Linking Summer Precipitation to Pollination Network Robustness in Sky Island Ecosystems

Alejandro Santillana

2024-12-09

Introduction

Plant-pollinator networks represent the intricate dependencies between species that sustain ecosystem services, like pollination. However, increasing anthropogenic pressures, including habitat fragmentation and climate change, destabilize these interactions (Urban 2015). This project investigates coextinction cascades—a phenomenon where primary species loss triggers secondary extinctions within ecological networks (Memmott et al. 2004; Kaiser-Bunbury et al. 2010). Quantifying community robustness—defined as the ability of networks to resist species loss—is critical for understanding ecosystem stability (Dunne et al. 2002). To quantify robustness, I simulated extinction cascades in bipartite networks ordered by species abundance.

Precipitation is critical in arid ecosystems like the Sky Islands, influencing floral resource availability and pollinator activity patterns, and is increasingly important as climate change impacts this region (Brusca et al. 2013). Summer precipitation totals provide a meaningful measure of climatic stress during peak pollination periods.

My project focuses on the following questions:

How robust are plant-pollinator networks in the Madrean Sky Islands to species loss ordered by abundance? How do environmental factors, specifically precipitation, relate to network robustness? Can climatic data provide predictive insights into the vulnerability of specific sites?

These questions explore new dimensions of bee conservation and ecological stability, leveraging both the Sky Islands and climate datasets. This analysis can guide targeted conservation interventions and improve understanding of ecological resilience.

Data

Datasets Used:

Sky Islands dataset (2012, 2017, 2018, 2021, 2022): Bipartite plant-pollinator matrices. PRISM Climate Data: Monthly precipitation rasters providing high-resolution climatic context. Site Metadata: Geospatial and sampling information necessary for aligning networks with climate data.

Data Cleaning:

Plant-Pollinator Networks: Ensure matrices are complete, visualize network structures, and filter incomplete surveys. Climate Data: Extract precipitation values for each site and season, ensuring alignment with sampling years and locations.

Import Precipitation Rasters from PRISM

The following script processes monthly precipitation rasters for summer months (June–September) across multiple years. These are summed to provide seasonal totals for analysis.

```
# Import monthly precipitation data for each field season
precip_2011_06_raster <- rast("PRISM_summer_precip/2011/PRISM_ppt_stable_4kmM3_201106_bil.bil")
precip_2011_07_raster <- rast("PRISM_summer_precip/2011/PRISM_ppt_stable_4kmM3_201107_bil.bil")
precip_2011_08_raster <- rast("PRISM_summer_precip/2011/PRISM_ppt_stable_4kmM3_201108_bil.bil")
precip_2011_09_raster <- rast("PRISM_summer_precip/2011/PRISM_ppt_stable_4kmM3_201109_bil.bil")

precip_2016_06_raster <- rast("PRISM_summer_precip/2016/PRISM_ppt_stable_4kmM3_201606_bil.bil")
precip_2016_07_raster <- rast("PRISM_summer_precip/2016/PRISM_ppt_stable_4kmM3_201607_bil.bil")
precip_2016_08_raster <- rast("PRISM_summer_precip/2016/PRISM_ppt_stable_4kmM3_201608_bil.bil")
precip_2016_09_raster <- rast("PRISM_summer_precip/2016/PRISM_ppt_stable_4kmM3_201609_bil.bil")

precip_2017_06_raster <- rast("PRISM_summer_precip/2017/PRISM_ppt_stable_4kmM3_201706_bil.bil")
precip_2017_07_raster <- rast("PRISM_summer_precip/2017/PRISM_ppt_stable_4kmM3_201707_bil.bil")
precip_2017_08_raster <- rast("PRISM_summer_precip/2017/PRISM_ppt_stable_4kmM3_201708_bil.bil")
precip_2017_09_raster <- rast("PRISM_summer_precip/2017/PRISM_ppt_stable_4kmM3_201709_bil.bil")

precip_2020_06_raster <- rast("PRISM_summer_precip/2020/PRISM_ppt_stable_4kmM3_202006_bil.bil")
precip_2020_07_raster <- rast("PRISM_summer_precip/2020/PRISM_ppt_stable_4kmM3_202007_bil.bil")
precip_2020_08_raster <- rast("PRISM_summer_precip/2020/PRISM_ppt_stable_4kmM3_202008_bil.bil")
precip_2020_09_raster <- rast("PRISM_summer_precip/2020/PRISM_ppt_stable_4kmM3_202009_bil.bil")

precip_2021_06_raster <- rast("PRISM_summer_precip/2021/PRISM_ppt_stable_4kmM3_202106_bil.bil")
precip_2021_07_raster <- rast("PRISM_summer_precip/2021/PRISM_ppt_stable_4kmM3_202107_bil.bil")
precip_2021_08_raster <- rast("PRISM_summer_precip/2021/PRISM_ppt_stable_4kmM3_202108_bil.bil")
precip_2021_09_raster <- rast("PRISM_summer_precip/2021/PRISM_ppt_stable_4kmM3_202109_bil.bil")

# Define the years and file paths
years <- c("2011", "2016", "2017", "2020", "2021")
months <- c("06", "07", "08", "09")
base_path <- "PRISM_summer_precip"

# Initialize a list to store the summed rasters for each year
summer_rasters <- list()

# Loop through each year
for (year in years) {

  # Initialize an empty list to hold monthly rasters
  monthly_rasters <- list()

  # Loop through each month
  for (month in months) {
    # Construct the file path
    file_path <- file.path(base_path, year, paste0("PRISM_ppt_stable_4kmM3_", year, month, "_bi
l.bil"))

    # Load the raster and append to the list
    monthly_rasters[[month]] <- rast(file_path)
  }

  # Stack and sum the monthly rasters
  stacked_rasters <- rast(monthly_rasters)
```

```
  summer_raster <- sum(stacked_rasters, na.rm = TRUE)

  # Save the raster to the list
  summer_rasters[[year]] <- summer_raster


  # Print a message for progress tracking
  print(paste("Processed year:", year))
}
```

```
## [1] "Processed year: 2011"
## [1] "Processed year: 2016"
## [1] "Processed year: 2017"
## [1] "Processed year: 2020"
## [1] "Processed year: 2021"
```

```
precip_2011_raster <- summer_rasters[["2011"]]
precip_2016_raster <- summer_rasters[["2016"]]
precip_2017_raster <- summer_rasters[["2017"]]
precip_2020_raster <- summer_rasters[["2020"]]
precip_2021_raster <- summer_rasters[["2021"]]

# Identify variables with the pattern "precip_****_**_raster"
monthly_rasters <- ls(pattern = "^precip_\\d{4}_\\d{2}_raster$")
rm(list = monthly_rasters)
```

