

Observed climate change impacts on plant communities in Mediterranean-type ecosystems: A systematic review

Thematic analysis

Hana Petersen

Preamble

The R workflow below describes the processes followed to format the raw extracted data and confidence index scores, and related data analysis. Comments (#) have been provided where appropriate and relevant.

Trends in responses to climate change-related disturbances

The code below is for formatting and summarising the relevant thematic data from the full list of included publications, and visualising trends in responses across regions.

```
# load libraries
library(tidyverse)
library(readxl)

# read data
if(file.exists("themes_expanded.csv")) {
  df_full <- read_csv(paste0(wd,"themes_expanded.csv"))
} else {
  df_full <- read_excel(paste0(wd,"Dataset_S1_Themes.xlsx"),
                       sheet = "themes_expanded")
}

df_full <- df_full %>%
  mutate(region = str_replace(region, "South and/or Southwestern Australia",
                              "SW Australia")) %>%
```

```

mutate(region = str_replace(region, "Cape Floristic Region, South Africa",
                             "Cape Floristic Region, RSA"))

# create long format data for responses
df_long <- df_full %>%
  pivot_longer(
    cols = starts_with("var_"),
    names_to = "response",
    names_prefix = "var_",
    values_to = "trend",
    values_drop_na = TRUE
  )

# create long format for trends
df_trends <- df_long %>%
  rename(disturbance = disturbance_type) %>%
  # select(c(1, 2, 6, 8, 19, 20)) %>%
  select(c("citation", "title", "region",
           "disturbance", "response", "trend")) %>%
  separate_rows(trend, sep = ", ") %>%
  separate_rows(disturbance, sep = ", ") %>%
  mutate(disturbance = str_replace(disturbance, "Pathogen/Insect outbreak",
                                   "Pathogen/Insect"))

# grouping
df_trends <- df_trends %>%
  filter(!is.na(trend)) %>%
  group_by(region, disturbance, response, trend) %>%
  summarise(count = n()) %>%
  filter(!trend == "N") %>% # remove trends assigned as "N"
  filter(!disturbance == "Anthropogenic") # remove all cases of "Anthropogenic"

# reorder responses
df_trends$trend <- factor(df_trends$trend,
                         levels = c("D", "NL", "I"))

# plot bar charts faceted by region and response
p1 <-
ggplot(df_trends, aes(x = disturbance, y = count, fill = trend)) +
  geom_bar(position = "fill", stat = "identity", colour = "black") +
  labs(x = "Disturbance type", y = "Frequency of reporting",
       fill = "Trend") +

