2.0_trends-analysis_explore

Devi Veytia

2025-07-08

```r
# general data handing
library(dplyr)
library(dbplyr)
library(RSQLite)
library(R.utils)
library(ggplot2)
library(ggalluvial)
library(tidyr)
library(stringr)
library(viridis)
library(countrycode)
library(broom)
library(conflicted)
library(tidyverse)
library(cowplot)
library(ggpubr)
library(patchwork)
library(igraph)
library(ggraph)
library(tidygraph)
library(ggplot2)
require(VLTimeCausality) # remotes::install_github("DarkEyes/VLTimeSeriesCausality")
library(QPress) # remotes::install_github("SWotherspoon/QPress")
library(reshape2)


conflict_prefer("select", "dplyr")
conflicts_prefer(dplyr::filter)




## AESTHETICS
factor_aes <- readxl::read_excel(here::here("R/mitigation_factor_aesthetics2.xlsx"))
typeAES <- factor_aes[which(factor_aes$variable == "oro_type"),]
typeAES <- typeAES[order(typeAES$order),]

componentAES <- factor_aes[which(factor_aes$variable == "component"),]
componentAES <- componentAES[order(componentAES$order),]

## Set seed
addTaskCallback(function(...) {set.seed(123);TRUE})
```

Introduction

Entry point: The distribution of and extent of scientific evidence, and its temporal variation, varies according to the type of ocean-related option (ORO) (Veytia et al, In review).

Aim: This work is a hypothesis generating exercise, which asks: What are the possible drivers of this variability? What are the socio-political antecedents of breakpoints/inflections? Over time, is variability in scientific evidence linked to policy, public interest and action? What does this tell us about the research/policy/action cycle?

This aim will be addressed in two parts:

Part 1: Narrative synthesis. Here we qualitatively explore break points in publication #s, as well as action/implementation, and discuss possible drivers of these changes.

Part 2: Quantitative synthesis.

2.1 - Match/mis-match. Do the most evidenced OROs appear most frequently in policy and public interest metrics? 2.2 - Network analysis. Do we find evidence for causal links between the different nodes of the research/policy/action cycle? If so, what does this tell us about the pathways towards implementing OROs?

Summary of main conclusions:

Over the entire time series, our quantitative analysis of causality shows publications and public interest to be direct drivers of action, but that policy and legislation play an influential intermediary role.

That said, public interest does not match well with publications, which may cause conflicting pressures that act on action. Legislation and policy show better alignment.

However, it is clear from our narrative synthesis that the right legislation can have enormous impact producing step changes in action. Therefore not all policy/legislation documents act equally on the system in terms of impact, and the potential for one document to have a large impact is arguably higher than all the other dimensions we investigated.

Load data

Data structure: list where each level is - data frame for publications: - data frame for n policy documents - data frame for deployment

each data frame has the following id variables: oro_type, component (publication, policy, deployment, etc), variable_name, each data frame has the following response variable: y

```r
load(here::here("data", "derived-data", "mitigationORO_pubs.RData")) #pubs
load(here::here("data", "derived-data", "n_nonBindPolicy_docs.RData")) # legDat
load(here::here("data", "derived-data", "n_legislation_docs.RData")) # polDat
load(here::here("data/derived-data/mitigationDeployDat.RData")) # allDeployDat
load(here::here("data/derived-data/mitigationPostsDat.RData")) # postsDat

year_lim <- c(2000, 2024) # years to analyse


# bind all data together
allComponentDat_model <- pubs %>%
  bind_rows(polDat) %>%
  bind_rows(legDat) %>%
  bind_rows(allDeployDat) %>%
  bind_rows(postsDat) %>%
  mutate(
    component = replace(component, component == "non-binding policy", "policy"),
    # y = scale(y, center=TRUE, scale=TRUE)
  ) %>%
  filter(year_lim[1] <= year & year <= year_lim[2],
         oro_type %in% typeAES$level)
```

Entry point: The distribution of and extent of scientific evidence, and its temporal variation, varies according to the type of ocean-related option (ORO)

Key message Mitigation ORO publication effort is weighted towards marine renewable energy (ocean, located). Over time, we observe varying rates of increase, with inflection points – notably in MRE-Ocean occurring in 2001. This inspired this analysis: What are the drivers of this inflection point? To go further, we complement our data set of # publications with # policy documents, # legislative documents, # social media posts, and # social media posts (positive sentiment)

Skip the code and just run the chunk to visualize at the end

```
p1_db <- RSQLite::dbConnect(RSQLite::SQLite(),
                      here::here("data","sqlite-databases","product1.sqlite"),
                                 create=FALSE)
error_bars <- tbl(p1_db, "pred_oro_type_mit_long") %>%
  group_by(level)%>%
  summarise(
    n_lower = n_distinct(analysis_id[0.5 <= (mean_prediction - std_prediction)]),
    n_mean = n_distinct(analysis_id[0.5 <= (mean_prediction)]),
    n_upper = n_distinct(analysis_id[0.5 <= (mean_prediction + std_prediction)])
  )%>%
  collect()%>%
  mutate(
    oro_type = gsub("[.]","-",level)
  )


predOroType <- tbl(p1_db, "pred_oro_type_mit_long") %>%
  # filter(0.5 <= mean_prediction) %>%
  rename(oro_type = level) %>%
  select(analysis_id, oro_type, mean_prediction, std_prediction) %>%
  mutate(
    lower_prediction = mean_prediction-std_prediction,
    upper_prediction = mean_prediction+std_prediction
  )
  # left_join(uniquerefs, by = "analysis_id") %>%
  # collect()

pred_effective <- tbl(p1_db, "pred_outcome_effectiveness")%>%
  filter(0.5 <= mean_prediction) %>%
  rename(effective_mean_pred = mean_prediction) %>%
  select(analysis_id, effective_mean_pred)
  # collect()
```

```r
pred_quant <- tbl(p1_db, "pred_outcome_quantitative")%>%
  filter(0.5 <= mean_prediction) %>%
  rename(quantitative_mean_pred = mean_prediction) %>%
  select(analysis_id, quantitative_mean_pred)
  # collect()

pred_primary <- tbl(p1_db, "pred_primary_research") %>%
  filter(0.5 <= mean_prediction) %>%
  rename(primary_mean_pred = mean_prediction) %>%
  select(analysis_id, primary_mean_pred)
  # collect()

outcome_df <- predOroType %>%
  left_join(pred_primary) %>%
  left_join(pred_quant) %>%
  left_join(pred_effective) %>%
  collect()%>%
  mutate(across(contains("_mean_pred"), function(x) ifelse(x < 0.5 | is.na(x), FALSE, TRUE)),
         across(contains("_prediction"), function(x) ifelse(0.5<= x | is.na(x), TRUE, FALSE)),
         oro_type = factor(
           gsub("[.]","-", oro_type),
           levels = typeAES$level,
           labels = typeAES$label
         )
         )


outcome_df_bars <- outcome_df%>%
  group_by(oro_type) %>%
  summarise(
    n_mean = n_distinct(analysis_id[mean_prediction==TRUE]),
    n_lower = n_distinct(analysis_id[lower_prediction==TRUE]),
    n_upper = n_distinct(analysis_id[upper_prediction==TRUE]),
    n_secondary = n_distinct(analysis_id[mean_prediction==TRUE &
                                          primary_mean_pred != TRUE &
                                          quantitative_mean_pred != TRUE &
                                          effective_mean_pred != TRUE]),
    n_primary = n_distinct(analysis_id[mean_prediction==TRUE &
                                         primary_mean_pred== TRUE &
                                          quantitative_mean_pred != TRUE &
                                          effective_mean_pred != TRUE]),
    n_quantitative = n_distinct(analysis_id[mean_prediction==TRUE &
                                              primary_mean_pred == TRUE&
                                          quantitative_mean_pred == TRUE &
                                          effective_mean_pred != TRUE]),
    n_effective = n_distinct(analysis_id[mean_prediction==TRUE &
                                           primary_mean_pred == TRUE&
                                          quantitative_mean_pred == TRUE&
                                          effective_mean_pred == TRUE])
  )

outcome_df %>%
  group_by(oro_type)%>%
  summarise(
    ntotal = n_distinct(analysis_id)
  )

outcome_error <- outcome_df_bars %>%
  select(oro_type, n_mean, n_lower, n_upper)

outcome_df_bars_long <- outcome_df_bars %>%
  select(-n_mean, -n_lower, -n_upper)%>%
  pivot_longer(cols = starts_with(c("n_")), names_to = "prediction_type", values_to = "value") %>%
  mutate(
    prediction_type = factor(gsub(".*_","", prediction_type),
                            levels = rev(c("secondary","primary", "quantitative", "effective")),
                            labels = rev(c("Secondary","Primary - Other",
                                           "Quantitative - Other", "Quantitative Effectiveness")))
  )




# Plot
outcomes_ggp <- ggplot(outcome_df_bars_long,
                       aes(x = oro_type, y = value, fill = prediction_type)) +
  geom_col(position = "stack") +
  geom_errorbar(data = outcome_error, aes(x = oro_type, ymin = n_lower, ymax = n_upper),
              width = .2,
              inherit.aes = FALSE)+
  scale_fill_brewer(type = "qual")+
  guides(fill=guide_legend(nrow=4))+
  labs(
    x = "ORO type",
    y="N articles (mean)",
    fill = "Outcome type"
  ) +
  theme_minimal(base_size = 10) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "bottom"
    )
outcomes_ggp


# # save counts to table
# write.csv(outcome_error %>%
#             rename(`ORO type` = oro_type,
#                    `N (mean)`=n_mean,
#                    `N (lower)`=n_lower,
#                    `N (upper)`=n_upper,
#                    ),
#           file = here::here("outputs/oroTypePredictionCounts.csv"))

DBI::dbDisconnect(p1_db)
```

What was driving the inflection in the mitigation ORO branch? A particular ORO or a particular affiliation?

```r
p1_db <- RSQLite::dbConnect(RSQLite::SQLite(),
                      here::here("data","sqlite-databases","product1.sqlite"),
                                 create=FALSE)

uniquerefs <- tbl(p1_db, "uniquerefs_update2025") %>%
  select(analysis_id, year)

predOroType <- tbl(p1_db, "pred_oro_type_mit_long") %>%
  left_join(uniquerefs, by = "analysis_id") %>%
  collect()

dbDisconnect(p1_db)

oroPub <- predOroType %>%
  mutate(
    std_prediction = ifelse(is.na(std_prediction), 0, std_prediction),
    level = gsub("[.]","-", level),
    year= as.numeric(year)
    ) %>%
  filter(!is.na(year)) %>%
  mutate(
    lower_prediction = mean_prediction - std_prediction,
    upper_prediction = mean_prediction + std_prediction
  ) %>%
  group_by(level, year) %>%
  summarise(
    y_mean = n_distinct(analysis_id[0.5 <= mean_prediction]),
    y_lower = n_distinct(analysis_id[0.5 <= lower_prediction]),
    y_upper = n_distinct(analysis_id[0.5 <= upper_prediction])
  )
  # mutate(
  #   oro_type = factor(
  #     level,
  #     levels = typeAES$level,
  #     labels = typeAES$label
  #   )
  #
  # )
allPub <- predOroType %>%
  mutate(
    std_prediction = ifelse(is.na(std_prediction), 0, std_prediction),
    level = gsub("[.]","-", level),
    year= as.numeric(year)
    ) %>%
  filter(!is.na(year)) %>%
  mutate(
    lower_prediction = mean_prediction - std_prediction,
    upper_prediction = mean_prediction + std_prediction
  ) %>%
  group_by(year) %>%
  summarise(
    y_mean = n_distinct(analysis_id[0.5 <= mean_prediction]),
    y_lower = n_distinct(analysis_id[0.5 <= lower_prediction]),
    y_upper = n_distinct(analysis_id[0.5 <= upper_prediction])
  )%>%
  mutate(level = paste("All mitigation OROs"))


allMit <- allComponentDat_model %>%
  filter(component == "publications") %>%
  group_by(year) %>%
  summarise(y = sum(y, na.rm=T)) %>%
  mutate(oro_type = "All mitigation OROs") %>%
  bind_rows(allComponentDat_model[allComponentDat_model$component == "publications",])

allMit <- oroPub %>%
  bind_rows(allPub)

oroTimeseries_ggp<- ggplot(allMit%>% filter(1980 <= year, year <= 2024), aes(x=year))+
  geom_rect(xmin = 2001, xmax = 2005, ymin=-Inf, ymax=Inf, alpha = 0.1, fill="grey")+
  geom_line(aes(y=log(y_mean), col=level))+
  geom_ribbon(aes(ymin = log(y_lower), ymax = log(y_upper), fill = level), alpha = 0.5)+
  facet_wrap(vars(level), scales = "free_y")+
  scale_color_manual(
    breaks = c("All mitigation OROs", typeAES$level),
    values = c("#35a7d9", typeAES$colour),
    labels = c("All mitigation OROs", typeAES$label)
  )+
  scale_fill_manual(
    breaks = c("All mitigation OROs", typeAES$level),
    values = c("#35a7d9", typeAES$colour),
    labels = c("All mitigation OROs", typeAES$label)
  )+
  scale_y_continuous()+
  scale_x_continuous(limits = c(1980,2024))+
  # guides(col=guide_legend(nrow=2))+
  labs(x="Year", y = "log(N publications)", col = "ORO type")+
  theme_minimal()+
  theme(
    legend.position = "none",
    axis.text.x = element_text(angle=45, hjust=1)
  )

oroTimeseries_ggp
# ggsave(
#   here::here("figures/supplemental/nPublicationsTimeseriesPlots.pdf"),
#   plot = oroTimeseries_ggp,
#   width = 7, height=5
# )
```

```r
# ggarrange(plotlist = list(outcomes_ggp, oroTimeseries_ggp),
#           nrow=1, labels = letters[1:2], align = "v", widths = c(1,1.7)) %>%
#   ggexport(
#     filename = here::here("figures/supplemental/oro_outcome_and_timeseries.pdf"),
#     width = 8, height = 5
#    )
```