Import and Transform Spatial Data

The following script imports a shapefile for study sites, ensuring consistency in coordinate reference systems (CRS) with PRISM rasters.

```
sites_shp <- vect("Sites/sites.shp")
print(crs(sites_shp))
```

```
## [1] "GEOGCRS[\"unknown\",\n    DATUM[\"World Geodetic System 1984\",\n        ELLIPSOID[\"WGS
84\",6378137,298.257223563,\n            LENGTHUNIT[\"metre\",1]],\n        ID[\"EPSG\",632
6]],\n    PRIMEM[\"Greenwich\",0,\n        ANGLEUNIT[\"Degree\",0.0174532925199433]],\n    CS[el
lipsoidal,2],\n        AXIS[\"longitude\",east,\n            ORDER[1],\n            ANGLEUNIT
[\"Degree\",0.0174532925199433]],\n        AXIS[\"latitude\",north,\n            ORDER[2],\n
ANGLEUNIT[\"Degree\",0.0174532925199433]]]"
```

```
print(crs(precip_2011_raster))
```

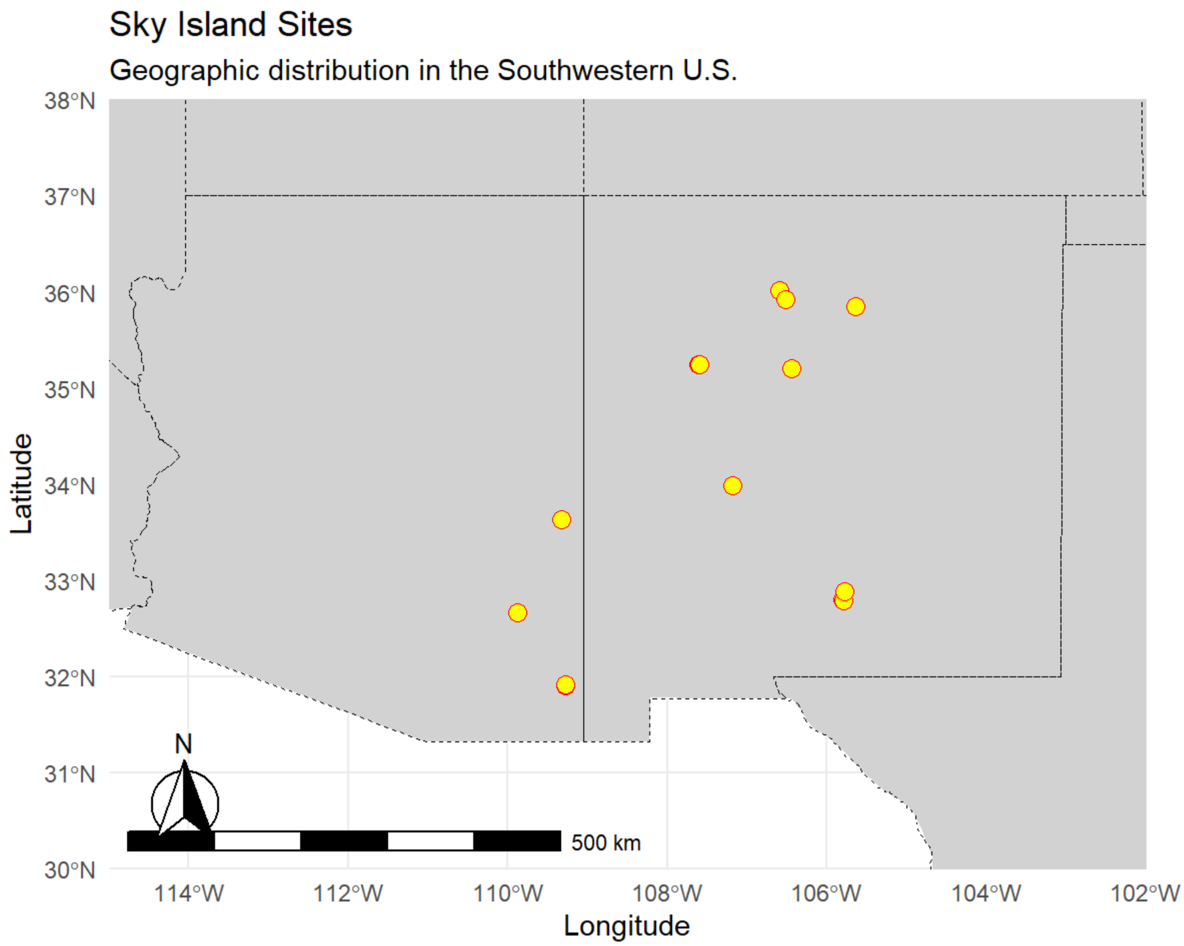
```
## [1] "GEOGCRS[\"NAD83\",\n    DATUM[\"North American Datum 1983\",\n        ELLIPSOID[\"GRS 19
80\",6378137,298.257222101,\n            LENGTHUNIT[\"metre\",1]],\n        ID[\"EPSG\",626
9]],\n    PRIMEM[\"Greenwich\",0,\n        ANGLEUNIT[\"Degree\",0.0174532925199433]],\n    CS[el
lipsoidal,2],\n        AXIS[\"longitude\",east,\n            ORDER[1],\n            ANGLEUNIT
[\"Degree\",0.0174532925199433]],\n        AXIS[\"latitude\",north,\n            ORDER[2],\n
ANGLEUNIT[\"Degree\",0.0174532925199433]]]"
```

```
sites_shp <- project(sites_shp, crs(precip_2011_raster))
```

