**Insect responses to climate change**

**GitHub:** <https://github.com/trenchproject/Johnson_Insect_Responses>

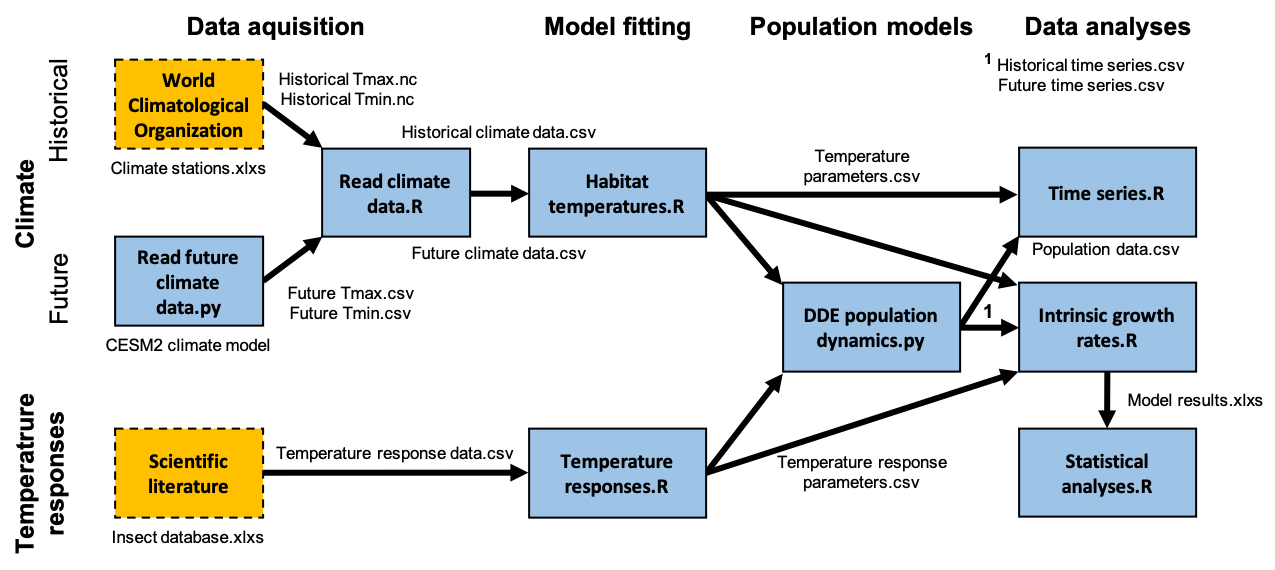
**Google Drive:** <https://drive.google.com/drive/u/0/folders/1ggsdJLmqfHytMuV4Iu6Muo-GWDQ7WYs4>

**Overview:** This document provides an overview of my project investigating how thermal performance curves inform predictions of insect population dynamics under global climate change. It includes brief descriptions of the GitHub files, an overview of the computational pipeline for the project, and provides protocols for obtaining and analyzing climate data.