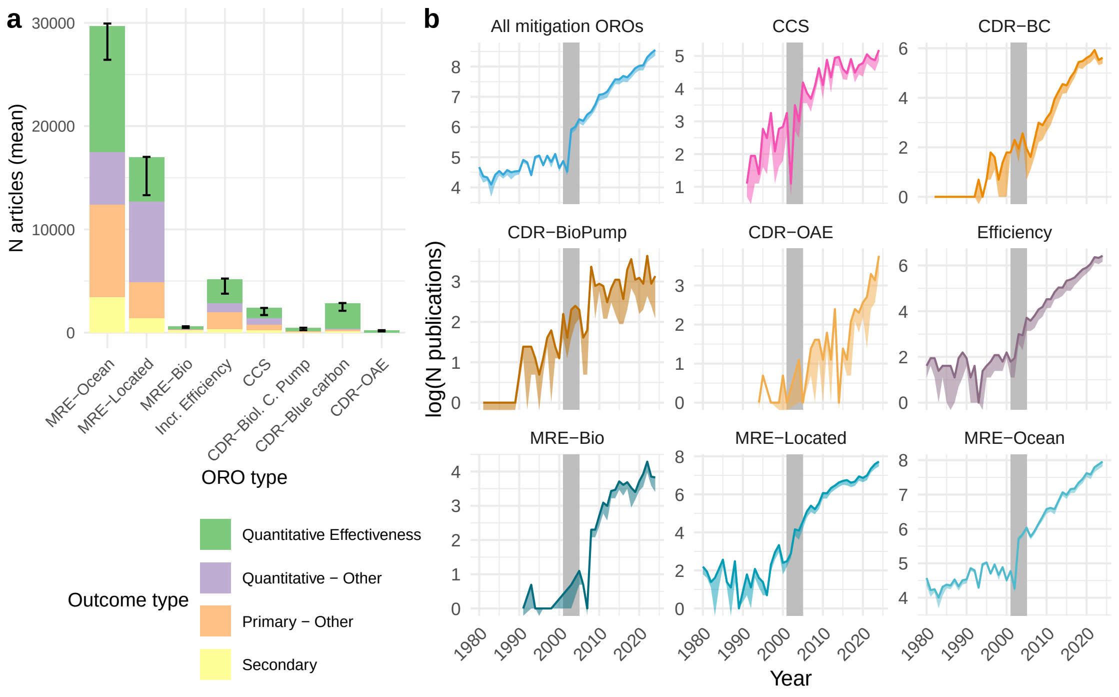

Figure 1: Distribution of N publications (predicted relevant by LLM classifier)

Complementary data

To contextualize our publication data, we have added the following metrics:

Number of policy and legislative documents - Calculated by web scraping ECOLEX & FAOLEX databases then keyword searching full document pdfs for keywords relevant for each type of ORO. Document type metadata used to determine legislation vs policy

Interest - N posts returned from keyword searches (reddit, youtube, bluesky, linked in), weighted by # likes

Support - N posts (from above) that are positive sentiment (predicted using a pre-trained sentiment classification LLM)

Action - various metrics (see below)

```r
# don't plot all components in the alluvial -- it would be too much
selectedComponents <- c("policy","legislation","public interest","public support")

alluvialComponents <- c("publications","deployment",selectedComponents)

# scale deployment data
allDeployDat_scale <- allComponentDat_model %>%
  filter(component == "deployment", !is.na(year)) %>%
  arrange(oro_type, year) %>%
  group_by(oro_type) %>%
  mutate(
    # y=cut(
    #   scales::rescale(y, to=c(0,1)),
    #   breaks = seq(0,1, by = 0.1),
    #   labels = seq(0.1,1, by = 0.1)
    #   ) %>% as.numeric()# if i do 0,1 it throws an error
    y = scales::rescale(y, to=c(0,1))
  ) %>%
  ungroup()


## plot all together

allComponentDat_scale <- allComponentDat_model %>%
  filter(component != "deployment") %>%
  bind_rows(allDeployDat_scale) %>%
  filter(component %in% alluvialComponents) %>%
  mutate(
    component = factor(component,
                       levels = componentAES$level[
                         componentAES$level %in% alluvialComponents
                       ],
                       labels = componentAES$label[
                         componentAES$level %in% alluvialComponents
                       ]),
    log_y = log(y)
  )




allComponentAlluvial <- ggplot(allComponentDat_scale,
                               aes(x=year, y=y, alluvium=oro_type, fill=oro_type, colour=oro_type))+
  # geom_alluvium(alpha = 0.8, decreasing=FALSE)+
  geom_alluvium(alpha = 0.8, curve_type = "sigmoid", decreasing=FALSE) +
  labs(y="",x="Year")+
  facet_grid(component~., scales = "free_y", switch="y")+
  scale_fill_manual(name = "ORO type",values = typeAES$colour, breaks = typeAES$level, labels = typeAES
  scale_colour_manual(name = "ORO type",values = typeAES$colour, breaks = typeAES$level, labels = typeA
  # scale_x_continuous(limits = c(1980,2022))+
  guides(color = guide_legend(nrow = 3), fill = guide_legend(nrow = 3))+
  theme_bw()+
  theme(
    legend.position = "bottom",
    strip.placement = "outside",
    legend.title = element_blank()
    )
allComponentAlluvial

## try on log scale?
allDeployDat_scale <- allComponentDat_model %>%
  filter(component == "deployment", !is.na(year)) %>%
  arrange(oro_type, year) %>%
  group_by(oro_type) %>%
  mutate(
```

```r
    y=cut(
      scales::rescale(y, to=c(0,1)),
      breaks = seq(0,1, by = 0.1),
      labels = seq(0.1,1, by = 0.1)
      ) %>% as.numeric()# if i do 0,1 it throws an error
    # y = scales::rescale(y, to=c(0,1))
  ) %>%
  ungroup()


## plot all together
componentAES <- componentAES %>%
  mutate(
    labelUnits = case_when(
      grepl("Pub|Pol|Leg", label) ~ paste(label, "(N)", sep="\n"),
      grepl("Inter|Support|Opposition", label) ~ paste(label, "(N posts\nweighted)", sep="\n"),
      TRUE ~ paste(label,"(various\nmetrics)", sep="\n")
    )
  )

allComponentDat_scale <- allComponentDat_model %>%
  filter(component != "deployment") %>%
  bind_rows(allDeployDat_scale) %>%
  filter(component %in% alluvialComponents) %>%
  mutate(
    component = factor(component,
                       levels = componentAES$level[
                         componentAES$level %in% alluvialComponents
                       ],
                       labels = componentAES$labelUnits[
                         componentAES$level %in% alluvialComponents
                       ]),
    log_y = log(y)
  )



ybreaks = pretty(allComponentDat_scale$log_y)
ybreaks = seq(
  from=min(allComponentDat_scale$log_y[!is.infinite(allComponentDat_scale$log_y)], na.rm=T),
  to=max(allComponentDat_scale$log_y[!is.infinite(allComponentDat_scale$log_y)], na.rm=T),
  by = 0.1
)
allComponentAlluvial_log <- ggplot(allComponentDat_scale %>%
                                     mutate(
                                       y= cut(log_y,
                                              breaks = ybreaks,
                                              labels = ybreaks[-1]
                                              ) %>% as.numeric()
                                     ),
                                   aes(x=year, y=y, alluvium=oro_type, fill=oro_type, colour=oro_type))
  # geom_alluvium(alpha = 0.8, decreasing=FALSE)+
  geom_alluvium(alpha = 0.8, curve_type = "sigmoid", decreasing=FALSE) +
  labs(y="log(Metric)",x="Year")+
  facet_grid(component~., scales = "free_y"
             # labeller = label_wrap_gen(width=25)
             )+ #switch="y"
  scale_fill_manual(name = "ORO type",values = typeAES$colour, breaks = typeAES$level, labels = typeAES
  scale_colour_manual(name = "ORO type",values = typeAES$colour, breaks = typeAES$level, labels = typeA
  # scale_x_continuous(limits = c(1980,2022))+
  # guides(color = guide_legend(nrow = 4), fill = guide_legend(nrow = 3))+
  guides(color = guide_legend(ncol = 1), fill = guide_legend(ncol = 1))+
  theme_bw()+
  theme(
    legend.position = "right",
    strip.placement = "outside"
    # legend.title = element_blank()
    )



# ggsave(
#   plot = allComponentAlluvial_log,
#   filename = here::here("figures/main/componentAlluvial_log.pdf"),
#   width=6, height=6, units="in"
#     )
```

```
## plot all together
componentAES <- componentAES %>%
  mutate(
    labelUnits = case_when(
      grepl("Pub|Pol|Leg", label) ~ paste(label, "(N)", sep="\n"),
      grepl("Inter|Support|Opposition", label) ~ paste(label, "(N posts\nweighted)", sep="\n"),
      TRUE ~ paste(label,"(various\nmetrics)", sep="\n")
    )
  )

allComponentDat_scale2 <- allComponentDat_model %>%
  filter(component != "deployment") %>%
  bind_rows(allDeployDat_scale) %>%
  mutate(
    component = factor(component,
                       levels = componentAES$level,
                       labels = componentAES$labelUnits),
    log_y = log(y)
  )



ybreaks = pretty(allComponentDat_scale2$log_y)
ybreaks = seq(
  from=min(allComponentDat_scale2$log_y[!is.infinite(allComponentDat_scale2$log_y)], na.rm=T),
  to=max(allComponentDat_scale2$log_y[!is.infinite(allComponentDat_scale2$log_y)], na.rm=T),
  by = 0.1
)
allComponentAlluvial_log2 <- ggplot(allComponentDat_scale2 %>%
                                    filter(!is.na(log_y), !is.na(component)) %>%
                                     mutate(
                                       y= cut(log_y,
                                              breaks = ybreaks,
                                              labels = ybreaks[-1]
                                              ) %>% as.numeric()
                                     ),
                                   aes(x=year, y=y, alluvium=oro_type, fill=oro_type, colour=oro_type))
  # geom_alluvium(alpha = 0.8, decreasing=FALSE)+
  geom_alluvium(alpha = 0.8, curve_type = "sigmoid", decreasing=FALSE) +
  labs(y="log(Metric)",x="Year")+
  facet_grid(component~., scales = "free_y"
             # labeller = label_wrap_gen(width=25)
             )+ #switch="y"
  scale_fill_manual(name = "ORO type",values = typeAES$colour, breaks = typeAES$level, labels = typeAES
  scale_colour_manual(name = "ORO type",values = typeAES$colour, breaks = typeAES$level, labels = typeA
  # scale_x_continuous(limits = c(1980,2022))+
  # guides(color = guide_legend(nrow = 4), fill = guide_legend(nrow = 3))+
  guides(color = guide_legend(ncol = 1), fill = guide_legend(ncol = 1))+
  theme_bw()+
  theme(
    legend.position = "right",
    strip.placement = "outside"
    # legend.title = element_blank()
    )


allComponentAlluvial_log2

# ggsave(
#   plot = allComponentAlluvial_log2,
#   filename = here::here("figures/supplemental/all_componentAlluvial_log.pdf"),
#   width=6, height=10, units="in"
#     )
```

A closer look at the ORO-specific 'action' metrics:

MRE - installed capacity (MW) - IRENA renewable electricity statistics

Efficiency - domestic freight transport energy efficiency (gCO2/tkm) - IEA

CCS - Storage Capacity associated with a project- oil and gas climate initiative

CDR-BC - # restoration projects - Duarte et al 2020

CDR-OAE/BioPump - # Field trials/startup companies - Ocean Visions Field trials, GESAMP climate intervention projects, OceanNETs ocean-based CDR companies

```
allDep_ggp <- ggplot(
  data = allComponentDat_model %>%
    filter(component == "deployment") %>%
    mutate(
    oro_type = factor(oro_type, levels = typeAES$level, labels = typeAES$label)
  )
)+
  geom_line(aes(year, y, col = oro_type))+
```

