25 [1] CRAN (R 4.3.1)
foreach      * 1.5.2       2022-02-02 [1] CRAN (R 4.3.0)
foreign      0.8-81       2020-12-22 [4] CRAN (R 4.0.3)
generics     0.1.3        2022-07-05 [1] CRAN (R 4.3.0)
ggplot2      * 3.4.4       2023-10-12 [1] CRAN (R 4.3.1)
glue         1.6.2        2022-02-24 [1] CRAN (R 4.3.0)
gtable       0.3.4        2023-08-21 [1] CRAN (R 4.3.1)
highr        0.10         2022-12-22 [1] CRAN (R 4.3.0)
htmltools    0.5.5        2023-03-23 [1] CRAN (R 4.3.0)
httr         1.4.6        2023-05-08 [1] CRAN (R 4.3.0)
INLA         * 23.04.24    2023-04-24 [1] local
inlabru      * 2.9.0       2023-08-28 [1] CRAN (R 4.3.1)
iterators    1.0.14       2022-02-05 [1] CRAN (R 4.3.0)
jsonlite     1.8.5        2023-06-05 [1] CRAN (R 4.3.0)
KernSmooth   2.23-18      2020-10-29 [4] CRAN (R 4.0.3)
knitr        1.43         2023-05-25 [1] CRAN (R 4.3.0)
labeling     0.4.3        2023-08-29 [1] CRAN (R 4.3.1)
lattice      0.20-41      2020-04-02 [4] CRAN (R 4.0.0)
lifecycle    1.0.3        2022-10-07 [1] CRAN (R 4.3.0)

```

magrittr	2.0.3	2022-03-30	[1]	CRAN	(R 4.3.0)
mapproj	* 1.2.11	2023-01-12	[1]	CRAN	(R 4.3.1)
maps	* 3.4.1	2022-10-30	[1]	CRAN	(R 4.3.1)
maptools	1.1-8	2023-07-18	[1]	CRAN	(R 4.3.1)
Matrix	* 1.6-1	2023-08-14	[1]	CRAN	(R 4.3.1)
MatrixModels	0.5-2	2023-07-10	[1]	CRAN	(R 4.3.1)
mnormt	2.1.1	2022-09-26	[1]	CRAN	(R 4.3.0)
munsell	0.5.0	2018-06-12	[1]	CRAN	(R 4.3.0)
numDeriv	2016.8-1.1	2019-06-06	[1]	CRAN	(R 4.3.0)
patchwork	* 1.1.2	2022-08-19	[1]	CRAN	(R 4.3.0)
pillar	1.9.0	2023-03-22	[1]	CRAN	(R 4.3.0)
pkgconfig	2.0.3	2019-09-22	[1]	CRAN	(R 4.3.0)
plyr	1.8.9	2023-10-02	[1]	CRAN	(R 4.3.1)
png	0.1-8	2022-11-29	[1]	CRAN	(R 4.3.1)
PointedSDMs	* 1.3	2023-10-12	[1]	Github (PhilipMostert/PointedSDMs@699ca57)	
proxy	0.4-27	2022-06-09	[1]	CRAN	(R 4.3.0)
R.devices	2.17.1	2022-06-21	[1]	CRAN	(R 4.3.0)
R.methodsS3	1.8.2	2022-06-13	[1]	CRAN	(R 4.3.0)
R.oo	1.25.0	2022-06-12	[1]	CRAN	(R 4.3.0)
R.utils	2.12.2	2022-11-11	[1]	CRAN	(R 4.3.0)
R6	* 2.5.1	2021-08-19	[1]	CRAN	(R 4.3.0)
ragg	1.2.5	2023-01-12	[1]	CRAN	(R 4.3.0)
raster	* 3.6-23	2023-07-04	[1]	CRAN	(R 4.3.1)
RColorBrewer	1.1-3	2022-04-03	[1]	CRAN	(R 4.3.0)
Rcpp	1.0.11	2023-07-06	[1]	CRAN	(R 4.3.1)
rlang	1.1.1	2023-04-28	[1]	CRAN	(R 4.3.0)
rmarkdown	2.22	2023-06-01	[1]	CRAN	(R 4.3.0)
rprojroot	2.0.3	2022-04-02	[1]	CRAN	(R 4.3.0)
scales	1.2.1	2022-08-20	[1]	CRAN	(R 4.3.0)
sessioninfo	1.2.2	2021-12-06	[1]	CRAN	(R 4.3.0)
sf	* 1.0-14	2023-07-11	[1]	CRAN	(R 4.3.1)
showtext	* 0.9-6	2023-05-03	[1]	CRAN	(R 4.3.1)
showtextdb	* 3.0	2020-06-04	[1]	CRAN	(R 4.3.1)
sn	2.1.1	2023-04-04	[1]	CRAN	(R 4.3.0)
sp	* 2.1-0	2023-10-02	[1]	CRAN	(R 4.3.1)
sysfonts	* 0.8.8	2022-03-13	[1]	CRAN	(R 4.3.1)
systemfonts	1.0.4	2022-02-11	[1]	CRAN	(R 4.3.0)
terra	1.7-46	2023-09-06	[1]	CRAN	(R 4.3.1)
textshaping	0.3.6	2021-10-13	[1]	CRAN	(R 4.3.0)
tibble	3.2.1	2023-03-20	[1]	CRAN	(R 4.3.0)
tidyselect	1.2.0	2022-10-10	[1]	CRAN	(R 4.3.0)
units	0.8-4	2023-09-13	[1]	CRAN	(R 4.3.1)
utf8	1.2.3	2023-01-31	[1]	CRAN	(R 4.3.0)
vctrs	0.6.3	2023-06-14	[1]	CRAN	(R 4.3.1)
withr	2.5.1	2023-09-26	[1]	CRAN	(R 4.3.1)
xfun	0.39	2023-04-20	[1]	CRAN	(R 4.3.0)
xml2	1.3.4	2023-04-27	[1]	CRAN	(R 4.3.0)
yaml	2.3.7	2023-01-23	[1]	CRAN	(R 4.3.0)

[1] /home/ahomev/e/emmass/R/x86_64-pc-linux-gnu-library/4.3
 [2] /usr/local/lib/R/site-library
 [3] /usr/lib/R/site-library
 [4] /usr/lib/R/library