**GitHub Files:**

* Climate station data.xlxs
  + Spreadsheet providing information on the climate stations from which the historical climate data is obtained.
  + To be updated when new climate data is obtained from the World Meteorological Organization (http://climexp.knmi.nl/start.cgi)
* DDE population dynamics.py
  + Python script for predicting insect population dynamics during historical or future time periods.
  + Must input the time period (“Historical” or “Future”), whether the analysis is for estimating species’ low density growth rate (LDG = True) or for predicting species’ population dynamics (LDG = False).
  + Can also specify the simulation time by changing max\_years on line 49
  + The script reads “Temperature response parameters.csv” and “Temperature parameters.csv” and outputs a dataset containing the species’ population dynamics as “<time period> time series <species name> <location>.csv” or “<time period> time series LDG <species name> <location>.csv
* Future climate data <location>.csv
  + CSV file containing future temperature data at <location>
  + See “Read climate data.R” for more information
* Future time series <species> <location>.csv
  + CSV file containing future population densities for <species> at <location>
  + See “DDE population dynamics.py” for more information
* Habitat temperatures.R
  + R script for estimating temperature parameters based on fits to historical and future temperature data
  + Must input location; the script estimates temperature parameters (manually entered into “Temperature parameters.csv”) and plots data and model fits
* Historical climate data <location>.csv
  + CSV file containing historical temperature data at <location>
  + See “Read climate data.R” for more information
* Historical time series <species> <location>.csv
  + CSV file containing historical population dynamics for <species> at <location>
  + See “DDE population dynamics.py” for more information
* Insect database.xlxs
  + Database containing species name, location, resource, latitude, longitude, reference, whether it has been used in the analysis, habitat type, subfamily, and the number of laboratory temperatures at which it was studied
  + To be updated when new species are added
* Intrinsic growth rates.R
  + R script for calculating the intrinsic growth rate, *r*, for the historical and future time periods, from both the thermal performance curve and the DDE model
  + Must enter the location and species name
  + Intrinsic growth rates must be manually entered into “Model results.csv”
* Johnson\_Insect\_Responses.Rproj
  + R project file
* Model results.xlxs
  + Database containing intrinsic growth rates, *r*, for each species in the historical and future time periods from the thermal performance curve and DDE model
  + See “Intrinsic growth rates.R” for more information
* ODE population dynamics.R
  + R script for predicting species’ population dynamics using an ODE model
  + Not currently used in analysis
* Population data <location>.csv
  + CSV file containing census data for species at <location>
  + Used to validate model in “Time series.R” (see script for more information)
* Project overview.docx
  + Document providing an overview of the project
* Read climate data.R
  + R script for reading netCDF files containing temperature data and outputting CSV files that can be read and analyzed using other R scripts in the project.
  + Must enter location and species name, and potentially update line 11 to the local directory on the user’s personal computer that contains the netCDF files (NOTE: for now, these files are not stored on Github)
  + The script reads data from “Historical Tmax <location>.nc”, “Historical Tmin <location>.nc”, “Future Tmax <location>.nc” and “Future Tmin <location>.nc” and outputs datasets with the day (starting at 0, with minima at time t and maxima at t+0.5) and temperature as “Historical climate data <location>.csv” and “Future climate data <location>.csv” which are then saved on GitHub
* Read future climate data.py
  + Python script for downloading daily maximum and minimum temperatures from the CESM climate model of the CMIP6 (cds.climate.copernicus.eu)
  + Must enter the latitude and longitude and location name, and may have to update the file name on lines 49 and 105 if there is an error
  + If the scripts yields the error: “WARNING Recovering from HTTP error [500 Internal Server Error]”, try logging into [cds.climate.copernicus.eu/#!/home](https://cds.climate.copernicus.eu/#!/home)
  + The script outputs datasets as “Future Tmax <location>.nc” and “Future Tmin <location>.nc” as well as “Future Tmax <location>.csv” and “Future Tmin <location>.csv”, which are analyzed using “Read climate data.R”
* README.md
  + Directory information for GitHub project
* Statistical analyses.R
  + R script for analyzing correlations between metrics (e.g., r, thermal safety margins, warming tolerance) quantified from the DDE model versus from the thermal performance curves (TPCs)
  + Plots metrics as a function of latitude and between the DDE model and TPCs
* Temperature parameters.csv
  + Database with parameters describing seasonal temperature variation at all locations
  + Fit using “Habitat temperatures.R” and used in “DDE population dynamics.py”
  + See “Habitat temperatures.R” for more information
* Temperature response data.csv
  + Database containing life history traits estimates at different temperatures in the laboratory for all species
  + See “Temperature responses.R” for more information
* Temperature response parameters.csv
  + Database containing parameters describing the temperature responses of species’ life history traits
  + Estimated using “Temperature responses.R” and used in “DDE population dynamics.py”
  + See “Temperature responses.R” for more information
* Temperature responses.R
  + R script for estimating species’ life history trait temperature responses using nls fits to data in “Temperature response data.csv”
  + Must select a species and then carefully run each module for each life history trait, which yield parameter estimates and plots the fits to laboratory data
  + Must manually enter temperature response parameters in “Temperature response parameters.csv”
* Time series.R
  + R script for plotting species’ population dynamics from “DDE population dynamics.py” and quantifying changes in population metrics and life history traits between historical and future time periods
  + Also plots model population dynamics with time series census data (where available) for model validation
  + Must select species (“Historical time series <species> <location>.csv” and “Future time series <species> <location>.csv”) and location of census data (“Population data <location>.csv”), if available