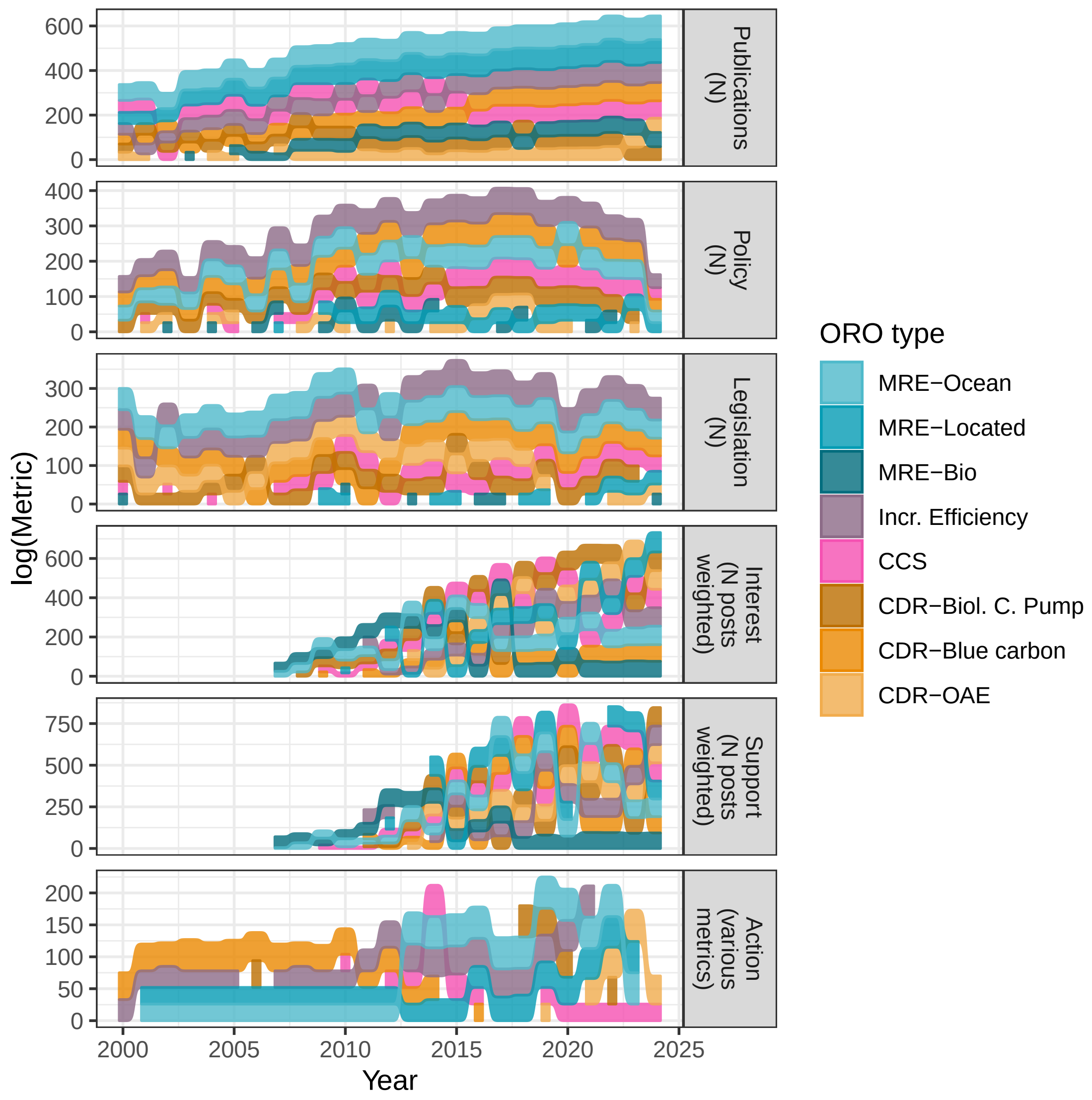

Figure 2: Data set alluvial plot

```
  geom_text(aes(label = stringr::str_wrap(variable_name, width = 25),
                fontface=3), x=-Inf, y=Inf,
            check_overlap = TRUE, vjust=1, hjust=0, size=3)+
  facet_wrap(vars(oro_type), scales="free")+
  scale_colour_manual(
    breaks = typeAES$label,
    values = typeAES$colour
  )+
  labs(
    x="Year",
    y="Action metric",
  )+
  theme_minimal()+
  theme(
    legend.position = "none"
  )

allDep_ggp

# ggsave(here::here("figures/supplemental/mitigationDeploymentIndicators.pdf"), plot = allDep_ggp, widt
```

Part 1: Narrative synthesis. Investigate drivers of break points

Key message Key changes in policy/legislation, catalysed by the right socio-political environment, can have enormous impact on publication & action.

Case 1: Increase in MRE-Ocean publications in 2001

This inflection was driven by a dramatic increase in publications by authors with Brazilian affiliations. Brazil experienced an energy crisis in 2001 due to droughts affecting hydroelectric grid – prompting diversification of renewable sources

```
## connect to database
p1_db <- RSQLite::dbConnect(RSQLite::SQLite(),
                            here::here("data","sqlite-databases","product1.sqlite"),
                                       create=FALSE)

uniquerefs <- tbl(p1_db, "uniquerefs_update2025") %>%
  select(analysis_id, year, affiliation)

idCountry <- tbl(p1_db, "pred_oro_type_long") %>%
  select(analysis_id) %>%
  distinct(analysis_id) %>%
  left_join(uniquerefs %>% select(analysis_id, year, affiliation), by = "analysis_id") %>%
  collect()



# Get a list of country names and codes
load(here::here("data/derived-data/countries_ls.RData"))
```

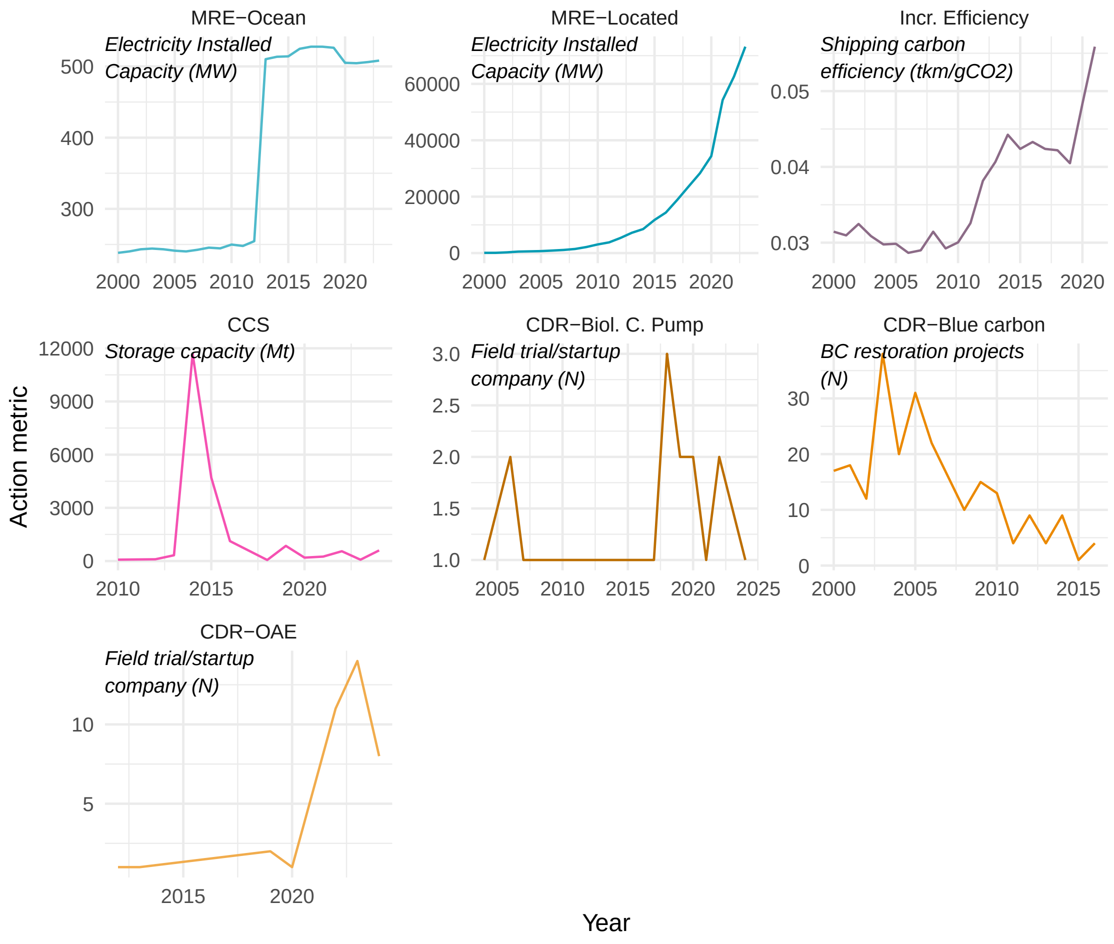

Figure 3: ORO-specific action metrics

```r
# countries_ls <- data.frame(name_en = countrycode::codelist$country.name.en) %>%
#   mutate(country_iso = countrycode::countrycode(sourcevar   = name_en,
#                                          origin      = "country.name",
#                                          destination = "iso3c"),
#          country=name_en)

source(here::here("R", "extract_all_affiliation.R"))

idCountry$countries <- extract_all_affiliation(idCountry$affiliation, countries_ls)

dbWriteTable(conn=p1_db,
             name= "oro_id_allCountryAffil_lookup",
             value=idCountry,
             append=FALSE,
             overwrite = FALSE)


dbDisconnect(p1_db)
```

```r
## Load data
p1_db <- RSQLite::dbConnect(RSQLite::SQLite(),
                      here::here("data","sqlite-databases","product1.sqlite"),
                                 create=FALSE)

# The author affiliations of each publication
idCountry <- tbl(p1_db, "oro_id_allCountryAffil_lookup") %>%
  select(analysis_id, countries, affiliation)%>%
  collect()

# publication year metadata
uniquerefs <- tbl(p1_db, "uniquerefs_update2025") %>%
  select(analysis_id, year)

# The article ids and their ORO type prediction
predOroType <- tbl(p1_db, "pred_oro_type_mit_long") %>%
  left_join(uniquerefs, by = "analysis_id") %>%
  collect()%>%
  left_join(idCountry, by="analysis_id")


dbDisconnect(p1_db)



## Calculate per oro type, the frequency of country affiliation per year
# columns, analysis_id, oro_type, year, country, count
uniqueCountries <- strsplit(idCountry$countries, " ; ") %>% unlist() %>% unique()

affTypeTop10 <- strsplit(idCountry$countries, ", ") %>% unlist() %>% table %>% sort(decreasing = TRUE)


oro_year_count_long <- data.frame()
```

```
for(country in uniqueCountries){
  dfTemp <- predOroType %>%
    filter(grepl(country, countries)) %>%
    group_by(level, year) %>%
    summarise(
      count = n_distinct(analysis_id)
    ) %>%
    mutate(
      country_unique = country
    )
  oro_year_count_long <- oro_year_count_long %>%
    bind_rows(dfTemp)
}

oro_year_count_long <- oro_year_count_long %>%
  mutate(
    country = ifelse(country_unique %in% affTypeTop10, country_unique, "Other")
  )

totals <- oro_year_count_long %>%
  group_by(level, year) %>%
  summarise(
    total = sum(count, na.rm=T)
  )
oro_year_count_long <- oro_year_count_long %>%
  left_join(totals) %>%
  mutate(
    prop = count/total,
    oro_type = factor(
      gsub("[.]","-", level),
      levels = typeAES$level,
      labels = typeAES$label
    ),
    year = as.numeric(year)
  )%>%
  filter(
    !is.na(year)
  )%>%
  filter(
    1980 <= year, year <= 2024
  )

summary(oro_year_count_long)


## Overlay a lineplot of the interannual trend in N publications for each ORO type
oroPub01 <- predOroType %>%
  mutate(
    level = gsub("[.]","-", level),
    year= as.numeric(year)
    ) %>%
  filter(!is.na(year)) %>%
  filter(
```

```r
    1980 <= year, year <= 2024
  )%>%
  mutate(
    lower_prediction = mean_prediction - std_prediction,
    upper_prediction = mean_prediction + std_prediction
  ) %>%
  group_by(level, year) %>%
  summarise(
    y_mean = n_distinct(analysis_id[0.5 <= mean_prediction]),
    y_lower = n_distinct(analysis_id[0.5 <= lower_prediction]),
    y_upper = n_distinct(analysis_id[0.5 <= upper_prediction])
  ) %>%
  group_by(level) %>%
  mutate(
    log_y_mean_01 = scales::rescale(log(y_mean), to = c(0,1), na.rm=T),
  ) %>%
  ungroup()%>%
  mutate(
    oro_type = factor(
      level,
      levels = typeAES$level,
      labels = typeAES$label
    )
  )



## Plot 1
# A stacked bar plot with year on x axis, proportion of publications by author country on the y
# overlain with the interannual trend in N publications
# faceted by ORO type

# This shows that the inflection in N publications at ~2000 is due to MRE-Ocean

countryProp_ggp <- ggplot(
  data = oro_year_count_long,
  aes(x=year)
)+
  geom_col(position = "stack", aes(y=prop, fill = country))+
  geom_line(data = oroPub01,
            aes(y=log_y_mean_01), col="#00f3ff")+
  facet_wrap(vars(oro_type), ncol = 2, scales="free_x")+
  scale_x_continuous(limits = c(1980,2024))+
  scale_fill_brewer(type = "qual")+
  guides(fill = guide_legend(nrow=3))+
  labs(
    x="Year",
    y = "Proportion of total publications",
    fill = "Country"
  )+
  theme_minimal()+
  theme(
    legend.position = "bottom"
  )

countryProp_ggp


ggsave(
  here::here("figures/supplemental/propCountAffilByOroYear.pdf"),
  plot = countryProp_ggp,
  width = 6, height=7
)




## Plot 2:
## zoom in on MRE-Ocean
mreO_countryProp_ggp <- ggplot(
  data = oro_year_count_long %>%
    filter(oro_type == "MRE-Ocean"),
  aes(x=year)
)+
  geom_col(position = "stack", aes(y=prop, fill = country))+
  geom_line(data = oroPub01%>%
    filter(oro_type == "MRE-Ocean"),
            aes(y=log_y_mean_01), col="#00f3ff")+
  scale_x_continuous(limits = c(1980,2024))+
  guides(fill = guide_legend(nrow=3))+
  scale_fill_brewer(type = "qual")+
  labs(
    x="Year",
    y = "Proportion of total publications",
    fill = "Country"
  )+
  theme_minimal(base_size = 10)+
  theme(
    legend.position = "bottom"
  )

mreO_countryProp_ggp


## Which countries in the 'Other' category are causing this increase?

oro_year_count_long %>%
    filter(oro_type == "MRE-Ocean", year %in% c(2002,2003)) %>%
  pivot_wider(names_from = year, values_from = count, id_cols = c(country_unique))%>%
  replace_na(list(`2002`=0, `2003`=0))%>%
  mutate(difference = (`2003`-`2002`)) %>%
  arrange(desc(difference)) %>%
  filter(!is.infinite(difference))%>%
  select(country_unique, `2002`, `2003`, difference) %>%
  filter(!(country_unique %in% affTypeTop10))%>%
  head()
```

```
#   country_unique `2002` `2003` difference
#   <chr>           <int>  <int>      <int>
# 1 Brazil              0     82         82
# 2 France              1      7          6
# 3 Canada              3      9          6
# 4 Italy               0      5          5
# 5 Norway              3      8          5
# 6 Spain               2      5          3

# looks like brazil went from 0 to 82 publications...

## Plot 3: Time series of Brazil publications
brazilPub_ggp <- oro_year_count_long %>%
    filter(oro_type == "MRE-Ocean", country_unique == "Brazil")%>%
  ggplot(aes(year, log(count)))+
  geom_line()+
  scale_y_continuous(
    name = "log(N MRE-Ocean publications)",
    sec.axis = sec_axis( transform=~exp(.), name="N publications")
  ) +
  labs(
    x = "Year"
    # y = "Brazil affiliated publications (N)"
  )+
  theme_minimal(base_size = 10)
brazilPub_ggp


ggarrange(
  plotlist = list(mreO_countryProp_ggp, brazilPub_ggp),
  labels = paste0(letters[1:2], ")"),
  label.x = 0, vjust =1.2, font.label = list(size = 10),

  align = "h"
)  %>%
  ggpubr::ggexport(filename = here::here("figures/supplemental/MRE_brazil_drivers.pdf"),
                   nrow = 1,
           width = 7, height= 4)
```

Case 2: Drivers of deployment break points

For CCS, Efficiency, CDR-OAE and MRE-Ocean, looks like change points in deployment time series – what could have caused these increases?