Geospatial Visualization

Sky Islands Map The map contextualizes the geographic positioning of the Sky Islands, a biodiversity hotspot spanning the southwestern U.S. and northern Mexico. Sites sampled are in the states of New Mexico and Arizona.

```
# Load the country or region boundary
world <- ne_countries(scale = "medium", returnclass = "sf")

# (Optional) Subset for specific countries if relevant
world <- world[world$name %in% c("United States of America"), ]

# Get U.S. state boundaries
states <- ne_states(country = "United States of America", returnclass = "sf")


sites <- st_as_sf(sites_shp)

ggplot() +
  # Add basemap
  geom_sf(data = world, fill = "lightgray", color = "white") +

  # Add your sites
  geom_sf(data = sites, color = "red", size = 3, shape = 21, fill = "yellow") +

  # Add state boundaries
  geom_sf(data = states, fill = NA, color = "black", linetype = "dashed") +

  # Add optional scalebar and north arrow
  annotation_scale(location = "bl", width_hint = 0.5) +
  annotation_north_arrow(location = "bl", which_north = "true",
                         style = north_arrow_fancy_orienteering) +

  # Define map extent for the southwestern U.S.
  coord_sf(xlim = c(-115, -102), ylim = c(30, 38), expand = FALSE) +

  # Customize the appearance
  theme_minimal() +
  labs(title = "Sky Island Sites",
       subtitle = "Geographic distribution in the Southwestern U.S.",
       x = "Longitude", y = "Latitude")
```