```

```

coord_flip() +
facet_grid(response ~ region) +
scale_fill_manual(values = palette_trends,
                  labels = c("Decreasing", "Non-linear", "Increasing")) +
scale_y_continuous(breaks = scales::breaks_pretty()) +
scale_x_discrete(limits = rev) +
theme_bw(18) +
theme_mte

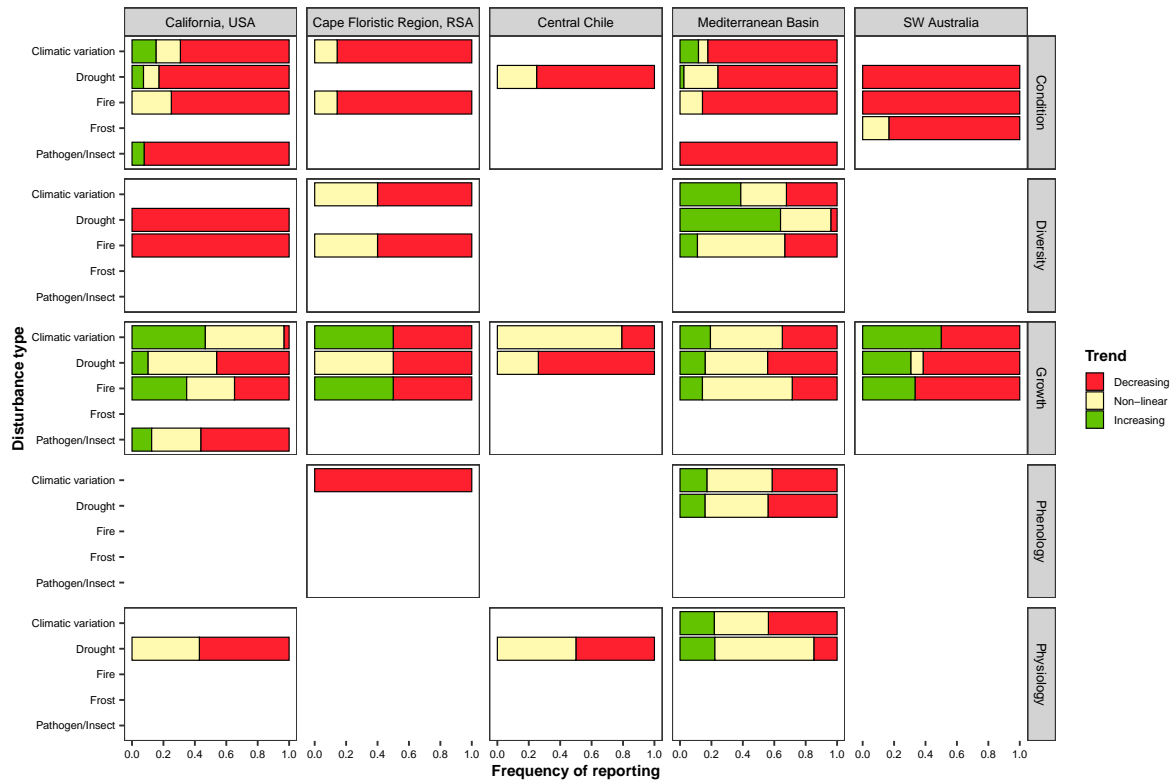
# create function that removes unwanted facets
remove_facets <- function(plot, layout) {
  layout <- strsplit(layout, split = '\\n')[[1]]
  layout <- lapply(layout, trimws)
  layout <- matrix(unlist(sapply(layout, strsplit, "")),
                  nrow = length(layout), byrow = T)
  layout <- which(layout == "#", arr.ind = TRUE)
  prm <- apply(layout, 1, \ (x) {
    c(glue::glue("panel-{x[1]}-{x[2]}"),
      glue::glue("strip-t-{x[2]}-{x[1]}"))
  })

  g <- ggplot2::ggplotGrob(plot)
  rm_grobs <- g$layout$name %in% prm
  g$grobs[rm_grobs] <- NULL
  g$layout <- g$layout[!rm_grobs, ]
  ggpubr::as_ggplot(g)
}

# remove unwanted facets
a <- c("aaaaa
      aa#a#
      aaaaa
      #a#a#
      a#aa#")

remove_facets(p1, a)

```



Confidence index scores

Distribution of confidence index scores

The code below is for formatting and summarising the confidence index data to visualise the distribution of C-index scores.

```
# load libraries
library(tidyverse)
library(readxl)

# read data
if(file.exists("ci_scores.csv")) {
  df_ci <- read_csv(paste0(wd,"ci_scores.csv"))
} else {
  df_ci <- read_excel(paste0(wd,"Dataset_S1_Themes.xlsx"),
    sheet = "ci_scores")
}
```

```

df_ci <- df_ci %>%
  select(c(1, 7, 10:16)) %>%
  group_by(region) %>%
  pivot_longer(!c(`identifier`, `region`),
               names_to = "score",
               values_to = "value") %>%
  group_by(score, region, value) %>%
  mutate(region = str_replace(region, "CFR, South Africa",
                              "Cape Floristic Region, RSA")) %>%

  summarise(count = n())

# compute means and medians
mean_median_ci <- df_ci %>%
  group_by(score) %>%
  summarise(mean = mean(value), median = median(value))

# modifications to allow ease of plotting
df_ci$overall <- "Overall score"
df_ci$sd <- "Data (SD)"
df_ci$se <- "Expectation (SE)"
df_ci$sq <- "Statistics (SQ)"
df_ci$sr <- "Reporting (SR)"

# plot individually first
p.overall <- df_ci[136:176,] %>%
  ggplot(aes(x = value, y = count, fill = region)) +
  geom_col(colour = "black") +
  geom_vline(xintercept = 13.2, linewidth = 0.5) +
  geom_vline(xintercept = 13, linewidth = 0.5, linetype = "dashed") +
  xlim(0, 21) +
  ylim(0, 17) +
  labs(x = "Confidence index score", y = "Number of publications") +
  scale_fill_manual(values = palette_mte) +
  theme_classic() +
  theme_mte +
  theme(legend.position = "none",
        plot.title = element_text(colour = "black", size = 8,
                                   hjust = 0.5, vjust = 0.5))

p.overall <- p.overall + facet_grid(. ~ overall) +
  theme(strip.background = element_rect(fill = "lightgrey"),
        strip.text = element_text(size = 12))