To explore this, fit a Bayesian multiple change point model (adapted from Cahill et al. 2015, doi: 10.1088/1748-9326/10/8/084002). This model estimates the number of probable change points, and then for each change point the posterior distribution.

Use the following response variable distributions depending on response variable: CDR-OAE - N field trials/start ups - count data - poisson MRE-Ocean - Installed capacity (MW) - non-negative continuous - Gamma Efficiency - Shipping carbon efficiency - non-negative continuous - Gamma CCS - Storage capacity (Mt) - non-negative continuous - Gamma

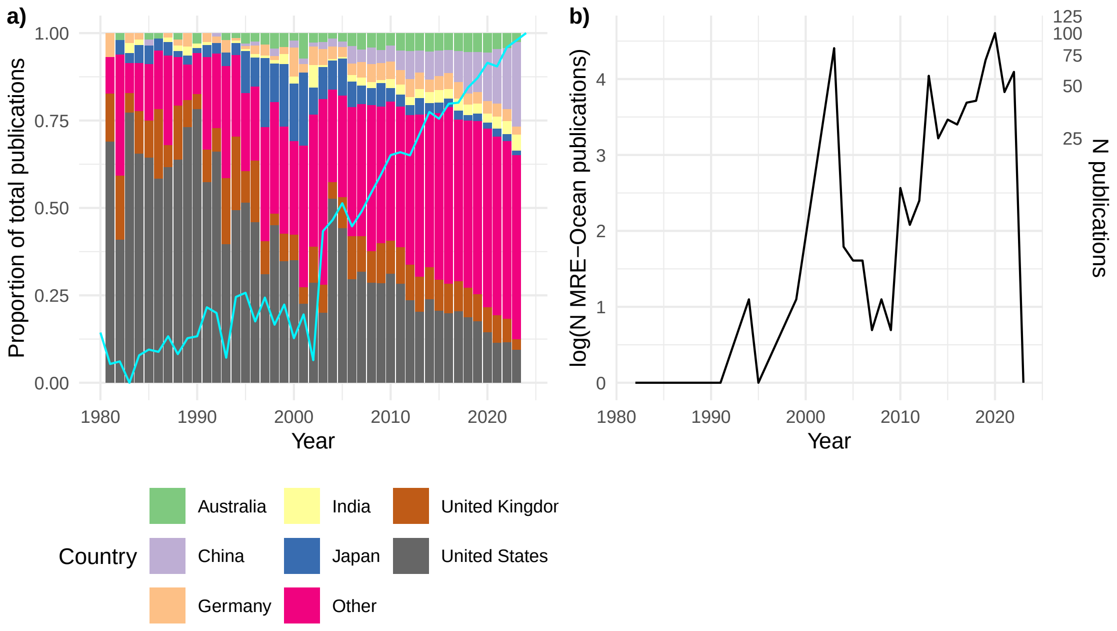

Figure 4: Brazil publications on MRE-Ocean contribute to inflection point in N publications

```
## The code below analyses all components, but we will only use deployment...

# The oro types to analyse
oroTypes = c("CCS","Efficiency","MRE-Ocean","CDR-OAE")
# The different time series components to analyse (e.g. deployment, legislation, policy, posts, etc)
# components <- unique(allComponentDat_model$component)
components <- "deployment"

# Jags initialization parameters
jagsInits <- list(
  # Maximum number of changepoints to look for
  "K_max"=2,
  # Time buffer for looking for change points.
  # i.e. don't find a change point at the first or last year
  "cpmin"="jagsData$MINX+1",
  "cpmax"="jagsData$MAXX-1")

# Compile all data into a list to iterate through
dataList <- list()
for(c in components){
  dataList[[c]] <- allComponentDat_model %>% filter(oro_type %in% oroTypes, component == c)

}

# Compile all model inputs into a list to iterate through
modelInputs <- list(
  "data" = dataList,
  "bugsModFiles" = list(
    "deployment" = "R/bugs-models/GammaModMultCP"
  ),
```

```r
  "jagsData" = list(jagsInits)[rep(1,length(components))]
)



## Loop through all oro types and time series (components) to run change point analysis
cutpoint_results <- data.frame()
cutpoint_densities <- data.frame()

for(oro in oroTypes){
  # oro = "CCS"

  for(i in 1:length(components)){
    # i = 4

    print(paste(oro,":", components[i]))

    # get model to fit
    modFile <- modelInputs$bugsModFiles[[components[i]]]

    # If OAE deployment, y is poisson not Gamma so change
    if(components[i] == "deployment" & oro %in% c("CDR-OAE")){
      modFile <- "R/bugs-models/PoissonModMultCP"
    }

    # Format data
    jagsData <- modelInputs$jagsData[[i]]
    myinits = list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 123)
    dat <- modelInputs$data[[components[i]]][modelInputs$data[[components[i]]]$oro_type == oro,]
    if(nrow(dat) < 5){
      print("Insufficient data")
      next
    }
    yearLims <- range(modelInputs$data[[i]][
      modelInputs$data[[components[i]]]$oro_type == oro,"year"
    ], na.rm=T)

    # Format specific inputs depending on the model

    if(grepl("Poisson", modFile)){
      # Fill missing years with 0 values
      dat <- dat %>% tidyr::complete(year = yearLims[1]:yearLims[2])
      dat$y[is.na(dat$y)] <- 0
      dat$y <- round(dat$y) # force discrete y values
      dat <- dat %>% tidyr::fill(component, oro_type, variable_name)

    }

    if(grepl("Gamma", modFile)){
      # Fill missing years with 0 values
      dat <- dat %>% tidyr::complete(year = yearLims[1]:yearLims[2])
      dat$y[is.na(dat$y)] <- 0
      dat$y[dat$y==0] <- 0.1
      dat <- dat %>% tidyr::fill(component, oro_type, variable_name)
    }

    # make sure no NAs
    dat <- na.omit(dat)

    # Skip if after subsetting data there is insufficient data points
    if(nrow(dat) < 5){
      print("Insufficient data after subsetting")
      next
    }
    #with(dat, plot(year, y))

    jagsData$y <- dat$y
    jagsData$x <- dat$year
    jagsData$MINX <- min(jagsData$x)
    jagsData$MAXX <- max(jagsData$x)
    jagsData$cpmin <- eval(parse(text = jagsData$cpmin))
    jagsData$cpmax <- eval(parse(text = jagsData$cpmax))

    if((jagsData$cpmax-jagsData$cpmin)<5){
      jagsData$cpmin <- jagsData$MINX
      jagsData$cpmax <- jagsData$MAXX
    }

    # jagsData$log_y <- log(dat$y)
    # jagsData$N <- length(dat$y)

    ## Fit mdel
    # modFile <- "R/bugs-models/GammaModMultCP"
    bugs.model <- readChar(modFile, file.info(modFile)$size)

    jagsModel <- tryCatch(
      {
        rjags::jags.model(
          file = textConnection(bugs.model),
          data = jagsData,
          inits = myinits,
          n.chains = 5,
          n.adapt = 1500,
          quiet = FALSE
        )
        },
     error = function(e){
       return(NULL)
     }
    )
    if(is.null(jagsModel)){
      next
    }


    update(jagsModel, 1000) # burnin
```

```r
  s = rjags::coda.samples(
        model = jagsModel,
        variable.names = c("alpha","beta","K","x_cp","z","pk"),
        n.iter = 1000,
        quiet = FALSE
      )

  qs <- summary(s)$quantile
  # qs
  # plot(s[,"pk[2]"]) # check mixing

  # Subset to only the change points that are likely
  # And if there are likely change points, only those that indicate a positive change
  K <- round(quantile(qs["K",], 0.5))
  if(grepl("ZIP", modFile)){
    K <- sum(0.5 <= qs[grepl("pk", rownames(qs)),"50%"])
  }

  if(K == 0){
    print("No significant change points")
    next
  }else{
    pk <- qs[grep("pk", rownames(qs)),"50%"] # probability for each change point
    x_cp <- qs[grep("x_cp", rownames(qs)),"50%"] # locations of each change point
    # arrange in order of decreasing probability to find the most K probable
    pkIndKeep <- order(pk, decreasing=T)
    pkIndKeep <- pkIndKeep[1:K]
    # then re-arrange in chronological order for calculating trends of segments
    pkIndKeep <- sort(pkIndKeep)

    # Find which change points mark a positive change in trend
    trends <- vector("numeric", K+1) # store the trend for each segment

    # starts <- c(min(jagsData$x), round(x_cp[pkIndKeep]))
    starts <- c(min(jagsData$x), sort(round(x_cp[pkIndKeep])))
    match(starts, round(x_cp[pkIndKeep]))
    ends <- c(round(x_cp[pkIndKeep]), max(jagsData$x))
    for(k in 1:(K+1)){
      if(grepl("Binom", modFile)){
        trends[k] <- jagsData$y[which.min(abs(jagsData$x-ends[k]))] - jagsData$y[which.min(abs(jagsDa
      }else if(grepl("ZIP|Pois|Gamma", modFile)){
        trends[k] <- sum(jagsData$y[which.min(abs(jagsData$x-starts[k])):which.min(abs(jagsData$x-ends
      }

    }
    # only keep the positive changes in trends
    matchInd <- match(starts[-1], round(x_cp[pkIndKeep]))
    pkIndKeep <- pkIndKeep[matchInd[which(0 < diff(trends))]]

    if(length(pkIndKeep) == 0){
      print("no changepoints marking increasing trend")
      next
    }else{
        pkIndNames <- names(x_cp[pkIndKeep])

        # # If there are two cut points within 3 years of each other,
        # # keep most probable one
        # if(1 < length(pkIndKeep)){
        #   if(diff(qs[pkIndNames,"50%"]) < 3){
        #     pkIndNames <- pkIndNames[1]
        #   }
        # }

        # save probability density of the cut points for plotting
        calc_density <- function(vals){
          dens <- density(vals, n=100)
          return(data.frame("year"=dens$x, "cp_density"=dens$y))
        }
        densTemp <- do.call(rbind, s)
        densTemp <- as.matrix(densTemp[,pkIndNames], ncol=length(pkIndNames))
        densTemp <- do.call(rbind.data.frame, apply(densTemp, 2, function(x) calc_density(x)))
        densTemp$cp_id <- rep(pkIndNames, 100)[sort(rep(1:length(pkIndNames),100))]


        # Save summary of quantiles
        dfTemp <- do.call(rbind, apply(qs[pkIndNames,,drop=FALSE], 1, function(x) as.data.frame(x)))
        dfTemp$quantile = rep(colnames(qs), length(pkIndNames))
        dfTemp$cp_id <- rep(pkIndNames, 5)[sort(rep(1:length(pkIndNames),5))] #rep(1:length(pkIndNames)
        rownames(dfTemp) <- NULL
        # Cap the distribution of cutpoints at the hard limits of the data
        colnames(dfTemp)[which(colnames(dfTemp) == "x")] <- "year"
        dfTemp$year <- ifelse(dfTemp$year < yearLims[1], yearLims[1], dfTemp$year)
        dfTemp$year <- ifelse(yearLims[2] < dfTemp$year, yearLims[2], dfTemp$year)
        # Add id variables
        addIdVariables <- function(df){
          df %>% mutate(
                        oro_type = oro,
                        component = dat$component[1],
                        variable_name = dat$variable_name[1])
        }
        dfTemp <- addIdVariables(dfTemp)
        densTemp <- addIdVariables(densTemp)

        ## Bind data to results
        cutpoint_results <- rbind(cutpoint_results, dfTemp)
        cutpoint_densities <- rbind(cutpoint_densities, densTemp)
        # remove jags model
        rm(jagsModel)
      }
    }



  } # end looping through ORO types
} # end looping through bugs models
```

```
## Save results
save(cutpoint_results,cutpoint_densities, file=here::here("outputs/cutpointResults_mitigation_deploymen
```

```
# Load data from previous chunk
load(here::here("outputs/cutpointResults_mitigation_deployment.RData"))

# ORO types to analyse
oroTypes <- c("CCS","Efficiency","MRE-Ocean","CDR-OAE")

# Get a summary of the change points
deplChangePoints <- cutpoint_results %>%
    filter(component == "deployment", oro_type %in% oroTypes)%>%
  pivot_wider(names_from = "quantile", values_from = "year")



# Get the posterior distribution to plot
yBackTransform <- function(value, ogRange, newRange){
  b <- diff(newRange) / diff(ogRange)
  a <- newRange[1] - b * ogRange[1]
  return(a + b * value)
}

cpDeployPlotData <- data.frame()
cpDensityPlotData <- data.frame()
yAxisBreaks <- data.frame()
for(oro in oroTypes){

  rawDat <- allComponentDat_model %>%
    filter(oro_type == oro, component == "deployment") %>%
    mutate(y_scale = scales::rescale(y, c(0,1), na.rm=T)) %>%
    mutate(
      label = paste(
        scales::number(y, accuracy =1, scale_cut = append(scales::cut_long_scale(), 1, 1),
                       big.mark = ".", decimal.mark = ",")
      )
    )
  ogRange <- range(rawDat$y, na.rm=T)
  newRange <- c(0,1)
  ogBreaks <- pretty(ogRange, n = 4)
  newBreaks <- yBackTransform(ogBreaks, ogRange, newRange)
  if(1 < ogRange[2]){
    ybl <- scales::number(ogBreaks, accuracy =1, scale_cut = append(scales::cut_long_scale(), 1, 1),
                       big.mark = ",", decimal.mark = ".")
  }else{
    ybl <- scales::number(ogBreaks, accuracy =0.001, scale_cut = append(scales::cut_short_scale(), 1, 1
                       big.mark = ",", decimal.mark = ".")
  }

  yAxisBreaks <- yAxisBreaks %>%
    bind_rows(
      data.frame(
        y_breaks = ogBreaks,
        y_breaks_label = ybl,
        y_scale_breaks = newBreaks,
        oro_type = oro,
        component = "deployment"
      )
    )

  cpDens <- cutpoint_densities %>%
    filter(component == "deployment") %>%
    mutate(cp_group = paste(oro, component, cp_id)) %>%
    group_by(cp_group) %>%
    mutate(
      cp_density = scales::rescale(cp_density, to=c(0,1))
    )

  cpDeployPlotData <- cpDeployPlotData %>% bind_rows(rawDat)
  cpDensityPlotData <- cpDensityPlotData %>% bind_rows(cpDens)

}

stripLabsDF <- cpDeployPlotData %>%
  distinct(oro_type, variable_name) %>%
  mutate(
    labels = paste(oro_type, variable_name, sep="\n")
  )
stripLabs <- stripLabsDF$labels
names(stripLabs) <-stripLabsDF$oro_type


changePoint_ggp <- ggplot()+
  geom_ribbon(data = cpDensityPlotData,
       aes(x=.data$year, ymax=.data$cp_density, ymin=0, group=.data$cp_group), #, fill = .data$cp_id
       fill = "lightgrey",
        alpha = 0.5)+ # , ymin=-Inf,
  geom_text(data = yAxisBreaks,
            aes(x=-Inf, y=.data$y_scale_breaks, label = .data$y_breaks_label),
            size = 3, col = "black", hjust = 1, vjust=0)+
  geom_vline(data = cutpoint_results %>%
    filter(component == "deployment", oro_type %in% oroTypes, quantile %in% c("25%","75%")),
             aes(xintercept=.data$year),
             linewidth = 0.5, linetype = "dashed")+

  geom_line(data = cpDeployPlotData,
            aes(x=.data$year, y=.data$y_scale),
            col = "black", linewidth = 1)+
  scale_y_continuous(
    breaks = NULL, #pretty_breaks(n = 4),
    name = ""
  )+
  scale_x_continuous(name = "Year")+

  # Color scales
```

```r
  # scale_fill_manual(breaks = c("x_cp[1]","x_cp[2]"), values = c("lightblue","lightpink"), guide="none

  facet_wrap(vars(oro_type), ncol=1, labeller = labeller(oro_type = stripLabs))+
  coord_cartesian(clip = "off")+
  theme_minimal()+
  theme(
    axis.text.y = element_blank(),
    plot.margin = unit(c(0.5,1,1,1.5), units = "cm"),
    legend.position = "bottom"
  )

changePoint_ggp

# ggsave(
#   here::here("figures/supplemental/changePointTimeSeries_deployment.pdf"),plot=changePoint_ggp,
#   width = 5, height=6, units='in'
# )
```

