Demonstration of the disclim Package for Implementing Climate Driven, Trait-Based Models of Disease Transmission

Michael C. Wimberly

2024-02-27

Setup

The tidyverse suite of packages is used for data manipulation and graphing, and devtools is required for the `load_all()` function. Right now `load_all()`is used to read the disclim functions from a local folder, but this should eventually be changed to install the package from GitHub and then load normally with `library()`.

```
library(tidyverse)
library(devtools)
mypath <- "C:/Users/wimb0002/OneDrive - University of Oklahoma/Work/disclim"
load_all(mypath)
```

Next, files containing the parameters are loaded. There are two sets of parameters from Miazgawicz et al. (2020) for malaria transmission by *Anopheles stephensi*, which correpond to the “estimated' and”lifetime” models. There is also a set of parameters from Mordecai et al. (2017) for dengue transmission by *Aedes aegypti*. Parameters are stored in CSV files that are currently being read in using the base R function `read.csv()`. In the future, it would be nice to have a package function that would read in the parameter files and check them for consistency.

```
mal_est_par <- read.csv(file.path(mypath, "Anstephensi_mal_est_Miazgowicz20.csv"))
den_aegypti_par <- read.csv(file.path(mypath, "Aeaegypti_dengue_Mordecai17.csv"))
```

Temperature-trait relationships

Each parameter table is a data frame with columns for 1) the variables in the model, 2) the forms of the corresponding temperature trait curves, and 3) values of the rate constant (rc) minimum temperature (tmin) and maximum temperature (tmax) parameters for each curve. The `getpar()` function retrieves parameters for a given variable from a parater table.

```
mal_est_par
```

```
##   variable      form          rc      tmin     tmax
## 1        a    briere 0.000099090 11.753360 43.94362
## 2       lf quadratic 0.050500020  1.736445 37.59197
## 3      EFD    briere 0.007258325  8.577127 39.99476
## 4      PDR    briere 0.000054155  9.064107 43.51257
## 5       bc quadratic 0.003672395 12.114979 38.13452
## 6      pEA quadratic 0.008091320 15.352860 36.95294
## 7      MDR    briere 0.000106632 13.362840 35.96712
```

```
getpar("a", "rc", mal_est_par)
```

```
## [1] 9.909e-05
```

```
getpar("a", "tmin", mal_est_par)
```

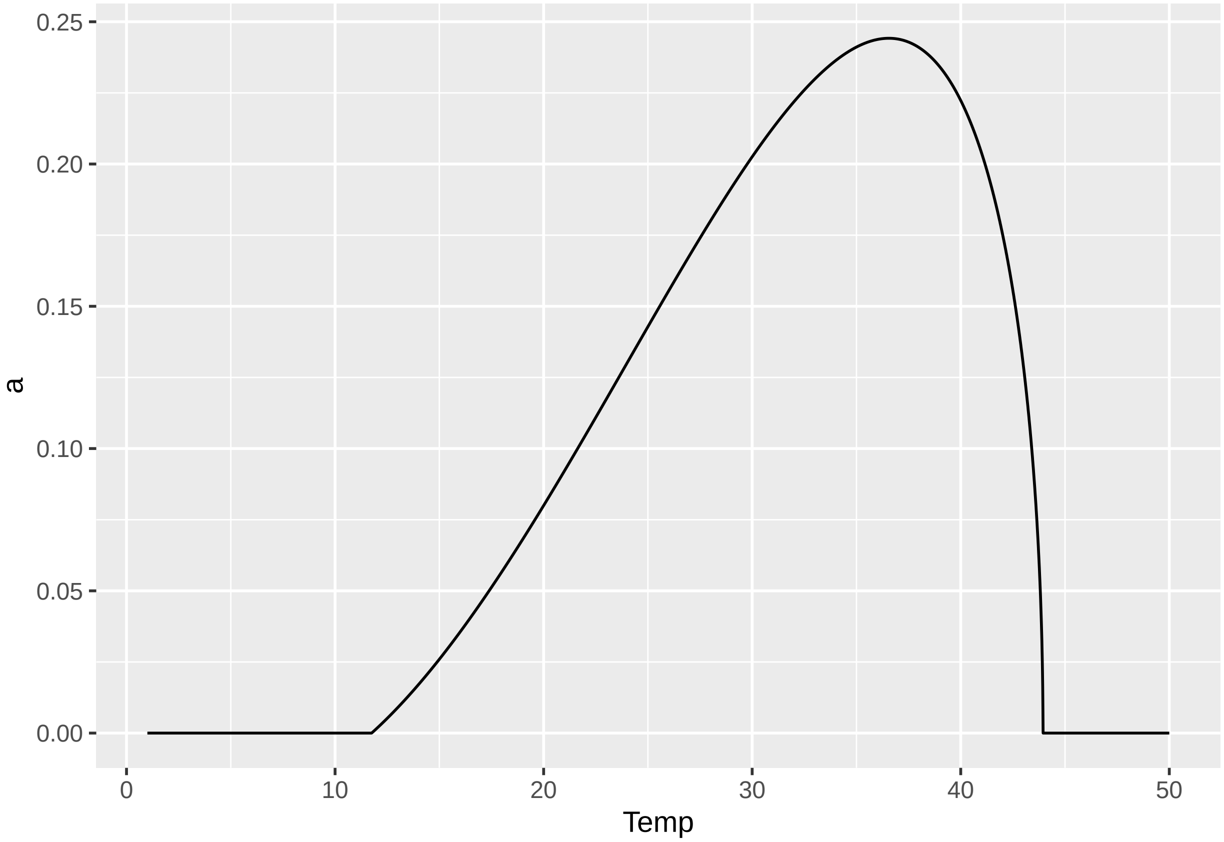
```
## [1] 11.75336
```

```
getpar("a", "tmax", mal_est_par)
```

```
## [1] 43.94362
```