```

```

p.se <- df_ci[50:71,] %>%
  ggplot(aes(x = value, y = count, fill = region)) +
  geom_col(colour = "black") +
  geom_vline(xintercept = 3.7, linewidth = 0.5) +
  geom_vline(xintercept = 4, linewidth = 0.5, linetype = "dashed") +
  xlim(-0.5, 7) +
  ylim(0, 40) +
  scale_fill_manual(values = palette_mte) +
  theme_classic() +
  theme1 +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        legend.position = "none",
        plot.title = element_text(colour = "black", size = 8,
                                   hjust = 0.5, vjust = 0.5))

p.se <- p.se + facet_grid(. ~ se) +
  theme(strip.background = element_rect(fill = "lightgrey"),
        strip.text = element_text(size = 12))

p.sd <- df_ci[27:49,] %>%
  ggplot(aes(x = value, y = count, fill = region)) +
  geom_col(colour = "black") +
  geom_vline(xintercept = 4, linewidth = 0.5) +
  geom_vline(xintercept = 4, linewidth = 0.5, linetype = "dashed") +
  xlim(0, 8) +
  ylim(0, 40) +
  scale_fill_manual(values = palette_mte) +
  theme_classic() +
  theme1 +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        legend.position = "none",
        plot.title = element_text(colour = "black", size = 8,
                                   hjust = 0.5, vjust = 0.5))

p.sd <- p.sd + facet_grid(. ~ sd) +
  theme(strip.background = element_rect(fill = "lightgrey"),
        strip.text = element_text(size = 12))

p.sq <- df_ci[72:94,] %>%
  ggplot(aes(x = value, y = count, fill = region)) +

```

```

geom_col(colour = "black") +
geom_vline(xintercept = 4.6, linewidth = 0.5) +
geom_vline(xintercept = 5, linewidth = 0.5, linetype = "dashed") +
xlim(0, 8) +
ylim(0, 60) +
scale_fill_manual(values = palette_mte) +
theme_classic() +
theme1 +
theme(axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      legend.position = "none",
      plot.title = element_text(colour = "black", size = 8,
                                hjust = 0.5, vjust = 0.5))

p.sq <- p.sq + facet_grid(. ~ sq) +
  theme(strip.background = element_rect(fill = "lightgrey"),
        strip.text = element_text(size = 12))

p.sr <- df_ci[95:104,] %>%
  ggplot(aes(x = value, y = count, fill = region)) +
  geom_col(colour = "black") +
  geom_vline(xintercept = 0.5, linewidth = 0.5) +
  geom_vline(xintercept = 0.5, linewidth = 0.5, linetype = "dashed") +
  xlim(-0.5, 1.5) +
  ylim(0, 100) +
  labs(fill = "Regions") + # legend appears next to this plot
  scale_fill_manual(values = palette_mte) +
  theme_classic() +
  theme1 +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        legend.text = element_text(vjust = 0.5),
        legend.title = element_text(face = "bold"),
        plot.title = element_text(colour = "black", size = 8,
                                    hjust = 0.5, vjust = 0.5))

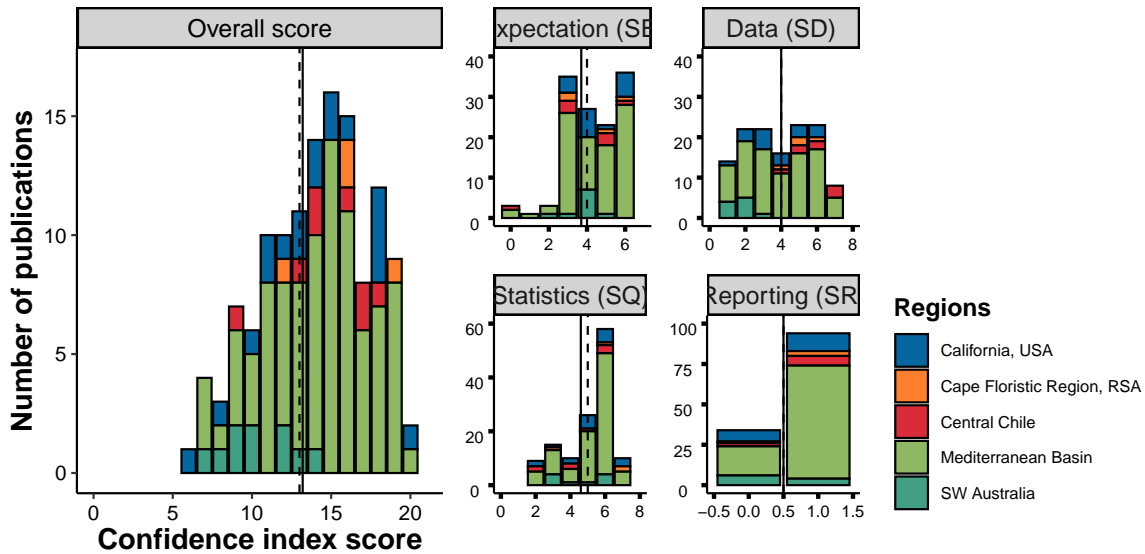
p.sr <- p.sr + facet_grid(. ~ sr) +
  theme(strip.background = element_rect(fill = "lightgrey"),
        strip.text = element_text(size = 12))

# plot together using {patchwork}
library(patchwork)