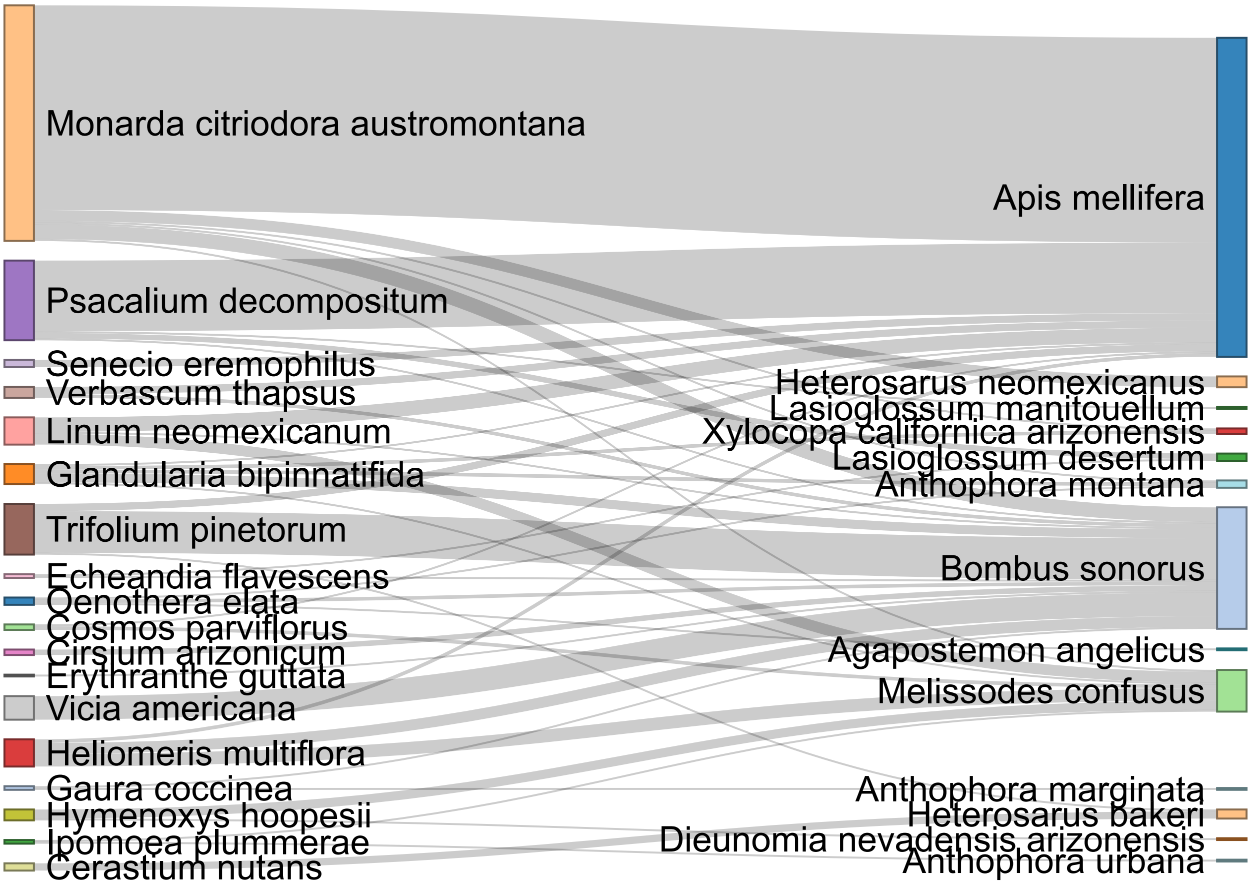


###

Importing and Visualizing Networks

A Sankey plot is a type of flow diagram that visualizes the connections between two sets of elements, often used to show weighted interactions. In this context, the Sankey plot for the CH.2012 network represents a plant-pollinator interaction network, illustrating how different pollinators interacted with various plant species in the Chiricahua Mountains site in the survey year 2012.

```
load("../skyIslands/data/networks/Yr_PlantPollinator_Bees.RData") #yearly networks for bees

# Select the specific network "CH.2012"
example_network <- nets[["CH.2012"]]
network_name <- "CH.2012" # Set the name explicitly for reference

# Transform the network data for the Sankey plot
data_long <- as.data.frame(example_network) %>%
  rownames_to_column %>%
  gather(key = 'key', value = 'value', -rowname) %>%
  filter(value > 0)

# Rename columns for clarity
colnames(data_long) <- c("source", "target", "value")
data_long$target <- paste(data_long$target, " ", sep = "")

# Create nodes data frame for the plot
nodes <- data.frame(name = c(as.character(data_long$source),
                             as.character(data_long$target)) %>% unique())

# Reformat the data for networkD3
data_long$IDsource <- match(data_long$source, nodes$name) - 1
data_long$IDtarget <- match(data_long$target, nodes$name) - 1

# Generate the Sankey plot
sankey_plot <- sankeyNetwork(Links = data_long, Nodes = nodes,
                             Source = "IDsource", Target = "IDtarget",
                             Value = "value", NodeID = "name",
                             fontSize = 18)

# Display the Sankey plot
sankey_plot
```



Methods: Data Wrangling and Summarizing

Extract Precipitation Data for Sites

This code extracts precipitation data from raster files for specified years and calculates average precipitation for each study site.

```r
years <- c(2011, 2016, 2017, 2020, 2021)
results_list <- list()

for (year in years) {
  # Load the raster for the year
  raster <- rast(paste0("PRISM_precip/PRISM_ppt_stable_4kmM3_", year, "_bil.bil"))

  # Extract precipitation values for subsites
  precip_values <- extract(raster, sites_shp)

  # Combine extracted values with site shapefile data
  subsite_data <- as.data.frame(sites_shp) # Convert shapefile to a data frame
  subsite_data$Precipitation <- precip_values[, 2] # Add extracted values
  subsite_data$Year <- year # Add the year

  # Group by Site and Year, then calculate mean precipitation
  site_data <- subsite_data %>%
    group_by(Site, Year) %>%
    summarize(Mean_Precipitation = mean(Precipitation, na.rm = TRUE), .groups = "drop")

  # Store the yearly averaged data in the results list
  results_list[[as.character(year)]] <- site_data
}

# Combine all yearly site-level results into a single data frame
all_precip_data <- do.call(rbind, results_list)
```

Performing Extinction Simulations and Calculating Robustness

This code simulates species extinctions within plant-pollinator networks, calculates their robustness (a measure of how well the network resists species loss), and organizes the results into a data frame for analysis.