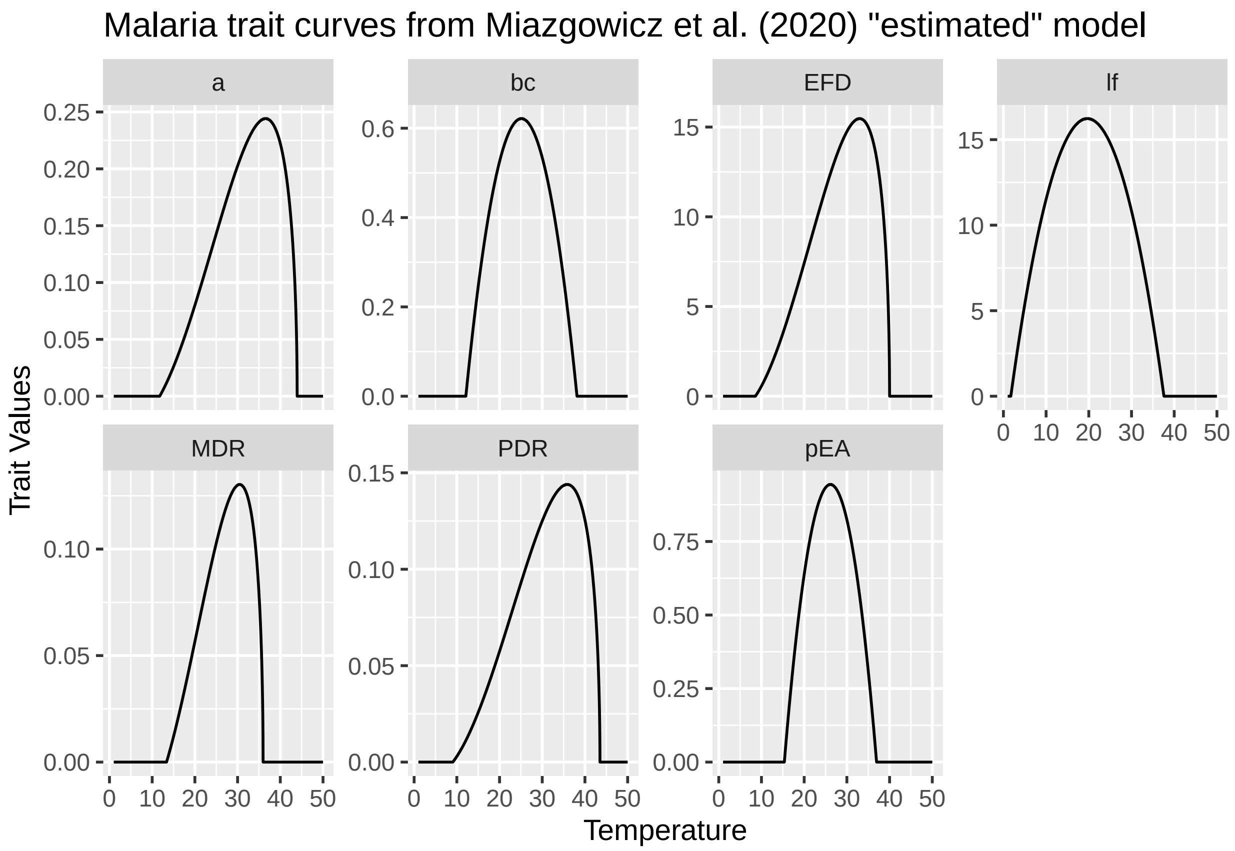
The `tempcurve()` function is used to calculate the trait values as a function of temperature for each variable. The function is vectorized, so a range of temperature can be provided.

```
Temp <- seq(1, 50, 0.01)
a <- tempcurve(Temp, "a", mal_est_par)
ggplot() +
  geom_line(aes(x = Temp, y = a))
```



