Model comparison

Louis Schroll

2024-04-18

```
# load package
library(tidyverse, warn.conflicts = FALSE)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(sf)
```

```
## Linking to GEOS 3.9.3, GDAL 3.5.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(nimble, warn.conflicts = FALSE)
```

```
## nimble version 1.0.1 is loaded.
## For more information on NIMBLE and a User Manual,
## please visit https://R-nimble.org.
##
## Note for advanced users who have written their own MCMC samplers:
##   As of version 0.13.0, NIMBLE's protocol for handling posterior
##   predictive nodes has changed in a way that could affect user-defined
##   samplers in some situations. Please see Section 15.5.1 of the User Manual.
```

```
library(patchwork)
# library(extrafont)

local_path <- "~/stage_M2/"
# load functions
path_to_Rfunc <- paste0(local_path, "2.code/pt2_telemetry_and_count/R_func")
sapply(paste0(path_to_Rfunc, "/", list.files(path_to_Rfunc)), source)
```

```
##         ~/stage_M2/2.code/pt2_telemetry_and_count/R_func/format_rsf_data_for_nimble.R
## value   ?
## visible FALSE
##         ~/stage_M2/2.code/pt2_telemetry_and_count/R_func/get_new_id.R
## value   ?
## visible FALSE
##         ~/stage_M2/2.code/pt2_telemetry_and_count/R_func/make_prediction.R
## value   ?
## visible FALSE
##         ~/stage_M2/2.code/pt2_telemetry_and_count/R_func/plot_prediction.R
## value   ?
## visible FALSE
##         ~/stage_M2/2.code/pt2_telemetry_and_count/R_func/prepare_data_Nmix.R
## value   ?
## visible FALSE
##         ~/stage_M2/2.code/pt2_telemetry_and_count/R_func/run_integrated_Nmix_RSF.R
## value   ?
## visible FALSE
##         ~/stage_M2/2.code/pt2_telemetry_and_count/R_func/run_Nmixture.R
## value   ?
## visible FALSE
##         ~/stage_M2/2.code/pt2_telemetry_and_count/R_func/run_RSF.R
## value   ?
## visible FALSE
```

```r
# load data
load(paste0(local_path, "1.data/all_seabirds_counts.rdata"))
load(paste0(local_path, "1.data/grid_cells.rdata"))
load(paste0(local_path, "1.data/RSF_data_scopoli_shearwaters.rdata"))
```

```r
best_cov <- list(
  goeland_leucophee_HR = c(
    "log_dist_to_shore",
    "log_bathymetry",
    "mean_winter_SST",
    "mean_spring_SST",
    "mean_summer_SST",
    "mean_autumn_SST"
  ),
  goeland_leucophee_R = c(
    "log_dist_to_shore",
    "mean_winter_SST",
    "mean_summer_SST",
    "mean_autumn_SST"
  ),