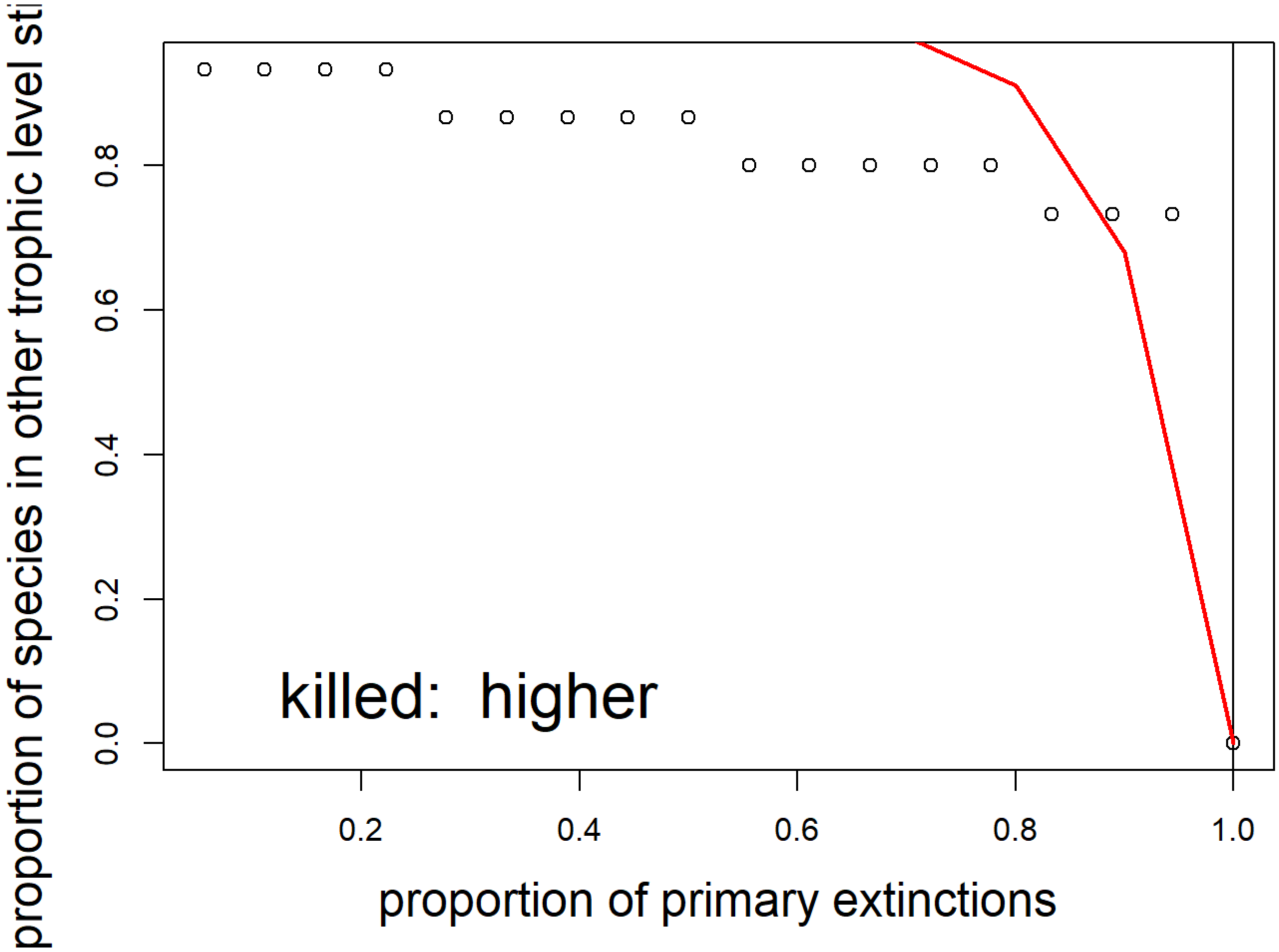
```
# Create an empty data frame to store the results
robustness_results <- data.frame(Network = character(), Site = character(), Year = integer(), Ro
bustness = numeric(), stringsAsFactors = FALSE)

# Loop through each network in the list `nets`
for (net_name in names(nets)) {
  # Extract the network
  network <- nets[[net_name]]

  # Run extinction simulation
  network_ext <- second.extinct(network, participant = "higher", method = "abun", nrep = 50, det
ails = FALSE)

  # Calculate the robustness (area under second.extinct curve)
  network_robustness <- robustness(network_ext)

  # Extract Site and Year from the network name
  site_year <- unlist(strsplit(net_name, "\\.")) # Split by "."
  site <- site_year[1]
  year <- as.integer(site_year[2])

  # Append the results to the data frame
  robustness_results <- rbind(robustness_results, data.frame(Network = net_name, Site = site, Ye
ar = year, Robustness = network_robustness))
}

# Print the resulting data frame
print(robustness_results)
```

```
##    Network Site Year Robustness
## 1  CH.2012   CH 2012 0.82838375
## 2  CH.2017   CH 2017 0.86367505
## 3  CH.2018   CH 2018 0.80074807
## 4  CH.2021   CH 2021 0.70642125
## 5  CH.2022   CH 2022 0.79160167
## 6  HM.2017   HM 2017 0.90249176
## 7  HM.2018   HM 2018 0.82549589
## 8  HM.2021   HM 2021 0.89709237
## 9  HM.2022   HM 2022 0.85794005
## 10 JC.2012   JC 2012 0.90548181
## 11 JC.2017   JC 2017 0.84459121
## 12 JC.2018   JC 2018 0.93540366
## 13 JC.2021   JC 2021 0.85274760
## 14 MM.2012   MM 2012 0.83455713
## 15 MM.2017   MM 2017 0.91582870
## 16 MM.2018   MM 2018 0.88201583
## 17 MM.2021   MM 2021 0.72426397
## 18 MM.2022   MM 2022 0.82447204
## 19 PL.2012   PL 2012 0.84464551
## 20 PL.2017   PL 2017 0.05758102
## 21 PL.2018   PL 2018 0.71238599
## 22 PL.2021   PL 2021 0.76976470
## 23 PL.2022   PL 2022 0.60036377
## 24 RP.2021   RP 2021 0.78573483
## 25 RP.2022   RP 2022 0.59027778
## 26 SC.2012   SC 2012 0.89479808
## 27 SC.2017   SC 2017 0.94399397
## 28 SC.2018   SC 2018 0.87775926
## 29 SC.2021   SC 2021 0.94882352
## 30 SC.2022   SC 2022 0.91059212
## 31 SM.2017   SM 2017 0.73974938
## 32 SM.2018   SM 2018 0.87202940
## 33 SM.2021   SM 2021 0.88491565
## 34 SM.2022   SM 2022 0.85691211
## 35 SS.2017   SS 2017 0.76565425
## 36 UK.2017   UK 2017 0.40740741
## 37 VC.2017   VC 2017 0.68805610
```