This process can be repeated for all the trait variables in a particular model.

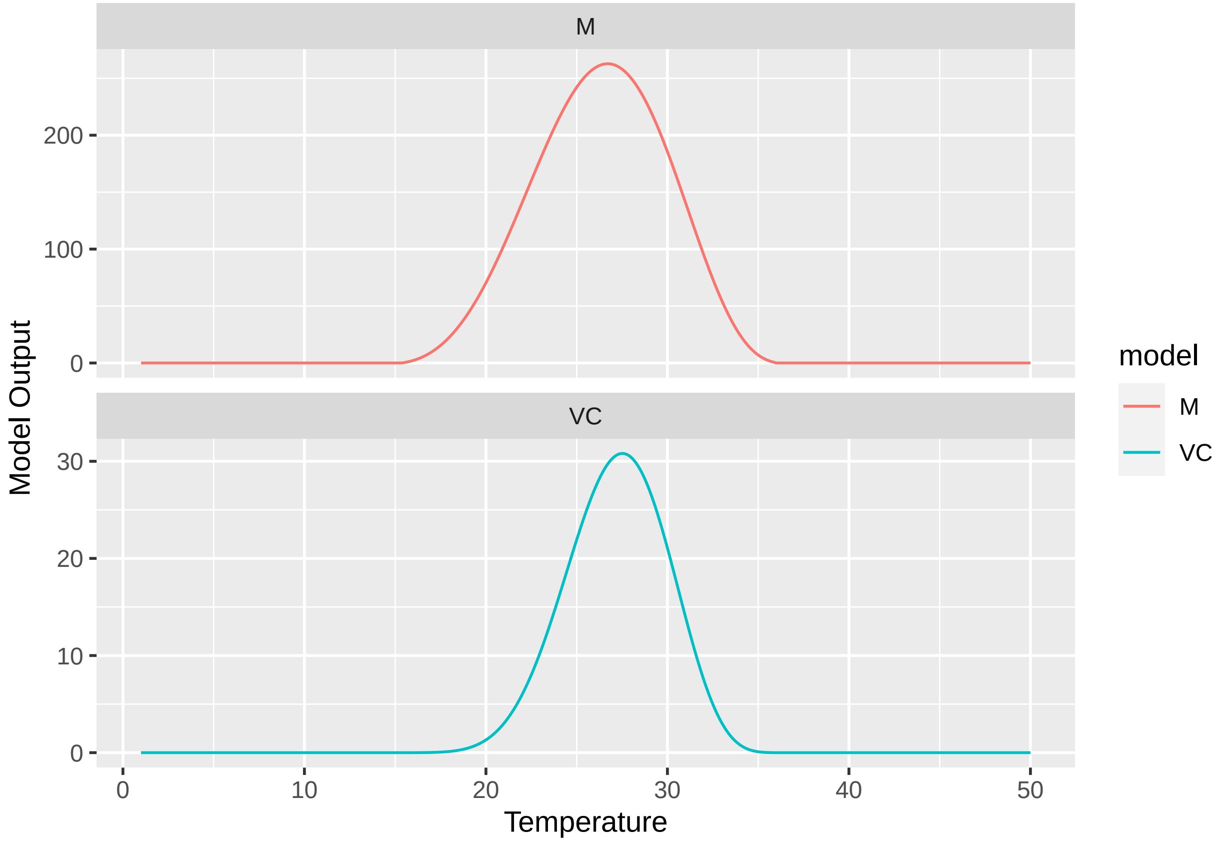
```r
lf <- tempcurve(Temp, "lf", mal_est_par)
EFD <- tempcurve(Temp, "EFD", mal_est_par)
PDR <- tempcurve(Temp, "PDR", mal_est_par)
bc <- tempcurve(Temp, "bc", mal_est_par)
pEA <- tempcurve(Temp, "pEA", mal_est_par)
MDR <- tempcurve(Temp, "MDR", mal_est_par)

traitvals <- c(a, lf, EFD, PDR, bc, pEA, MDR)
tempvals <- rep(Temp, 7)
traitnames <- c("a", "lf", "EFD", "PDR", "bc", "pEA", "MDR")
trait <- rep(traitnames, each = length(Temp))
traitdf <- data.frame(tempvals, traitvals, trait)

ggplot(data = traitdf) +
  geom_line(aes(x = tempvals, y = traitvals)) +
  facet_wrap(~ trait, ncol = 4, scales = "free_y") +
  labs(title = 'Malaria trait curves from Miazgowicz et al. (2020) "estimated" model',
       x = "Temperature",
       y = "Trait Values")
```