**Computational pipeline**

****

**Protocols**

Downloading historical climate data

1. Open “Read climate data.R” and navigate web browser to: <http://climexp.knmi.nl/selectdailyseries.cgi?id=someone@somewhere>
2. Under GHCN-D, click “minimum temperature”
3. Enter latitude and longitude from “Insect database.xlxs” under the second bullet point of the “Select” section
4. Under “Time, distance” section, click “Get stations”
5. Select climate station based on the following criteria
   1. First, select the climate station with the closest latitude (1st priority) and longitude (2nd priority) to those entered in step 3 (generally, this is the top result), this climate station becomes the ‘candidate station’
   2. Second, scan the other climate stations listed, if one of these stations has significantly more years of data (>15 years) or more recent data (last year >5 years more recent) than the candidate station, calculate the difference in latitude and longitude between these ‘alternative stations’ and the candidate station from step 5a
   3. The IPCC CMIP6 climate model has a resolution of 1 degree in latitude and 1.5 degrees in longitude, so if an alternative station is less than 1 degree in latitude and (ideally) less than 1.5 degrees in longitude from the candidate station, select the alternative station
   4. Generally, the station selected in step 5a will be used. If there are multiple possible stations after going through steps 5b and 5c, please email me
6. Click “get data” under selected climate station
7. Under “Time series”, view the top graph plotting temperature [Celsius] versus year. If there are significant time gaps (>5 years) between the data, view the plots for any alternative stations from step 5b and select the climate station with the fewest gaps in the record (email me if it is unclear which station should be used)
8. After confirming the station in step 7, click “raw data” (I open it in a new tab)
9. Open “Climate stations.xlsx” and under the row corresponding to the insect at the latitude entered in step 3, enter the station name (“Name”), its exact latitude (“Lat”) and longitude (“Lon”), elevation (“Elev”), station code (“Code”), WMO code (“WMO”), start date (“Start\_yr”, “Start\_mo”, and “Start\_day”) and end date (“End\_yr”, “End\_mo”, and “End\_day”)
10. Return to the “Time series” webpage, and click on “netcdf”, “Download netcdf file” and select “save file” to download the data
11. Navigate to Download folder, and rename the file “Historical Tmin <location>.nc”, where <location> is the location of the insect in “Insect database.xlsx”
12. Repeat steps 1-11 for the maximum temperature using the same climate station used in step 5 for the minimum temperature. In “raw data” (step 7), confirm that the start date for the maximum temperatures is after the start date from the minimum temperatures entered into “Climate station data.xlxs”; if not, enter this start. Rename this new netcdf file: “Historical Tmax <location>.nc”
13. At this point, there should be two downloaded files (“Historical Tmin <location>.nc” and “Historical Tmax <location>.nc”). Move these files to the project folder “Climate data”
14. See “Formatting climate data” section below

Downloading future climate data

1. Open “Read future climate data.py”
2. Enter latitude and longitude from “Insect database.xlxs” under “lat” and “lon” on lines 12 and 13, and give a location name for “loc” on line 14
3. Enter the path to the folder where the climate files are stored on lines 21 and 22
4. Highlight and run lines 1-47, this will download a netCDF file of daily maximum temperatures at the climate model grid closest to the entered latitude and longitude
   * 1. If the console prints: “WARNING Recovering from HTTP error [500 Internal Server Error]”, log into: <https://cds.climate.copernicus.eu/#!/home> and repeat step 4
     2. Sometimes the server is down and you cannot log in, then you have to just wait several hours or try again another day
5. Navigate to the folder in step 3, copy just the name of the most recently downloaded netCDF (.nc) file, and paste it within the parenthesis on line 51
6. Highlight and run lines 50-138, this will download a netCDF file of daily minimum temperatures at the climate model grid closest to the entered latitude and longitude
7. You can delete the two json and two png files associated with these climate data in the folder entered in step 3
8. At this point, there should be two netCDF files (“Future Tmin <location>.nc” and “Future Tmax <location>.nc”), which can be copied to our shared Google Drive, and two csv files (“Future Tmin <location>.csv” and “Future Tmax <location>.csv”). Move these csv files to the project folder that contains “Read climate data.R”
9. See “Formatting climate data” section below

Formatting climate data

1. In “Read climate data.R”, enter the location (“loc”), insect (“species”), and start date (“date”; format YYYY-MM-DD) from “Climate station data.xlxs” on lines 15-17
2. Select all lines and run the script if both the historical and future climate data have been downloaded OR highlight and run lines 6-17 and then either the “HISTORICAL CLIMATE DATA” section (lines 1-56) or “FUTURE CLIMATE DATA” section (lines 60-81)
3. There should now be the two csv files (“Historical climate data <location>.csv” and “Future climate data <location>.csv”). Copy these files to our shared Google Drive folder. All done!