Extinction Curve Plot

This plot demonstrates an extinction cascade sequence derived from the use of the second.extinct function. Robustness of the networks is defined as the area under the curve.

```
trial <- second.extinct(nets[["CH.2018"]], participant = "higher", method = "abun", nrep = 1000,
details = FALSE)
trial_slope <- slope.bipartite(trial, plot.it = TRUE, ylab="y")
```



Merge Precipitation Data with Robustness Results

This code merges the precipitation data (all_precip_data) with the robustness results (robustness_results) and performs some data cleaning steps. The precipitation data (all_precip_data) represents the year in which the precipitation was measured, but the robustness data (robustness_results) represents the following year (because the year of the network data in robustness_results corresponds to the year after the precipitation data). So, adding 1 to the Year of the precipitation data aligns it with the corresponding year in the robustness data, ensuring the years match when the data is merged. The final line filters out rows from final_data since the sites (“SS”, “UK”, or “VC”) were surveyed less than the others, and are problematic for the analysis.

```
# Add one year to the precipitation year to align with robustness data
all_precip_data <- all_precip_data %>%
  mutate(Year = Year + 1)

final_data <- merge(robustness_results, all_precip_data, by = c("Site", "Year"))
final_data <- final_data %>%
  filter(!Site %in% c("SS", "UK", "VC"))
```

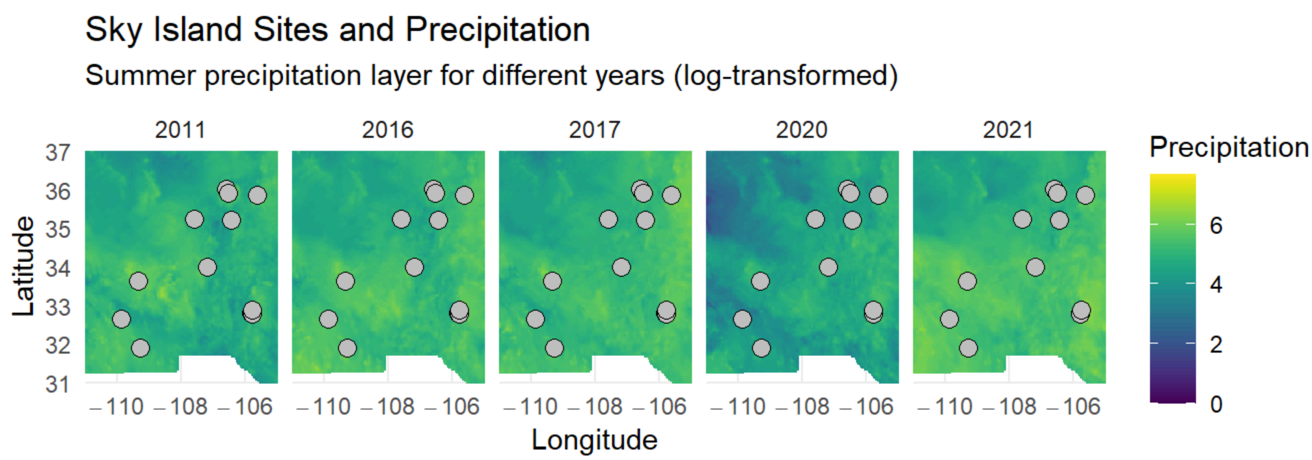
Precipitation map

This code produces a faceted map that displays the spatial distribution of summer precipitation for the years 2011, 2016, 2017, 2020, and 2021. Each map in the facet corresponds to one year. The precipitation values are log-transformed to handle potential zero values and make the differences more visually interpretable. The map also includes the locations of study sites overlaid on the precipitation data. The color scale represents the log-transformed precipitation values, helping to highlight variations in precipitation across the region and over time.

```
# Define a list of the rasters
rasters <- list(
  precip_2011 = precip_2011_raster,
  precip_2016 = precip_2016_raster,
  precip_2017 = precip_2017_raster,
  precip_2020 = precip_2020_raster,
  precip_2021 = precip_2021_raster
)

# Define a list of corresponding years for labeling
years <- c("2011", "2016", "2017", "2020", "2021")

# Calculate global minimum and maximum precipitation values across all rasters
min_precip <- min(sapply(rasters, function(r) min(values(r), na.rm = TRUE)))
max_precip <- max(sapply(rasters, function(r) max(values(r), na.rm = TRUE)))

# Create an empty list to store all the data for ggplot
plot_data <- list()

# Loop through each raster and prepare data for ggplot
for (i in 1:length(rasters)) {

  # Get the current raster and year
  current_raster <- rasters[[i]]
  current_year <- years[i]

  # Convert the current raster to a data.frame for ggplot
  precip_df <- as.data.frame(current_raster, xy = TRUE, na.rm = TRUE)  # Remove NAs

  # Rename the third column to "precip"
  colnames(precip_df)[3] <- "precip"

  # Apply log transformation to precipitation values (log(x + 1) to avoid log(0))
  precip_df$precip_log <- log(precip_df$precip + 1)

  # Add the year to the data for faceting
  precip_df$year <- current_year

  # Store the data in the list
  plot_data[[i]] <- precip_df
}

# Combine all the data into a single data frame
plot_data_df <- do.call(rbind, plot_data)

# Convert SpatVector to sf object
sites <- st_as_sf(sites_shp)

# Create the plot with facet wrap
plot <- ggplot(plot_data_df) +
  # Add the precipitation raster layer
  geom_raster(aes(x = x, y = y, fill = precip_log)) +
```

```
  # Add a color scale for the precipitation with fixed limits for consistency
  scale_fill_viridis(option = "D", name = "Precipitation",
                     direction = 1) +
# limits = c(log(min_precip + 1), log(max_precip + 1))) +

  # Add the site locations
  geom_sf(data = sites, color = "black", size = 3, shape = 21, fill = "grey") +

  # Customize map extent (adjust as needed)
  coord_sf(xlim = c(-111, -105), ylim = c(31, 37), expand = FALSE) +

  # Customize the appearance
  theme_minimal() +
  labs(
    title = "Sky Island Sites and Precipitation",
    subtitle = "Summer precipitation layer for different years (log-transformed)",
    x = "Longitude",
    y = "Latitude"
  ) +
  # Round axis labels
  scale_x_continuous(breaks = seq(-110, -106, by = 2),
                     labels = function(x) round(x, 2)) +  # Adjust longitude rounding
  scale_y_continuous(breaks = seq(31, 37, by = 1),
                     labels = function(y) round(y, 2)) +  # Adjust latitude rounding

  # Facet wrap by year
  facet_wrap(~ year, nrow=1)

# Print the plot
print(plot)
```

```
## Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted
## ℹ Consider using `geom_tile()` instead.
## Raster pixels are placed at uneven horizontal intervals and will be shifted
## ℹ Consider using `geom_tile()` instead.
```