Look for external drivers of deployment change points

```r
# Load cut point results
load(here::here("outputs/cutpointResults_mitigation_deployment.RData"))

oroTypes <- c("CCS","Efficiency","MRE-Ocean","CDR-OAE")


# Get a summary table of the quantiles of each cut point
deplChangePoints <- cutpoint_results %>%
    filter(component == "deployment", oro_type %in% oroTypes)%>%
  pivot_wider(names_from = "quantile", values_from = "year")


# Write table to csv
write.csv(deplChangePoints, file = here::here("outputs/cutpointSummaryTable_mitigation_deployment.csv")



# investigate raw data for these years --------------

## MRE-Ocean -----------------
MREOcean_changeYears <- readxl::read_excel(here::here("data/raw-data/external/IRENA-electricity-statist
                         sheet = "Country") %>%
  mutate(year = as.numeric(Year), oro_type = "MRE-Ocean")%>%
  filter(
    Technology %in% c("Marine energy"),
    !is.na(`Electricity Installed Capacity (MW)`),
    deplChangePoints$`25%`[deplChangePoints$oro_type == "MRE-Ocean"]-5 <= year &
      year <= deplChangePoints$`75%`[deplChangePoints$oro_type == "MRE-Ocean"]+5
  )


topDiff <- MREOcean_changeYears %>%
  arrange(Country, year) %>%
  group_by(Region) %>%
  mutate(difference = `Electricity Installed Capacity (MW)` - first(`Electricity Installed Capacity (MW
  filter(0 < difference)%>%
  select(
    Region, Country, year, `Electricity Installed Capacity (MW)`, difference
  ) %>%
  arrange(desc(difference))
topDiff %>% head()

mreOceanTime_ggp<- ggplot(MREOcean_changeYears %>%
         group_by(Region, year) %>%
         summarise(y = sum(`Electricity Installed Capacity (MW)`, na.rm=T)),
       aes(as.integer(year), y, col = Region))+
  # geom_rect(
  #   xmin = deplChangePoints$`25%`[deplChangePoints$oro_type == "MRE-Ocean"],
  #   xmax = deplChangePoints$`75%`[deplChangePoints$oro_type == "MRE-Ocean"],
  #   ymin = -Inf, ymax = Inf,
  #   fill = "lightgrey", alpha = 0.5, col="transparent"
  # )+
  # geom_vline(xintercept = deplChangePoints$`50%`[deplChangePoints$oro_type == "MRE-Ocean"], col="red"
  geom_line()+
  scale_x_continuous(breaks = scales::pretty_breaks())+
  geom_text(data = data.frame(
    year = topDiff$year[1]-3,
    y= 150,
    label = stringr::str_wrap("Sihwa Lake Tidal Power Station (255 MW), Republic of Korea", width = 20)
  ), aes(year,y,label = label),size=3, inherit.aes = FALSE, hjust=0.5)+
  labs(
    x="Year",
    y = "Electricity Installed Capacity (MW)",
    col = "Region"
  )+
  theme_minimal()+
  theme(
    legend.position = "right"
  )
mreOceanTime_ggp

# The jump in Asia was due to Republic of Korea -- Sihwa Lake Tidal Power Station 255 MW


## CCS -------------------------------
ccs_changeYears <- readxl::read_excel(here::here("data/raw-data/external/CRSC_CYCLE_4_FINAL_2024_160724
  filter(grepl("shore|sea", area, ignore.case=T), !is.na(year_of_publication)) %>%
  filter(project_spec == "YES") %>%
  mutate(
    year = as.numeric(year_of_publication),
    `Storage capacity (Mt)` = rowSums(select(., sum_low, sum_mid, sum_high), na.rm = TRUE)
    ) %>%
  filter(
    deplChangePoints$`25%`[deplChangePoints$oro_type == "CCS"]-5 <= year &
      year <= deplChangePoints$`75%`[deplChangePoints$oro_type == "CCS"]+5
  )
```

```r
summary(ccs_changeYears)



ccsStorTime_ggp <- ggplot(data=ccs_changeYears %>%
         group_by(country, year) %>%
         summarise(y = sum(`Storage capacity (Mt)`, na.rm=T)))+
  # geom_vline(xintercept = deplChangePoints$`50%`[deplChangePoints$oro_type == "CCS"], col="red")+
  # geom_rect(
  #   xmin = deplChangePoints$`25%`[deplChangePoints$oro_type == "CCS"],
  #   xmax = deplChangePoints$`75%`[deplChangePoints$oro_type == "CCS"],
  #   ymin = -Inf, ymax = Inf,
  #   fill = "lightgrey", alpha = 0.5, col="grey"
  # )+
  geom_line(aes(year, y, col = country))+
  # geom_line(data=ccs_changeYears %>%
  #        group_by(region, year) %>%
  #        summarise(y = sum(`Storage capacity (Mt)`, na.rm=T)),
  #      aes(year, y, col = region))+
  geom_point(aes(year, y, col = country), size=3)+
  labs(
    x="Year",
    y = "Storage capacity (Mt)",
    col = "Country"
  )+
  theme_minimal()+
  theme(
    legend.position = "right"
  )
ccsStorTime_ggp

norwayCCS <- ccs_changeYears %>%
  ungroup()%>%
  filter(country == "Norway", year == 2014) %>%
  as.data.frame()

library(ggOceanMaps)
norCCS2014 <- qmap(norwayCCS, size=`Storage capacity (Mt)`,bathymetry=TRUE)





## Efficiency ----------------------------

eff_changeYears <- readxl::read_excel(
  here::here("data/raw-data/external/IEA-shipping-energy-efficiency.xlsx"),
  sheet = "Data") %>%
  mutate(
    y= 1/carbon_intensity_gCO2_per_tkm,
    year = as.numeric(format(as.Date(paste0(year,"-01-01")), "%Y")),
    variable_name = "Shipping carbon efficiency (tkm/gCO2)",
    component = "deployment",
    oro_type = "Efficiency"
    ) %>%
  filter(
    deplChangePoints$`25%`[deplChangePoints$oro_type == "Efficiency"]-5 <= year &
      year <= deplChangePoints$`75%`[deplChangePoints$oro_type == "Efficiency"]+5
  )

IMOLab <- data.frame(
    year = 2011,
    y= mean(eff_changeYears$y, na.rm=T),
    label = stringr::str_wrap("IMO Resolution MEPC.203(62) (2011)", width = 15))

effTime_ggp<- ggplot(eff_changeYears,
       aes(year, y))+
  geom_line()+
  geom_text(data = IMOLab, aes(year,y,label = label), size=3, inherit.aes = FALSE, hjust=1, nudge_x=-0.
  geom_vline(xintercept = 2011, col="red")+
  labs(
    x="Year",
    y = "Shipping carbon efficiency (tkm/gCO2)"
  )+
  theme_minimal()+
  theme(
    legend.position = "none"
  )
effTime_ggp


## CDR-OAE ---------------------------------------
# Ocean visions field trials
fieldTrials <- readxl::read_excel(
  here::here("data/raw-data/external/Ocean-visions-mCDR-field-trial-database.xlsx"),
  sheet = "Data") %>%
  mutate(
    `Start of Pilot` = replace(`Start of Pilot`, `Start of Pilot` %in% c("5.12.2023","2023"), as.numeri
    trialID = row_number()
  ) %>%
  mutate(year = as.Date(as.numeric(`Start of Pilot`), origin = "1899-12-30")) %>%
  mutate(year = ifelse(year == as.Date("1952-12-30"), as.Date("2023-01-01"), year)) %>%
  mutate(year = as.Date(year, origin = as.Date("1970-01-01"))) %>%
  separate_rows(`All CDR Methods`, sep=",") %>%
  mutate(
    oro_type = case_when(
      grepl("OAE|Alkalinity|Weathering", `All CDR Methods`) ~ "CDR-OAE",
      grepl("Upwelling", `All CDR Methods`) ~ "CDR-BioPump",
      TRUE ~ "Other"
    ),
    year = as.numeric(format(year, "%Y")),
    source = "Field trial"
  ) %>%
  filter(!is.na(year) & oro_type != "Other")
```

```r
# Founding year of mcdr startups
# Need to de-duplicate and add year

GESAMP_companies <- readxl::read_excel(
  here::here("data/raw-data/external/GESAMP_wg41_ocean_climate_intervention_projects_31_may_2024.xlsx")
  select(Company, Type, Website) %>%
  mutate(
    Company = trimws(gsub("[[:punct:]]", "", Company)),
    source = "GESAMP WG 41"
  )%>%
  filter(!is.na(Company))

# Load in OceanNET companies and de-deuplicate from GESAMP companies
OceanNET_companies <- read.csv(
  here::here("data/raw-data/external/OceanNETs_D18_oceanbased_CDR_companies/dataset/D1_8_database_ocean
  mutate(
    Company = trimws(gsub("[[:punct:]]", "", Company)),
    source = "OceanNETs (2020) D1.8 database"
  )%>%
  filter(!is.na(Company), !is.na(Lat))%>%
  select(-c(contains("X"))) %>%
  filter(!(tolower(Company) %in% tolower(GESAMP_companies$Company)), Company != "Qilibrium")

## Join and write to csv file so I can look up the years on linkedIN
# # Join
# mCDRCompanies <- GESAMP_companies %>%
#   bind_rows(OceanNET_companies)
#
# # write to file
# write.csv(mCDRCompanies, file = here::here("data/derived-data/mCDR-companies-dedup.csv"))
#


## read in the years
mCDRCompanies <- readxl::read_excel(here::here("data/raw-data/mCDR-companies-linkedin.xlsx")) %>%
  filter(!is.na(`Company founded on (linkedin)`)) %>%
  rename(year = `Company founded on (linkedin)`) %>%
  select(Company, year) %>%
  left_join(GESAMP_companies %>% select(Company, Type), by="Company") %>%
  left_join(OceanNET_companies %>% select(Company, Type), by="Company") %>%
  mutate(
    Type = ifelse(is.na(Type.x), Type.y, Type.x)
  ) %>%
  select(Company, Type, year) %>%
  mutate(
    oro_type = case_when(
      grepl("Upwell|fertilization", Type, ignore.case=TRUE) ~ "CDR-BioPump",
      # grepl("Biomass sinking|Farming|Harvesting|Aquaculture", Type, ignore.case=TRUE) ~ "CDR-Cult",
      grepl("OAE|Alkalinity|weathering", Type) ~ "CDR-OAE",
      TRUE~"CDR-Other"
    ),
    source = "Startup",
    year = as.numeric(year)
  ) %>%
  filter(oro_type != "CDR-Other")

# Ocean visions community and www.cdr.fyi/leaderboards can't be scraped because they use private APIs
# Although the latter a list of suppliers can be found here without metadata
# https://www.cdr.fyi/api/search




## Combine data sources



oae_changeYears <- fieldTrials %>%
  bind_rows(mCDRCompanies) %>%
  filter(
    oro_type == "CDR-OAE",
    2020 <= year &
      year <= 2022
  )

# A combinationo f startups and field trials
oae_changeYears %>%
  group_by(
    year, source
  )%>%
  summarise(
    n = n()
  )

# Whos conducting the field trials in 2022?
unique(oae_changeYears$`Leading Organization`[oae_changeYears$year == 2022]) # "Vesta"   "GEOMAR"  "Lim
fieldtrials_2022 <- unique(oae_changeYears$`Leading Organization`[oae_changeYears$year == 2022])
fieldtrials_2022 <- fieldtrials_2022[!is.na(fieldtrials_2022)]
fieldtrials_2022 <- c("Field trials (2022):", fieldtrials_2022)
fieldtrials_2022 <- paste(fieldtrials_2022, collapse = "\n")

# Who are the startups in 2021?
unique(oae_changeYears$Company[oae_changeYears$year == 2021])
startups_2021 <- unique(oae_changeYears$Company[oae_changeYears$year == 2021])
startups_2021 <- startups_2021[!is.na(startups_2021)]
startups_2021 <- c("Start ups (2021):", startups_2021)
startups_2021 <- paste(startups_2021, collapse = "\n")

#  [1] NA                                  "Ebb Carbon"                        "Cequest"
#  [4] "Pronoe"                            "The Charles Darwin Rescue Plan" "Skyology"
#  [7] "Vycarb"                            "Carbon to Sea"                     "EDAC Labs"
# [10] "The acidd project"                 "CarbonRun"


oaeTime_ggp<- ggplot(fieldTrials %>%
                      bind_rows(mCDRCompanies) %>%
                      filter(
                        oro_type == "CDR-OAE",
                        2018 <= year &
                          year <= 2024
                      ) %>%
                      group_by(
                        year, source
                      )%>%
                      summarise(
                        n = n()
                      ),
       aes(year, n, fill = source))+
  geom_col()+
  scale_fill_brewer(
    name = "Data\nsource",
    breaks = c("Field trial","Startup"),
    palette = "Dark2"
  )+
  geom_text(
    data = data.frame(
      year = 2022, n=10, label = startups_2021
    ),
    aes(year, n, label = label), size=2.7, inherit.aes = FALSE, hjust=1, nudge_x=-0.5, check_overlap = 
  geom_text(x=2021.5, y=13, label = fieldtrials_2022, size=2.7, inherit.aes = FALSE, hjust=0, nudge_x=-
            check_overlap=TRUE)+
  scale_x_continuous(breaks = 2018:2024)+
  geom_vline(xintercept = 2020.5, col="red")+
  labs(
    x="Year",
    y = "N Field trials/Start ups"
  )+
  theme_minimal()+
  coord_cartesian(clip="off")+
  theme(
    legend.position = "right"
  )
oaeTime_ggp




## Plot -----------------------------------------------


# ggpubr::ggarrange(
#   plotlist = list(mreOceanTime_ggp+theme(plot.margin = unit(c(1.2,0,0,0.5),"cm")),
#                   ccsStorTime_ggp+theme(plot.margin = unit(c(1.2,0,0,0.5),"cm")),
#                   norCCS2014+theme(plot.margin = unit(c(1.2,0,0,0.5),"cm")),
#                   effTime_ggp+theme(plot.margin = unit(c(1.2,0,0,0.5),"cm")),
#                   oaeTime_ggp+theme(plot.margin = unit(c(1.2,0,0,0.5),"cm"))
#                   ),
#   labels = paste0(letters[1:5], ") ",c("MRE-Ocean","CCS","CCS - Norway","Incr. Efficiency","mCDR-OAE"
#   label.x = 0, vjust =1.2, font.label = list(size = 10),
#
#   align = "hv"
# )  %>%
#   ggpubr::ggexport(filename = here::here("figures/supplemental/qualDepChangePointPlots.pdf"),
#                    nrow = 3,
#            width = 15, height= 7)
```