  petit_puffin_HR = c("log_dist_to_shore","log_bathymetry",'mean_CHL',"sd_SAL","mean_SSH"),
  petit_puffin_R = c("mean_CHL", "mean_SSH"),

  puffin_de_scopoli_R = c("log_mean_CHL", "sd_SAL", "mean_SSH", "sd_SSH"),

  sterne_caugek_HR = c(
    "mean_CHL","mean_SSH","mean_winter_SST","mean_spring_SST",
    "mean_summer_SST"
  ),
  sterne_caugek_R = c(
    "log_bathymetry",
    "mean_CHL",
    "mean_SSH",
    "sd_SSH",
    "log_sd_VEL"
  )
)
```

Scopoli shearwater

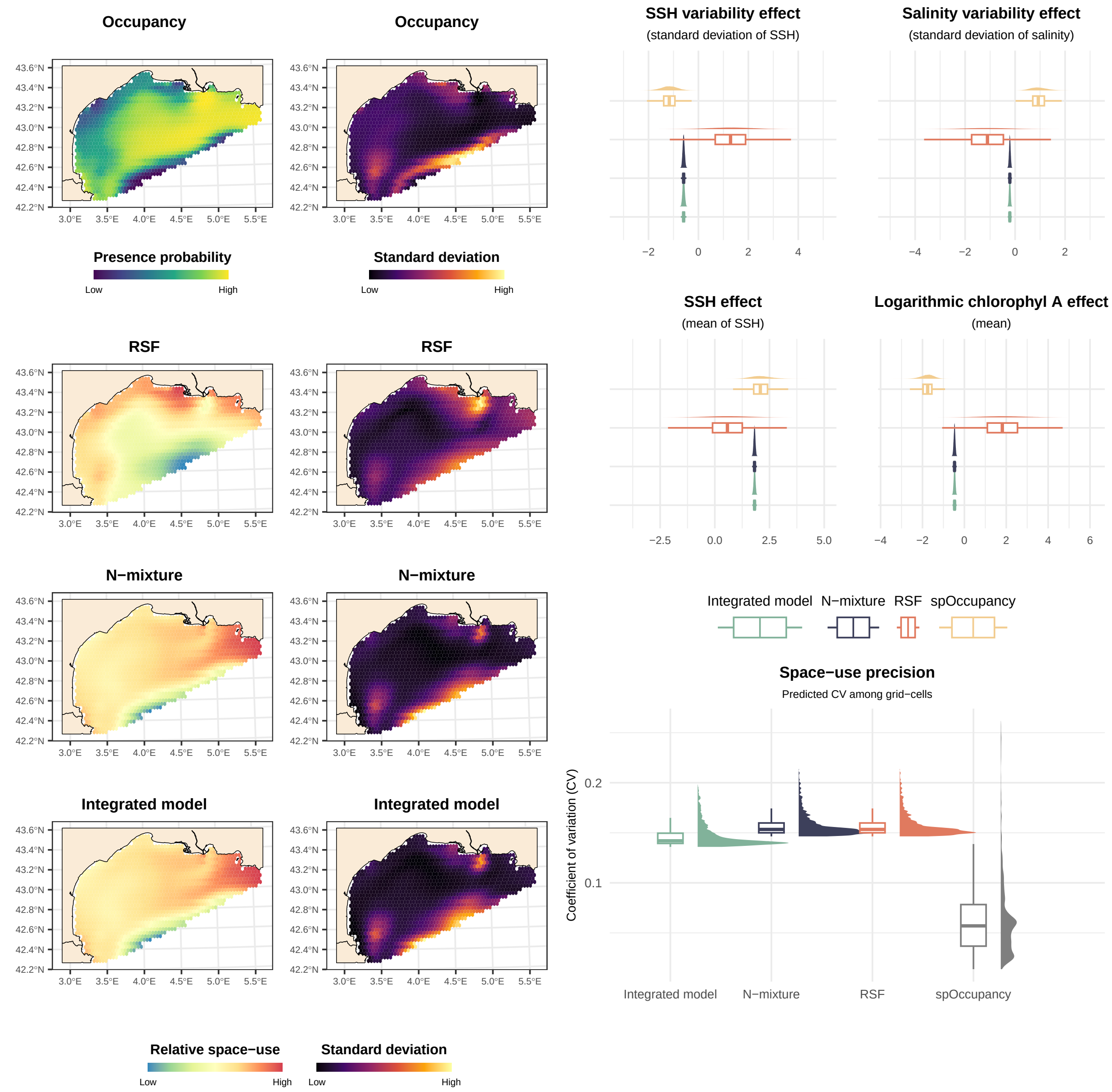
```
load(paste0(local_path, "3.results/mcmc_outputs/mcmc_output_puffin_de_scopoli_R.rdata"))
load(paste0(local_path, "3.results/spOccupancy_outputs/spOccupancy_puffin_de_scopoli_R.RData"))
load(paste0(local_path, "3.results/grid_spOccupancy_results.RData"))

selected_cov <- best_cov[["puffin_de_scopoli_R"]]

samples_spOcc <- model_result$beta.samples

# Make predictions
grid_nmix <- make_prediction(samplesNmixture, grid, selected_cov)
grid_RSF <- make_prediction(samplesRSF, grid, selected_cov, include_intercept = F, rsf_intercept = "beta
grid_int <- make_prediction(samplesint,
                            grid,
                            selected_cov,
                            rsf_intercept = "beta_pop[1]")
```

```
# Plot all the maps
# 0/3 spOccupancy
grid_scopoli <- spOccupancy_res_grid %>%
  select(grid_c, mean_psi_puffin_de_scopoli_R, sd_psi_puffin_de_scopoli_R)
names(grid_scopoli) <- c("mean_psi", "sd_psi", "grid_c")
plots_spOccupancy <- plot_occupancy(grid_scopoli, plot_title = "Occupancy")
# 1/3 - N-mixture
plots_nmix <- plot_prediction(grid_nmix, plot_title = "N-mixture")
# 2/3 - RSF
plots_rsf <- plot_prediction(grid_RSF, plot_title = "RSF")
plots_rsf_mean <- plots_rsf$mean_psi_plot
# 3/3 - RSF & N-mixture
plots_int <- plot_prediction(grid_int,
                             add_colonies = F,
                             plot_title = "Integrated model")
```

```
predictive_maps <- (plots_spOccupancy$mean_psi_plot + plots_spOccupancy$sd_psi_plot) /
  ((plots_rsf_mean + plots_rsf$sd_psi_plot) /
  (plots_nmix$mean_psi_plot + plots_nmix$sd_psi_plot) /
  (plots_int$mean_psi_plot + plots_int$sd_psi_plot) +
  plot_layout(guides = "collect") +
  plot_annotation(
    title = '',
    #caption = 'Source: Migralion project & PELMED 2017-2021',
    theme = theme(
      plot.title = element_text(size = 20,
                                face = "bold",
                                hjust = 0.5),
      legend.position = "bottom",