```

```
nested <- (p.overall|(p.se|p.sd)/(p.sq|p.sr))

# view multi-panel figure
nested
```



Confidence index score analysis

The code below is for formatting and summarising the relevant extracted data from the full list of included publications, and visualising the C-index scores across regions and biological responses.

```
# load libraries
library(tidyverse)
library(readxl)

# read data
if(file.exists("themes_expanded.csv")) {
  df_full <- read_csv(paste0(wd,"themes_expanded.csv"))
} else {
  df_full <- read_excel(paste0(wd,"Dataset_S1_Themes.xlsx"),
                        sheet = "themes_expanded")
}
```



```

# shorten long names
df_full <- df_full %>%
  mutate(region = str_replace(region, "South and/or Southwestern Australia",
                                "SW Australia")) %>%
  mutate(region = str_replace(region, "Cape Floristic Region, South Africa",
                                "Cape Floristic Region, RSA"))

# calculate mean and SD ci understanding and detection scores by region
df_U <- df_full %>%
  group_by(region) %>%
  summarise(mean_U = mean(ci_U), sd_U = sd(ci_U))

df_D <- df_full %>%
  group_by(region) %>%
  summarise(mean_D = mean(ci_D), sd_D = sd(ci_D))

ud_regions <- bind_cols(df_D, df_U[2:3]) %>%
  add_column(response = "Overall", .before = 1)

# create long format data for responses
df_long <- df_full %>%
  pivot_longer(
    cols = starts_with("var_"),
    names_to = "response",
    names_prefix = "var_",
    values_to = "trend",
    values_drop_na = TRUE
  )

# calculate mean and SD ci understanding and detection scores by response
df_U_resp <- df_long %>%
  group_by(response, region) %>%
  summarise(mean_U = mean(ci_U), sd_U = sd(ci_U))

df_D_resp <- df_long %>%
  group_by(response, region) %>%
  summarise(mean_D = mean(ci_D), sd_D = sd(ci_D))

ud_responses <- bind_cols(df_D_resp, df_U_resp[3:4])

# combine responses and regions datasets
ud_responses <- bind_rows(ud_responses, ud_regions)

```

```

ud_responses$response <- factor(ud_responses$response,
                                levels = c("Overall", "Condition",
                                             "Diversity", "Growth",
                                             "Phenology", "Physiology"))

# summarise counts per response category per region
response_region_counts_df <- df_full %>%
  select(citation, region, response_category) %>%
  separate_longer_delim(response_category, delim = ", ") %>%
  unique() %>%
  group_by(response_category, region) %>%
  unique() %>%
  summarise(count = n())

# calculate publication counts per region
# create new category called "Overall"
pub_region_counts_df <- df_full %>%
  select(doi, region) %>%
  group_by(region) %>%
  unique() %>%
  summarise(count = n()) %>%
  add_column(response_category = "Overall", .before = 1)

pub_counts_df <- bind_rows(response_region_counts_df, pub_region_counts_df)

# combine with ci scores df
ud_responses <- bind_cols(ud_responses,
                          count_pubs = pub_counts_df$count)

# plot ci understanding and detection scores per region
ggplot(ud_responses, aes(x = mean_D, y = mean_U)) +
  geom_abline(intercept = 0, slope = 1,
              size = 1, linetype = "dashed", colour = "grey") +
  geom_point(size = 10, colour = "grey", alpha = 0.5) +
  geom_point(aes(x = mean_D, y = mean_U,
                 size = count_pubs, colour = region)) +
  scale_size(guide = "none") +
  guides(color = guide_legend(override.aes = list(size = 4))) +
  geom_errorbar(aes(ymin = mean_U-sd_U, ymax = mean_U+sd_U,
                   colour = region), width = 0.2) +
  geom_errorbar(aes(xmin = mean_D-sd_D, xmax = mean_D+sd_D,
                   colour = region), width = 0.2) +

```

```

xlim(0, 10.5) +
ylim(0, 10.5) +
labs(x = "Detection", y = "Understanding",
     colour = "Regions") +
facet_wrap(vars(response), nrow = 2) +
scale_colour_manual(values = palette_mte) +
theme_classic() +
theme_mte

```