Part 1 summary figure

```
plotList = list(
    brazilPub_ggp,
    mreOceanTime_ggp+theme(plot.margin = unit(c(1.2,0,0,0.5),"cm")),
                  ccsStorTime_ggp+theme(plot.margin = unit(c(1.2,0,0,0.5),"cm")),
                  norCCS2014+theme(plot.margin = unit(c(1.2,0,0,0.5),"cm")),
                  effTime_ggp+theme(plot.margin = unit(c(1.2,0,0,0.5),"cm")),
                  oaeTime_ggp+theme(plot.margin = unit(c(1.2,0,0,0.5),"cm"))
                  )


p_all <- wrap_plots(plotList, ncol = 2, widths = c(1,1)) +
  theme(plot.tag.position = c(0, 1),  # tags top-left
        plot.tag = element_text(size = 10))+
  plot_annotation(
  tag_levels = "a",
  tag_prefix = "",
  tag_suffix = ")"
)

# # Save
# ggsave(
#   here::here("figures/main/allChangePointPlots.pdf"),
#   p_all,
#   width = 15, height = 7
# )
```

Part 2: Quantitative analysis

2.1: Match/mis-match (non-temporal)

Key message Public interest does not match well with publications (not significant). Legislation and policy show better alignment (significant).

NB: Because compares OROs between each other, cannot include deployment because all the metrics are on different scales

Kieran questions

- Did I select the correct probability distributions for the response variables?
- Do you notice any errors in the analysis/interpretation?

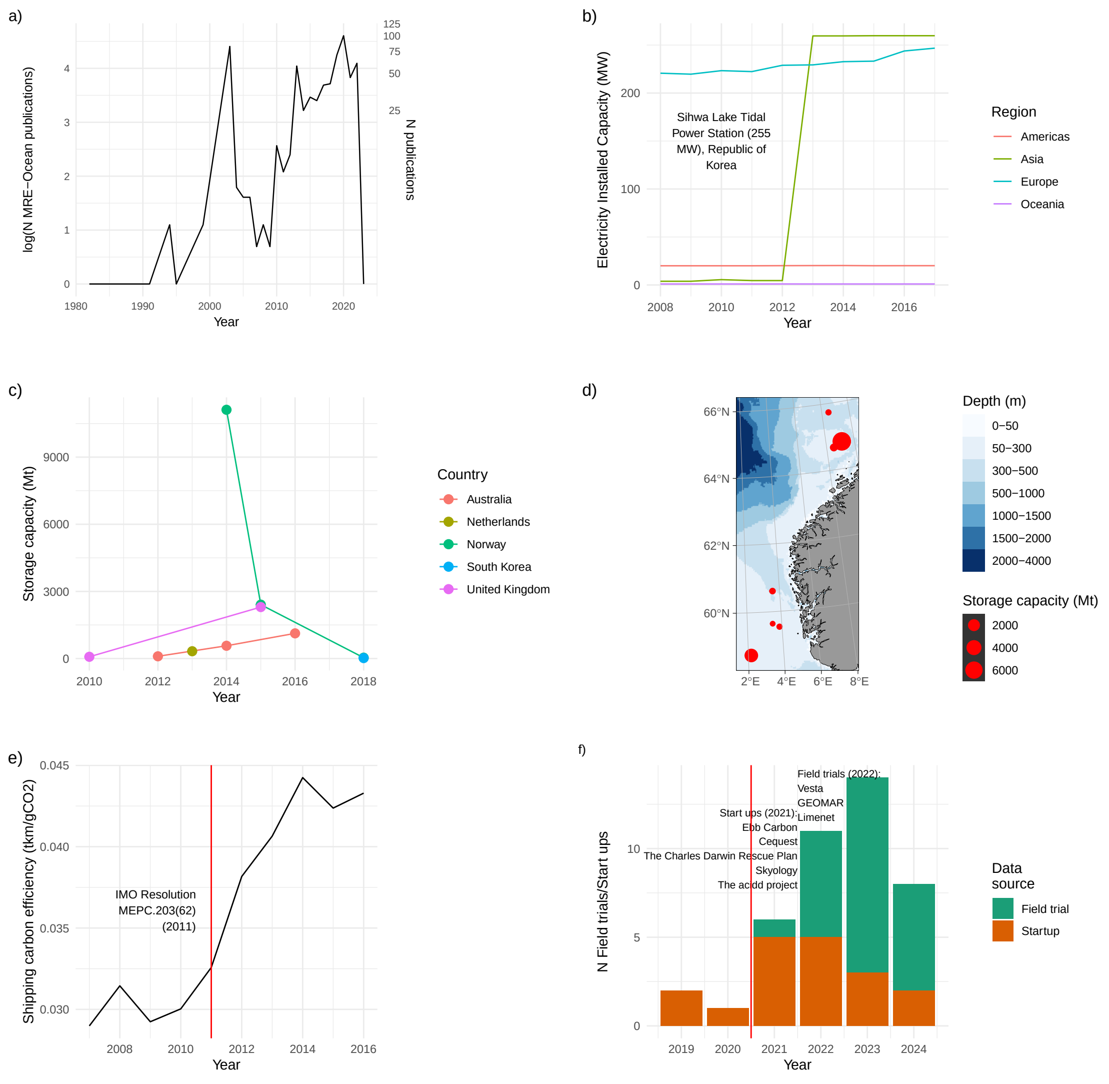

Figure 5: Qualitative analysis of change points

```
# Depending on the component (response variable type) the distribution changes, so make a lookup table
family_lookup <- allComponentDat_model %>% ungroup() %>%
  distinct(component) %>%
  select(component) %>%
  filter(!(component %in% c("public opposition","publications","deployment"))) %>%
  mutate(
    family = case_when(
      component %in% c("policy","legislation") ~ "poisson()",
      component %in% c("public support (relative)","public support agreement") ~ "binomial()",
      TRUE ~ "Gamma()"
    )
  )


## Loop through the different components and fit glm
# sum metric values across years

# different components to loop through
components <- unique(allComponentDat_model$component)
components <- components[!(components %in% c("publications","deployment","public opposition"))]

ggps <- vector("list", length(components))
names(ggps) <- components
modelResults <- vector("list", length(components))
for(c in 1:length(components)){

  ## Fit the model
  # If needed, depending on the metric, some additional data manipulations need to be made first

  if(components[c] == "public support agreement"){
    # If public support agreement, fit as a binomial model where success = public support, and failure
    tmpDat <- allComponentDat_model %>%
      filter(component %in% c("publications", "public support","public opposition")) %>%
      group_by(oro_type, component) %>%
      summarise(
        y = sum(y, na.rm=T)
      )%>%
      mutate(y=round(y)) %>%
      pivot_wider(names_from = component, values_from = y, id_cols = c(oro_type)) %>%
      na.omit()
    colnames(tmpDat) <- gsub(" ","_", colnames(tmpDat))
    # tmpDat$publications <- scale(tmpDat$publications)[,1] ## ?scale?

    mod <- glm(cbind(public_support, public_opposition) ~ publications,
               data = tmpDat,
               family = eval(parse(text = family_lookup$family[family_lookup$component == components[c]

    # get a variable on the response scale for plotting
    tmpDat<- tmpDat %>%
      mutate(
        response_y = public_support/(public_opposition+public_support)
      )
  }
```

```
else if(components[c] == "public support (relative)"){
  # If public support relative, calculate as the proportion positive posts relative to total N posts
  tmpDat <- allComponentDat_model %>%
    filter(component %in% c("publications", "public support","public interest weighted")) %>%
    group_by(oro_type, component) %>%
    summarise(
      y = sum(y, na.rm=T)
    )%>%
    mutate(y=round(y)) %>%
    pivot_wider(names_from = component, values_from = y, id_cols = c(oro_type)) %>%
    mutate(public_support_relative = `public support`/`public interest weighted`) %>%
    na.omit()
  colnames(tmpDat) <- gsub(" ","_", colnames(tmpDat))
  # tmpDat$publications <- scale(tmpDat$publications)[,1] ## ?scale?

  mod <- glm(public_support_relative ~ publications,
             weights = tmpDat$public_interest_weighted,
             data = tmpDat,
             family = eval(parse(text = family_lookup$family[family_lookup$component == components[c]
  # get a variable on the response scale for plotting
  tmpDat<- tmpDat %>%
    mutate(
      response_y = public_support_relative
    )

}else{
  # Otherwise no additional data manipulations need to be made, just fit the model
  tmpDat <- allComponentDat_model %>%
    filter(component %in% c("publications", components[c])) %>%
    group_by(oro_type, component) %>%
    summarise(
      y = sum(y, na.rm=T)
    )%>%
    pivot_wider(names_from = component, values_from = y, id_cols = c(oro_type)) %>%
    na.omit()
  colnames(tmpDat)[colnames(tmpDat) == components[c]] <- "y"

  if(family_lookup$family[family_lookup$component == components[c]] == "poisson()"){
    tmpDat$y <- round(tmpDat$y)
  }

  colnames(tmpDat) <- gsub(" ","_", colnames(tmpDat))
  # tmpDat$publications <- scale(tmpDat$publications)[,1] ## ?scale?


  mod <- glm(y ~ publications,
             data = tmpDat,
             family = eval(parse(text = family_lookup$family[family_lookup$component == components[c]
  # get a variable on the response scale for plotting
  tmpDat<- tmpDat %>%
    mutate(
      response_y = y
    )
  }


  ## Get model predictions
  predDat <- data.frame(
    publications = seq(from = min(tmpDat$publications), to = max(tmpDat$publications), length.out = 50)
  )
  preds <- predict(mod, newdata = predDat, se.fit = TRUE, type = "response")
  predDat$mean_fit <- preds$fit
  predDat$lower_fit <- preds$fit - preds$se.fit
  predDat$upper_fit <- preds$fit + preds$se.fit

  ## store model results
  modSummary <- summary(mod)
  modSummary <- as.data.frame(t(modSummary$coefficients["publications",]))
  modSummary$component = components[c]
  modelResults[[c]] <- modSummary


  pval = data.frame(
    label = paste(
    "p-value <", ifelse(
      modSummary[,grep("Pr", names(modSummary))] < 0.01, "0.01",
      (
        signif(modSummary[,grep("Pr", names(modSummary))], digits=2)
      ))
  )
  )


  ## Plot
  ggps[[c]] <- ggplot()+
    geom_ribbon(data = predDat, aes(x=publications, ymin = lower_fit, ymax = upper_fit), alpha = 0.5)+
    geom_line(data = predDat, aes(x=publications, y=mean_fit), col="red")+
    # geom_text(data = tmpDat, aes(x=publications, y = public_support/(public_opposition+public_support
    ggrepel::geom_text_repel(data = tmpDat, aes(x=publications, y = response_y, label = oro_type), forc
                             size=2.5)+
    ggrepel::geom_text_repel(data = pval, aes(label = stringr::str_wrap(label, 8), fontface=3),
                             x=Inf, y=-Inf, hjust=1, vjust=0, col="red",
                             size=2.5)+
    labs(
      x = "Publications (N)",
      y = componentAES$label[componentAES$level == components[c]]
    )+
    theme_minimal()+
    theme(
      axis.title = element_text(size=10),
      plot.margin = unit(c(0.7,0.3,0.3,0.3),"cm")
    )


}
```

```
modelResults <- do.call("bind_rows", modelResults)
modelResults$family <- family_lookup$family[match(modelResults$component, family_lookup$component)]
modelResults$component <- componentAES$label[match(modelResults$component, componentAES$level)]


# ## save
#
# write.csv(
#   modelResults,
#   here::here("outputs/publications_otherDim_glmResults.csv")
# )
#
# ggpubr::ggarrange(
#   plotlist = ggps,
#   labels = paste0(letters[1:length(ggps)],") ", componentAES$label[match(names(ggps), componentAES$le
#   label.x = 0, vjust =1.2, font.label = list(size = 10),
#   align = "hv"
#
# )%>%
#   ggpubr::ggexport(filename = here::here("figures/supplemental/allGlmFitPlots.pdf"),
#                    ncol = 3,
#            width = 10, height= 10)
#
```

```
selectedComponents <- c("policy","legislation","public interest","public support")


# ggpubr::ggarrange(
#   plotlist = ggps[names(ggps) %in% selectedComponents],
#   labels = paste0(letters[1:length(selectedComponents)],") ", componentAES$label[match(selectedCompon
#   label.x = 0, vjust =1.2, font.label = list(size = 10),
#   align = "hv"
#
# )%>%
#   ggpubr::ggexport(filename = here::here("figures/main/selectedGlmFitPlots.pdf"),
#                    ncol = 2,
#            width = 7, height= 5.5)
```