Results

Fit a Linear Mixed-effects Model

The model investigates the relationship between network robustness (the response variable) and two fixed effects:

Mean Precipitation (in mm, representing moisture availability from the previous year). Year (temporal factor, representing the year of data collection). It also accounts for site-specific variability as a random effect (intercept for “Site”), which allows the model to control for differences between sites while focusing on the broader relationships.

```
model <- lmer(Robustness ~ Mean_Precipitation + Year + (1 | Site), data = final_data)

summary(model)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Robustness ~ Mean_Precipitation + Year + (1 | Site)
##    Data: final_data
##
## REML criterion at convergence: -7.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.3788 -0.1838  0.1264  0.3931  1.1541
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Site     (Intercept) 0.005868 0.0766
##  Residual             0.017920 0.1339
## Number of obs: 34, groups:  Site, 8
##
## Fixed effects:
##                      Estimate Std. Error t value
## (Intercept)        13.3774552 14.6660734   0.912
## Mean_Precipitation -0.0002758  0.0001547  -1.784
## Year               -0.0061477  0.0072603  -0.847
##
## Correlation of Fixed Effects:
##             (Intr) Mn_Prc
## Men_Prcpttn -0.122
## Year        -1.000  0.116
```

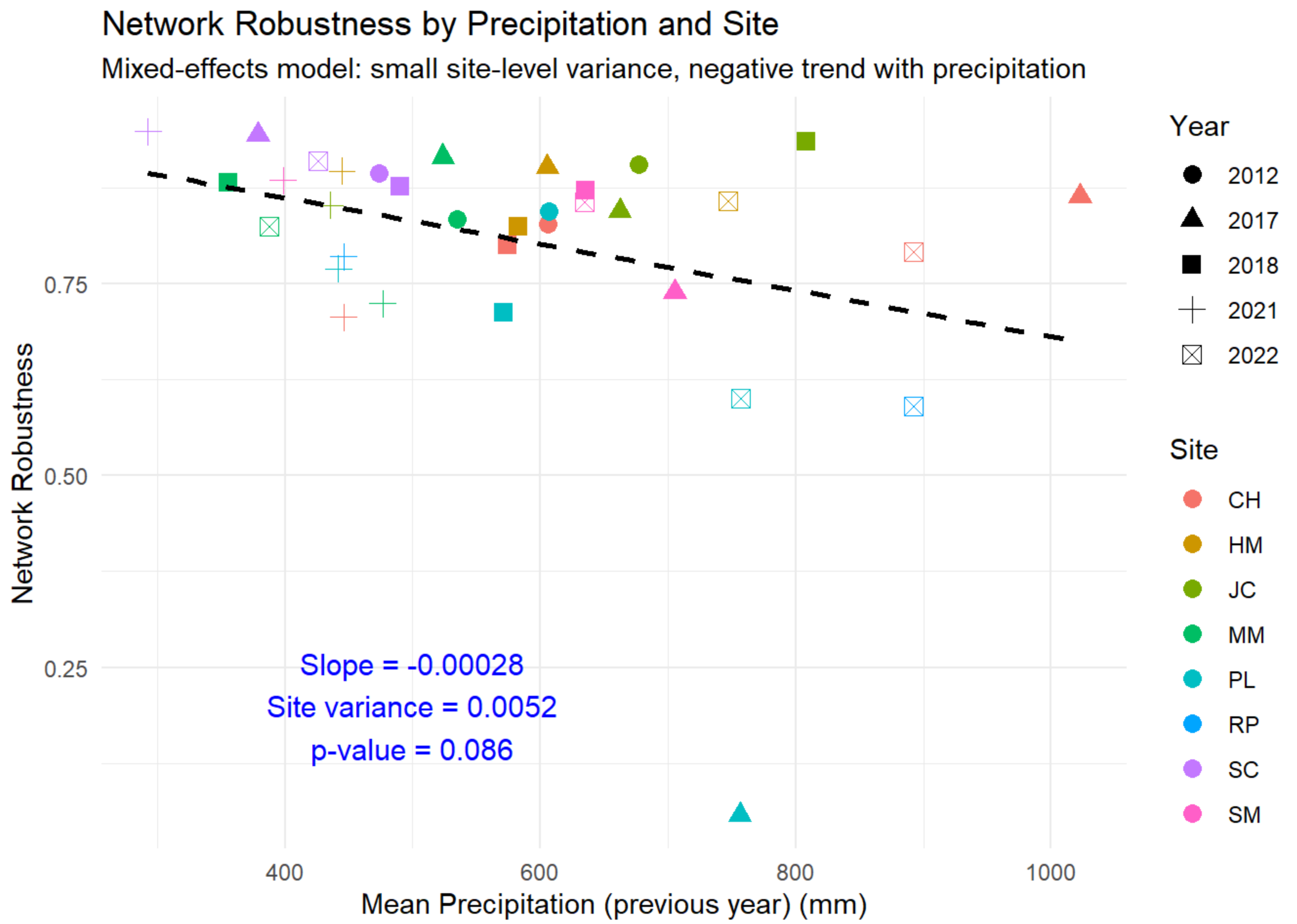
Robustness by Site: Scatterplot with regression line to display the relationship between precipitation and network robustness.