# Vectorial capacity

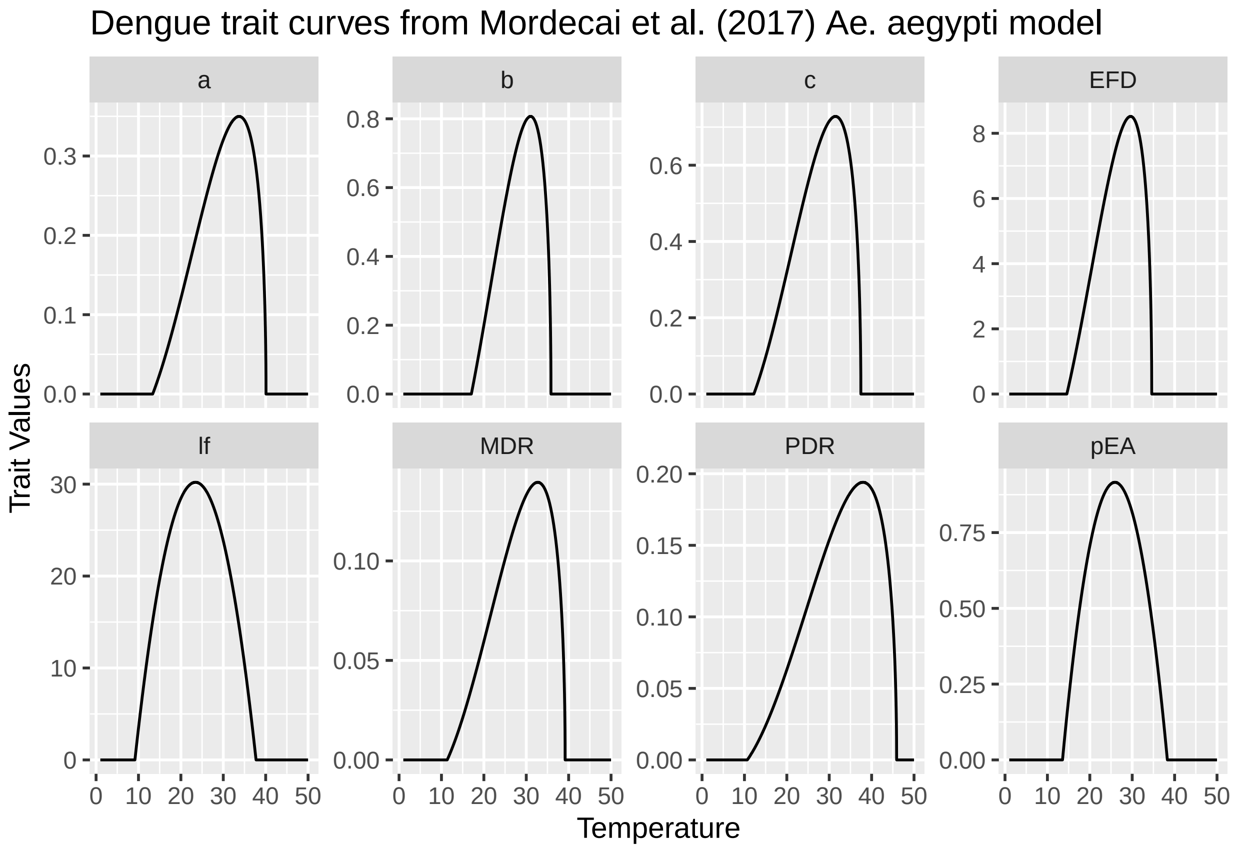
The `vc_rossmac()` function implements the temperature-dependent vectorial capacity (VC) equation derived from the Ross-Madonald malaria transmission model. Arguments include a vector of temperature values and a parameter table. The `bc` argument is set fo `TRUE` by default, indicating that a single `bc` trait variable is used for transmission. The argument should be set to `FALSE` for models where there are separate `b` and `c` trait variables, as is the case with the Mordecai et al. (2017) dengue transmission models. There is a separate vc_life() function for calculating VC based on the "lifetime" model from Miazgowicz et al. (2020).

```
mal_est_m <- m_trait(Temp,
                     lf = lf,
                     EFD = EFD,
                     pEA = pEA,
                     MDR = MDR)

mal_est_vc <- vc_rossmac(Temp,
                         M = mal_est_m,
                         a = a,
                         lf = lf,
                         PDR = PDR,
                         bc = bc,
                         bcvar = TRUE)

mal_vals <- c(mal_est_m, mal_est_vc)
tempvals <- rep(Temp, 2)
modelnames <- c("M", "VC")
model <- rep(modelnames, each = length(Temp))
mal_df <- data.frame(tempvals, mal_vals, model)
ggplot(data = mal_df) +
  geom_line(aes(x = tempvals, y = mal_vals, color = model)) +
  facet_wrap(~ model, scales = "free_y", ncol = 1) +
  labs(y = "Model Output", x = "Temperature")
```