2.2 Network analysis

Key message Publications and public interest to be direct drivers of action, but that policy and legislation play an influential intermediary role

Method steps:

Step 1. For each ORO, determine if an edge exists between two nodes using variable lag transfer entropy Step 2. Generalize these findings across OROs by creating a meta network across all the OROs. If an edge exists for an ORO, it is included in the meta network. Step 3. Use the meta network to build a qualitative network model, and simulate press perturbations to the different nodes and record the impact on action.

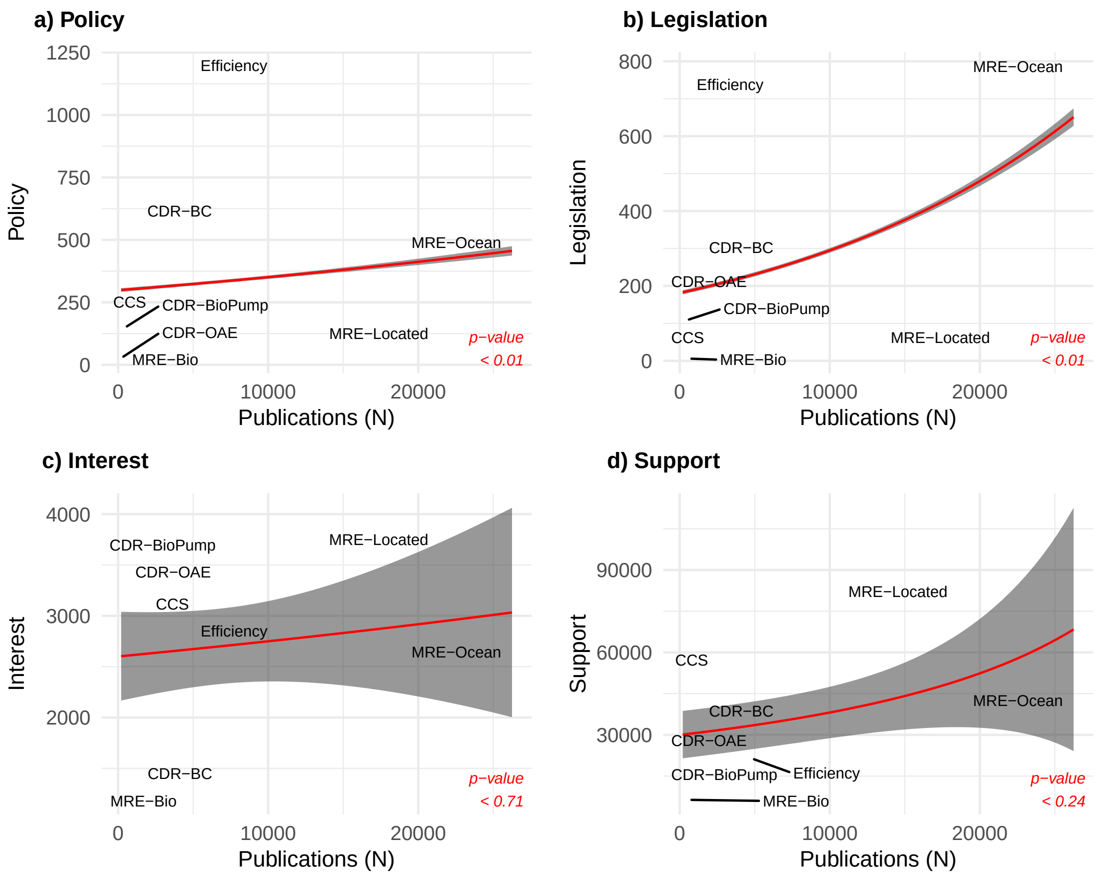

Figure 6: Match/mis-match glm fits

Step 1: Variable-lag transfer entropy

For each ORO, determine whether there are links between the time series of the different nodes in our conceptual model (publications, policy, legislation, public interest metrics, action). However, there are likely time lags between the different time series and I am uncertain of how long these lags are; and the relationships may be nonlinear.

To investigate what might be the best analysis, I think transfer entropy is a good option (see the publications below):

- Behrendt et al. 2019. RTransferEntropy— Quantifying information flow between different time series using effective transfer entropy
- Amornbunchornvej et al 2021. Variable-lag Granger Causality and Transfer Entropy for Time Series Analysis

Uses: To detect significance of information flow from X -> Y with a time lag. unlike Granger Causaility, can detect non-linear relationships.

Data Assumptions:

- evenly spaced, no NA (fill NAs with 0 in our time series)
- stationary data: refers to the idea that the statistical properties of a time series do not change over time. More specifically, a stationary time series is one in which the mean, variance, and autocorrelation structure are constant over time (no noticible changing levels, increasing variance). (use Box-Cox transform to make data normal)
- time lag is fixed (except see variable lag-Transfer entropy with bootstrapping, as bootstrapping approach increases the performance of transfer entropy methods in this task - install.packages("VLTimeCausality"))

VL-TE Pipeline:

- correct for heteroskedasity using a Box-Cox transformation to make your data normal (caret::BoxCoxTrans)
- get an iterable list of the pairwise edges for the different nodes in the time series
- Use VL-TE (bootstrapped) from the VLTTimeCausality package to get TE ratio and p-value (higher the ratio, stronger evidence for causality)
- Plot the TE ratio for each of the edges to determine the evidence for the relationship

Kieran questions

- Does this approach to use variable-lag transfer entropy seem sensible?
- Do you notice any errors in the analysis/interpretation?

```r
# Define levels to loop through

# Loop 1 -- different oros
oros <- unique(allComponentDat_model$oro_type)

# Loop 2 -- process all possible edges
# Define nodes and edges to process
nodes <- unique(allComponentDat_model$component)
edges <- expand.grid(nodes, nodes)
edges <- edges[edges[,1] != edges[,2],]

# remove edge pairs whose calculations are inter-related (all the public interest metrics)
```

```
publicEdges <- grepl("public", edges[,1]) & grepl("public", edges[,2])
edges <- edges[!publicEdges,]


# Function to make BoxCoxTransformation -- make data normal
BCTransform <- function(series){
  BCMod <- caret::BoxCoxTrans(series)
  series_trans <- predict(BCMod, series)
}

# Wrapper to try the box-cox transformation with error handling
try_BCTransform <- function(series){
  tryCatch(
    {BCTransform(series)},
    error = function(e){
      return(series)
    }
  )
}


# # Define function to only use BC transform is HeteroSkedasticity present?
# # Decided not to use and just use BC transformation
# correctHeteroskedasticity <- function(series, years){
#   lmMod <- lm(series~years)
#   isHet <- lmtest::bptest(lmMod)
#   if(isHet$p.value <= 0.05){
#     series_trans <- BCTransform(series)
#     return(series_trans)
#   }else{
#     return(series)
#   }
# }
# try_correctHeteroskedasticity <- function(series, years){
#   tryCatch(
#     {correctHeteroskedasticity(series, years)},
#     error = function(e){
#       return(series)
#     }
#   )
# }



## Loop 1 -- loop through OROs
TE_results <- data.frame()
TE_results_summary <- data.frame()
for(oro in oros){
  # oro <- "MRE-Ocean" # for testing

  # Data processing
  # Fillin missing values with a 0 -- so that same years are represented across time series
  # correct for heteroskedasiticy
```

```r
oroDat <- allComponentDat_model %>%
  filter(oro_type == oro) %>%
  complete(component, year=year_lim[1]:year_lim[2],
           fill = list(y=0), explicit= FALSE) %>%
  group_by(component) %>%
  mutate(y_trans = try_BCTransform(y)) %>%
  ungroup()
# ggplot(oroDat, aes(x=year, y=y_trans))+geom_line()+facet_wrap(vars(component), scales="free_y")


edgesResults <- data.frame(
  oro_type = rep(oro, nrow(edges)),
  NodeX = edges[,1],
  NodeY = edges[,2],
  TE_XCauseY = rep(NA, nrow(edges)),
  TE_pval = rep(NA, nrow(edges)),
  TE_ratio = rep(NA, nrow(edges))
)

# Loop through the edges and test for causality
for(e in 1:nrow(edges)){

  TE_out <- tryCatch(
    {VLTransferEntropy(
    Y= oroDat$y_trans[oroDat$component == edges[e,1]],
    X= oroDat$y_trans[oroDat$component == edges[e,2]],
    maxLag = 5,
    VLflag=TRUE,nboot=100, alpha = 0.05)},
    error = function(e){
      return(NULL)
    }
  )
  if(is.null(TE_out)){
    edgesResults$TE_pval[e] <- NA
    edgesResults$TE_XCauseY[e] <- NA
    edgesResults$TE_ratio[e] <- NA
  }else{
    edgesResults$TE_pval[e] <- TE_out$pval
    edgesResults$TE_XCauseY[e] <- TE_out$XgCsY_trns # TS$X causes TS$Y TRUE/FALSE
    edgesResults$TE_ratio[e] <- TE_out$TEratio
  }


}

# summarize results to most causal direction if a two-way effect found
# ie. When TRUE for both directions, resolve by picking the direction with the highest TE ratio
edgesResultsSummary <- data.frame()
uniqueEdges <- edges[!apply(edges, 1, is.unsorted), ]
for(ue in 1:nrow(uniqueEdges)){

  incl = vector(length = nrow(edgesResults))
  for(i in 1:nrow(edgesResults)){
      tmp <- c(edgesResults[i,c("NodeX","NodeY")]) %in% uniqueEdges[ue,]
      incl[i] <- sum(tmp)==2
    }
    tmpDat <- edgesResults[incl,] %>%
      filter(TE_XCauseY) %>%
      arrange(desc(TE_ratio)) %>%
      slice_head(n=1)

    edgesResultsSummary <- edgesResultsSummary %>%
      bind_rows(tmpDat)
  }


  # Bind results
  TE_results <- TE_results %>%
    bind_rows(edgesResults)
  TE_results_summary <- TE_results_summary %>%
    bind_rows(edgesResultsSummary)

}

# clean
rm(oroDat, nodes, edges, TE_out, edgesResults)



## Check data
# View(TE_results)
# View(TE_results_summary)
# TE_results %>% filter(!is.na(TE_XCauseY)) %>%filter(TE_XCauseY == TRUE)  %>% View


# save(TE_results, TE_results_summary, file = here::here("data/derived-data/TE_results.RData"))
```

```r
load(here::here("data/derived-data/TE_results.RData")) # TE_results, TE_results_summary
```

Step 2: Create Meta Network

Aggregate Causal edges (identified from transfer entropy) Across oro_types

Aggregate the TE results across oro_types to build a meta-causal graph:

For each unique edge, compute:

- N significant: how many oro_types had significant causality. This indicates agreement/consistency
- Mean TE_ratio: average causal strength across significant cases.