To visualize the relationship between mean precipitation and network robustness across different sites and years, this code generates a scatterplot with a regression line where: Mean precipitation is plotted on the x-axis and network robustness on the y-axis. Points are colored by site and shaped by year. A regression line is added to show the relationship between precipitation and robustness, and the slope and p-value of this relationship are annotated on the plot. The plot provides insight into the relationship between precipitation and network robustness while accounting for differences across sites and years.

```
ggplot(final_data, aes(x = Mean_Precipitation, y = Robustness, color = Site, shape = as.factor(Y
ear))) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se = FALSE, aes(group = 1), color = "black", linetype = "dashed") +
  labs(
    title = "Network Robustness by Precipitation and Site",
    subtitle = "Mixed-effects model: small site-level variance, negative trend with precipitatio
n",
    x = "Mean Precipitation (previous year) (mm)",
    y = "Network Robustness",
    color = "Site",
    shape = "Year"
  ) +
  annotate("text", x = 500, y = 0.2, label = "Slope = -0.00028\nSite variance = 0.0052\np-value
= 0.086", size = 4, color = "blue") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: The following aesthetics were dropped during statistical transformation: shape
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



Discussion and Conclusion

From the model results, mean precipitation was negatively correlated with network robustness, though this effect was not statistically significant ($p > 0.05$). The fixed effect estimate for mean precipitation (-0.00026) indicates a slight negative trend, which contrasts with my initial hypothesis. The non-significance of this relationship, with a t-value of -1.670, suggests that the influence of precipitation on robustness may be weaker or more context-dependent than expected. This discrepancy could result from the small sample size, site-specific interactions, or unmeasured confounding factors that moderate the relationship between precipitation and robustness.

Year did not exhibit a significant effect on robustness, as reflected in its low estimate (-0.0063) and t-value (-0.867). This lack of temporal variation suggests that, within the studied time frame, other environmental or ecological drivers, rather than temporal shifts alone, are more critical in influencing network stability. Additionally, the relatively small variance attributed to the site-specific random effect (variance = 0.0058) implies that while site-to-site variability exists, it accounts for only a modest portion of the overall variation in robustness. The larger residual variance (0.0181) underscores the potential role of other unmeasured factors in shaping network robustness.

Despite these complexities, the abundance-driven extinction order provided insight into the structural dynamics of these networks. Common species—those with the highest interaction frequencies—play a disproportionately critical role in maintaining network stability. The loss of these abundant species resulted in pronounced declines in robustness, emphasizing their foundational role in supporting the overall resilience of ecological networks. This finding aligns with theoretical and empirical studies, highlighting that abundant species with numerous interactions often serve as keystone elements in ecological systems.

In conclusion, while the influence of precipitation on network robustness appears to be complex and less pronounced than I initially hypothesized, the findings underscore the need for future research to disentangle these dynamics further. Expanding the model to include a larger number of sites, more years, and additional environmental variables, such as temperature, habitat diversity, and resource phenology, could illuminate the multifaceted drivers of network stability.

Limitations and Next Steps: Abundance as the sole extinction order may overlook functional and fitness roles in network dynamics. Climate data resolution could be improved to capture microclimatic variations.

References: Brusca, R. C., Wiens, J. F., Meyer, W. M., Eble, J., Franklin, K., Overpeck, J. T. & Moore, W., 2013 Dramatic response to climate change in the southwest: Robert whittaker’s 1963 arizona mountain plant transect revisited. Ecology and evolution 3, 3307-3319.

Dunne, J. A., Williams, R. J. & Martinez, N. D. Network structure and biodiversity loss in food webs: robustness increases with connectance. Ecology Letters 5, 558–567 (2002).

Kaiser-Bunbury, C. N., Muff, S., Memmott, J., Müller, C. B. & Caflisch, A. The robustness of pollination networks to the loss of species and interactions: a quantitative approach incorporating pollinator behaviour. Ecology Letters 13, 442–452 (2010).

Memmott, J., Waser, N. M. & Price, M. V. Tolerance of Pollination Networks to Species Extinctions. Proceedings: Biological Sciences 271, 2605–2611 (2004).

Urban, M. C. Accelerating extinction risk from climate change. Science 348, 571–573 (2015).