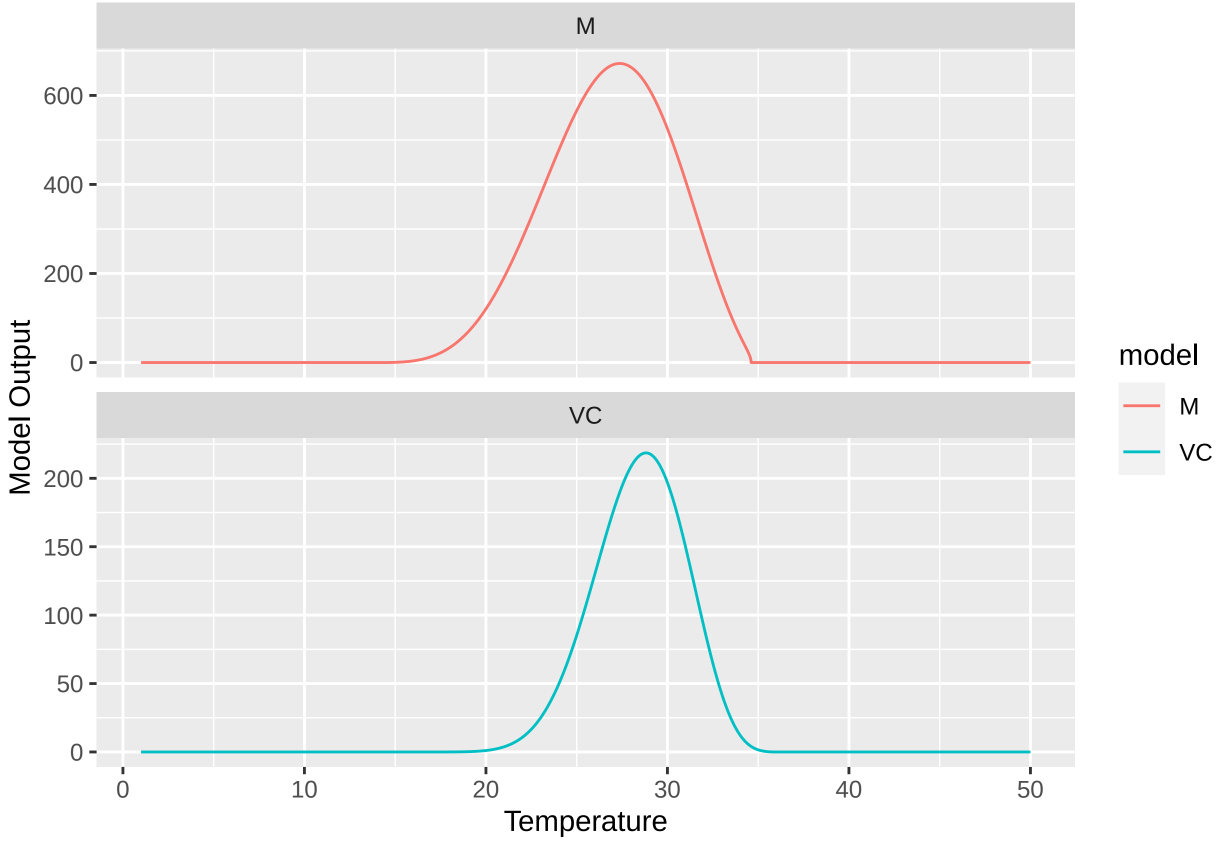


The vectorial capacity equations incorporate trait-based estimates of the mosquito abundance parameter (M). These values can also be extracted nad plotted as a function of temperature.

```r
a <- tempcurve(Temp, "a", den_aegypti_par)
lf <- tempcurve(Temp, "lf", den_aegypti_par)
EFD <- tempcurve(Temp, "EFD", den_aegypti_par)
PDR <- tempcurve(Temp, "PDR", den_aegypti_par)
b <- tempcurve(Temp, "b", den_aegypti_par)
c <- tempcurve(Temp, "c", den_aegypti_par)
pEA <- tempcurve(Temp, "pEA", den_aegypti_par)
MDR <- tempcurve(Temp, "MDR", den_aegypti_par)

traitvals <- c(a, lf, EFD, PDR, b, c, pEA, MDR)
tempvals <- rep(Temp, 8)
traitnames <- c("a", "lf", "EFD", "PDR", "b", "c", "pEA", "MDR")
trait <- rep(traitnames, each = length(Temp))
traitdf <- data.frame(tempvals, traitvals, trait)

ggplot(data = traitdf) +
  geom_line(aes(x = tempvals, y = traitvals)) +
  facet_wrap(~ trait, ncol = 4, scales = "free_y") +
  labs(title = 'Dengue trait curves from Mordecai et al. (2017) Ae. aegypti model',
       x = "Temperature",
       y = "Trait Values")
```



```r
den_est_m <- m_trait(Temp,
                     lf = lf,
                     EFD = EFD,
                     pEA = pEA,
                     MDR = MDR)

den_est_vc <- vc_rossmac(Temp,
                         M = mal_est_m,
                         a = a,
                         lf = lf,
                         PDR = PDR,
                         b = b,
                         c = c,
                         bcvar = FALSE)

den_vals <- c(den_est_m, den_est_vc)
tempvals <- rep(Temp, 2)
modelnames <- c("M", "VC")
model <- rep(modelnames, each = length(Temp))
den_df <- data.frame(tempvals, den_vals, model)
ggplot(data = den_df) +
  geom_line(aes(x = tempvals, y = den_vals, color = model)) +
  facet_wrap(~ model, scales = "free_y", ncol = 1) +
  labs(y = "Model Output", x = "Temperature")
```


M
VC
Model Output
Temperature
model
M
VC