Visualize the aggregated meta-network where:

- Edge width = mean TE ratio.
- Edge transparency = consistency (e.g. normalized N significant).
- Node color = node cluster (identified from hierarchical clustering)

```r
meta_links <- TE_results %>%
  filter(TE_pval < 0.05,TE_XCauseY==TRUE, !is.na(TE_ratio), TE_ratio > 0) %>%
  group_by(NodeX, NodeY) %>%
  summarise(
    n_sig = n(),
    avg_ratio = mean(TE_ratio, na.rm = TRUE),
    .groups = 'drop'
  )%>%
  mutate(
    relative_nsig = n_sig / max(n_sig)
  ) %>%
  arrange(desc(relative_nsig), desc(avg_ratio))


# Make meta graph
meta_graph <- graph_from_data_frame(meta_links, directed = TRUE)

# Calculate clusters
clusters <- cluster_walktrap(meta_graph)


# Visualize meta network
# arrow width ~ average TE_ratio
# arrow colour ~ agreement
# node colour ~ cluster
E(meta_graph)$weight <- meta_links$avg_ratio
E(meta_graph)$color <- scales::alpha("black", meta_links$relative_nsig)
E(meta_graph)$width <- log1p(E(meta_graph)$weight) * 1.5
V(meta_graph)$color <- clusters$membership



## Plot
# svg(file = here::here("figures/supplemental/meta_TE_network_diagraph.svg"),width = 8, height = 8)
# plot(meta_graph)
# dev.off()
```

Quantify Node Importance and Influence using network centrality metrics:

- Out-degree / in-degree: raw count of causal effects given/received, ie driver vs responder.
- Eigenvector : influence in the broader network.
- Betweenness: mediators .

```r
# causal influencers - legislation, policy
influencers <- degree(meta_graph, mode = "out") %>%
  sort(decreasing = T) %>%
  as.data.frame() %>%
  tibble::rownames_to_column("component") %>%
  mutate(metric = "Out-degree (N effects given)")

# causal responders - deployment - 8
responders <- degree(meta_graph, mode = "in") %>%
  sort(decreasing = T) %>%
  as.data.frame() %>%
  tibble::rownames_to_column("component") %>%
  mutate(metric = "In-degree (N effects received)")


# general influence - legislation                    policy           public interest
eigenvalues <- sort(eigen_centrality(meta_graph)$vector, decreasing =TRUE)  %>%
  as.data.frame() %>%
  tibble::rownames_to_column("component") %>%
  mutate(metric = "Eigenvector centrality")

# mediators - deployment - 14
mediators <- sort(betweenness(meta_graph), decreasing=TRUE)%>%
  as.data.frame() %>%
  tibble::rownames_to_column("component") %>%
  mutate(metric = "Betweenness (N shortest paths)")


## format results and save

metaNetCentralityMetrics <- rbind(
  influencers,
  responders,
  eigenvalues,
  mediators
)%>%
  mutate(
    component = factor(
      component,
      levels = componentAES$level,
      labels = componentAES$label
    )
  )
colnames(metaNetCentralityMetrics)[2] <- "Value"


# ## Save results
# metaNetCentralityMetricsSummary <- metaNetCentralityMetrics %>%
#   group_by(metric) %>%
#   slice_max(n=3,order_by = Value)
#
# write.csv(
#   metaNetCentralityMetricsSummary,
#   file = here::here("outputs/metaNetworkMetrics_summary.csv")
# )
#
# write.csv(
#   metaNetCentralityMetrics,
#   file = here::here("outputs/metaNetworkMetrics_all.csv")
# )
```

```
## Plot of the network
plot_ggraph_meta_network <- function(g, plotTitle=NULL, titleSize = 10, nodeTextSize = 3) {
  g_tidy <- as_tbl_graph(g)


  gg <- ggraph(g_tidy, layout = "circle") +  # You can try "kk" or "circle" or "fr" for spacing
    geom_edge_link(aes(width = 1/log1p(weight), alpha = relative_nsig),
                   arrow = arrow(length = unit(5, 'mm')),
                   color = "black", show.legend = FALSE) +
    # geom_node_point(size = 6, aes(color = as.factor(color))) +
    geom_node_point(size = 7, aes(color = name))+
    # scale_color_brewer(type = "qual", guide="none")+
    scale_color_manual(values = componentAES$colour,
                       breaks = componentAES$level,
                       guide="none")+
    geom_node_label(aes(label = componentAES$label[match(name, componentAES$level)]), repel = TRUE, #nu
                   size = nodeTextSize, alpha=0.8
                   ) +
    theme_void() +
    theme(
      plot.margin = unit(c(5,5,5,5), units = "mm")
      )
  if(!is.null(plotTitle)){
    gg <- gg + labs(title = paste(plotTitle)) +
      theme(
        plot.title = element_text(size = titleSize, hjust = 0.5)
      )
  }
  return(gg)
}

metaNetwork_ggp <- plot_ggraph_meta_network(
  meta_graph,nodeTextSize = 3.5
)


## Plot of a heatmap of each metric value
networkMetrics_ggp <- ggplot(
  data = metaNetCentralityMetrics %>%
    group_by(metric) %>%
    mutate(
      Value_scaled = scales::rescale(Value, to = c(0,1))
    ),
  aes(x=component, y=metric)
)+
  geom_tile(aes(fill = Value_scaled))+
  scale_y_discrete(
    labels = function(x) stringr::str_wrap(x, width=15)
  )+
  scale_fill_distiller(palette = "Blues", direction=1)+
  labs(
    x = "",
    y="Network metric",
    fill = "Metric\n(scaled)"
  )+
```

```r
  theme_minimal(base_size = 12)+
  theme(
    axis.text.x = element_text(angle = 45, hjust=1)
    # legend.position = "bottom"
  )
networkMetrics_ggp


# ## combine plots
# ggarrange(plotlist = list(metaNetwork_ggp, networkMetrics_ggp),
#           labels = paste0(letters[1:2], ")"),
#           label.x = 0, vjust =1.2, font.label = list(size = 13)
# )  %>%
#   ggpubr::ggexport(filename = here::here("figures/supplemental/metaNetwork_Graph_and_MetricHeatmap.pd
#                    nrow = 1,
#            width = 10, height= 4.5)
```

Step 3: Build a Qualitative Network Model

Aim: Drawing upon the causal links identified using transfer entropy (ie. the meta-network in step 2), what does it imply about the drivers of ORO action? What levers can be pulled to affect change?

Analysis:

1. Pool the transfer entropy results. If an edge is present in an ORO-level network, keep the edge in the meta network. Optional: if the level of agreement is below a certain threshold, the edge is uncertain and simulated with a probability weight (for now this is set to FALSE)
2. From the pooled network, create a qualitative network model
3. For each node (except action), simulate a press perturbation and record the response of action.

Kieran questions

- Does this approach seem sensible?
- Do you notice any errors in the analysis/interpretation?

```r
nSims <- 5000

source(here::here("R/impact.barplot.myMod"))
source(here::here("R/extend.vector"))
source(here::here("R/my.QPress.R"))

addTaskCallback(function(...) {set.seed(123);TRUE})

## Constant variables
# Do we want to simulate with probability weights? T/F
weightSimulation = FALSE
# the threshold of agreement below which the edge is sample with a probability weight of 0.5
uncertainThreshold = 1
oroTypes <- unique(allComponentDat_model$oro_type)
qp_results_df <- data.frame()
```

```r
## 1. Pool the transfer entropy results.
sig_te_edges <- TE_results %>%
  filter(TE_XCauseY == TRUE,!is.na(TE_ratio)) %>%
  group_by(NodeX, NodeY) %>%
  summarise(
    n_sig = n(),
    .groups = 'drop'
  )%>%
  mutate(
    Probability = n_sig / max(n_sig),
    Group = ifelse(uncertainThreshold <= Probability,0,1)
  ) %>%
  arrange(desc(Probability)) %>%
  select(From=NodeX, To=NodeY, Group, Probability) %>%
  mutate(
    Type = "P"
  )

# Format the data frame for Q Press
sig_nodes <- as.character(unique(c(sig_te_edges$From, sig_te_edges$To)))

sig_te_edges <- sig_te_edges %>%
  mutate(
    From = factor(From, levels = sig_nodes),
    To = factor(To, levels=sig_nodes),
    Type = factor(Type, levels = c("N","P","U","Z")),
    # Group = factor(Group)
  )
sig_te_edges$Pair <- 1:nrow(sig_te_edges)

allEdgesLabels <- edge.labels(sig_te_edges)

# Mod <- retain.groups(sig_te_edges, groups=0) # all certain interactions
Mod <- enforce.limitation(sig_te_edges%>% select(-Probability)) # enforce self-limiting effect
# s<- community.sampler(Mod)
# s$select(0.5)
# ModSim <- system.simulate(100, Mod,sampler = s) # simulate
# print(impact.table(ModSim))


## 2. Build Qualitative network model
# Optional: Simulate with weights?
if(weightSimulation){
  uncertainSampleProbsDf <- sig_te_edges %>%
  filter(Group == 1) %>%
  select(Pair, Probability) %>%
  distinct(Pair,.keep_all = TRUE)

  ModSim <- my.system.simulate(nSims, Mod,required.groups = c(0), uncertainSampleProbsDf=uncertainSampl
}else{
  ModSim <- system.simulate(100, Mod)
}
```

```r
## 3. For each node (except action), simulate a press perturbation and record the response of action.

# Loop over interventions on each node → observe effect on "deployment"
all_nodes <- sig_nodes[sig_nodes != "deployment"]

for (node in all_nodes) {
  press <- c(1)
  names(press) <- node

  # Set evidence and query deployment
  temp <- impact.barplot.myMod(ModSim,
      perturb = press,
      plot=FALSE, percentage = TRUE
  )
  temp <- as.data.frame(t(temp["deployment",]))
  temp$cluster <- "Meta-network"
  temp$Press_perturb <- node

  qp_results_df <- qp_results_df %>% bind_rows(temp)

}



## Results
# Plot results
uniquePP <- unique(qp_results_df$Press_perturb)

qnm_ggp <- ggplot(qp_results_df %>% filter(Positive>0),
       aes(x=Press_perturb, y=stringr::str_wrap(cluster, width=10), fill = Positive))+
  geom_tile()+
  scale_fill_stepsn(name = stringr::str_wrap("Positive effect on action (% sims)", width=25), n.breaks=
  # scale_fill_steps2(name = stringr::str_wrap("Positive effect on action (% sims)", width=25), n.break
  scale_x_discrete(
    breaks = componentAES$level[
      componentAES$level %in% uniquePP
    ],
    labels = componentAES$label[
      componentAES$level %in% uniquePP
    ]
  )+
  labs(y="Meta-network",x="Press perturbation")+
  theme_bw()+
  theme(
    legend.position = "bottom",
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.text.x=element_text(angle=45, hjust=1)
  )

qnm_ggp
```

```
# ## Save plot
# ggsave(
#   here::here("figures/supplemental/qp_metaNetwork_heatmap.pdf"),
#   plot = qnm_ggp,
#   width=5, height=4, units="in"
# )
```

```
# plotList = list(metaNetwork_ggp+theme(plot.margin = unit(rep(0.1, 4), "cm")),
#                  networkMetrics_ggp+theme(plot.margin = unit(rep(0.1, 4), "cm")),
#                  qnm_ggp+theme(plot.margin = unit(rep(0.1, 4), "cm")))
#
# combined_patchwork <- wrap_plots(plotList, ncol = 2, heights = c(1,0.75)) + plot_annotation(tag_level
#
# ggsave(here::here("figures/main/metaNetwork_networkMetricHeatmap_QNMHeatmap.pdf"),
#         combined_patchwork, width = 10, height = 9)
```

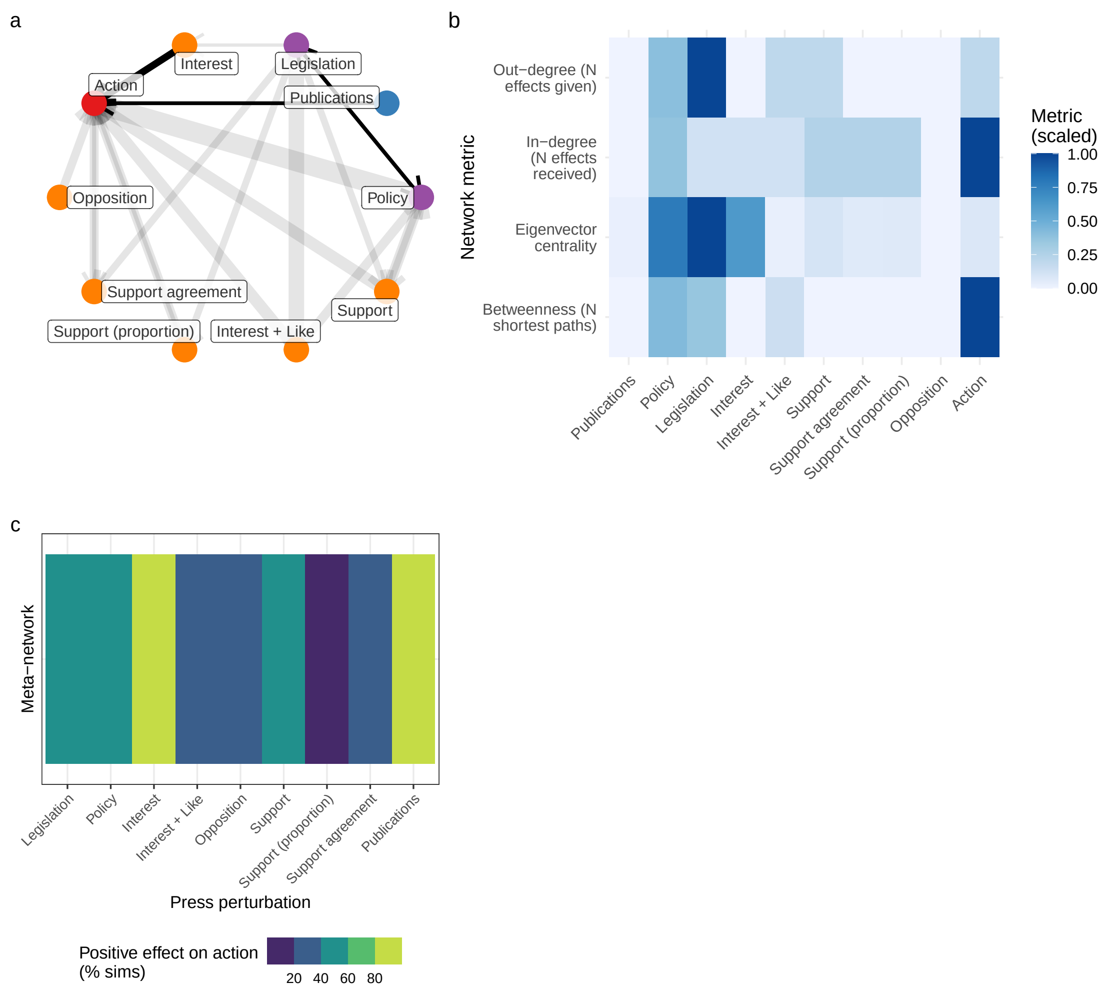

Figure 7: Network analysis figures a) meta netowrk, b) network metrics, c) quanitative network model response of action to press perturbation