    )
  )) +
  plot_layout(heights = c(1, 3.8))

coeff_df <- gather_coeff_values(sampleNmixture, samplesRSF, samplesint, samples_spOcc, selected_cov)

sal_plot <- plot_coeff_by_model(coeff_df,
                    covariate_name = "sd_sal",
                    plot_title = "Salinity variability effect",
                    plot_subtitle = "(standard deviation of salinity)")

sd_ssh_plot <- plot_coeff_by_model(coeff_df,
                    covariate_name = "sd_ssh",
                    plot_title = "SSH variability effect",
                    plot_subtitle = "(standard deviation of SSH)")

mean_ssh_plot <- plot_coeff_by_model(coeff_df,
                                covariate_name = "mean_ssh",
                                plot_title = "SSH effect",
                                plot_subtitle = "(mean of SSH)")

chl_plot <- plot_coeff_by_model(coeff_df,
                                covariate_name = "log_mean_chl",
                                plot_title = "Logarithmic chlorophyl A effect",
                                plot_subtitle = "(mean)")

coefficient_plot <- (sd_ssh_plot + sal_plot + mean_ssh_plot + chl_plot) +
  plot_layout(guides = "collect", tag_level = "new") +
  plot_annotation(
    theme = theme(
      legend.position = "bottom",
      legend.text = element_text(
        #family  = "Arial",
        face = "bold",
        size = 9
      )))

precision_plot <- plot_space_use_precision(grid_nmix = grid_nmix,
                         grid_RSF = grid_nmix,
                         grid_int = grid_int,
                         grid_spOcc = grid_scopoli)

plot_left <- (coefficient_plot / precision_plot) +
  plot_layout(heights = c(7, 4))

predictive_maps | plot_left +
  plot_layout(
    guide = "keep"
  ) +
  plot_annotation(
    title = "Scopoli shearwater space-use",
    #caption = 'Source: Migralion project & PELMED 2017-2021',
    theme = theme(
      plot.title = element_text(size = 20, face = "bold", hjust = 0.5),
      legend.position = 'top'
    )
  )
```

```
## Warning: Using the ‘size‘ aesthetic with geom_polygon was deprecated in ggplot2 3.4.0.
## i Please use the ‘linewidth‘ aesthetic instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## Warning: Using the ‘size‘ aesthetic with geom_segment was deprecated in ggplot2 3.4.0.
## i Please use the ‘linewidth‘ aesthetic instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Sandwich tern