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

Michael C. Wimberly

2024-10-18

Setup

The tidyverse suite of packages is used for data manipulation and graphing, and devtools is required for the `load_all()` function. Right now, the most straightforward way to use disclim is to (1) download the package from the GitHub archive at https://github.com/EcoGRAPH/disclim and (2) install it on a local folder on your computer, and (3) use `load_all()`is used to access this folder and read the disclim functions into the local environment. Before the package can be installed directly from GitHub and loaded with the `library()` function, some additional work will be needed to formalize access to the parameter tables.

```r
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 correspond 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 in the main package directory and are currently being read in using the base R function `read.csv()`. This is not really the right place for them, and future versions of the package will include a better mechanism for organizing and accessing these parameter datasets.

```r
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"))
```

Malaria 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 parameter table.

```r
mal_est_par
```

```
##   variable      form          p1        p2       p3
## 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", "p1", mal_est_par)
```

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

```
getpar("a", "p2", mal_est_par)
```

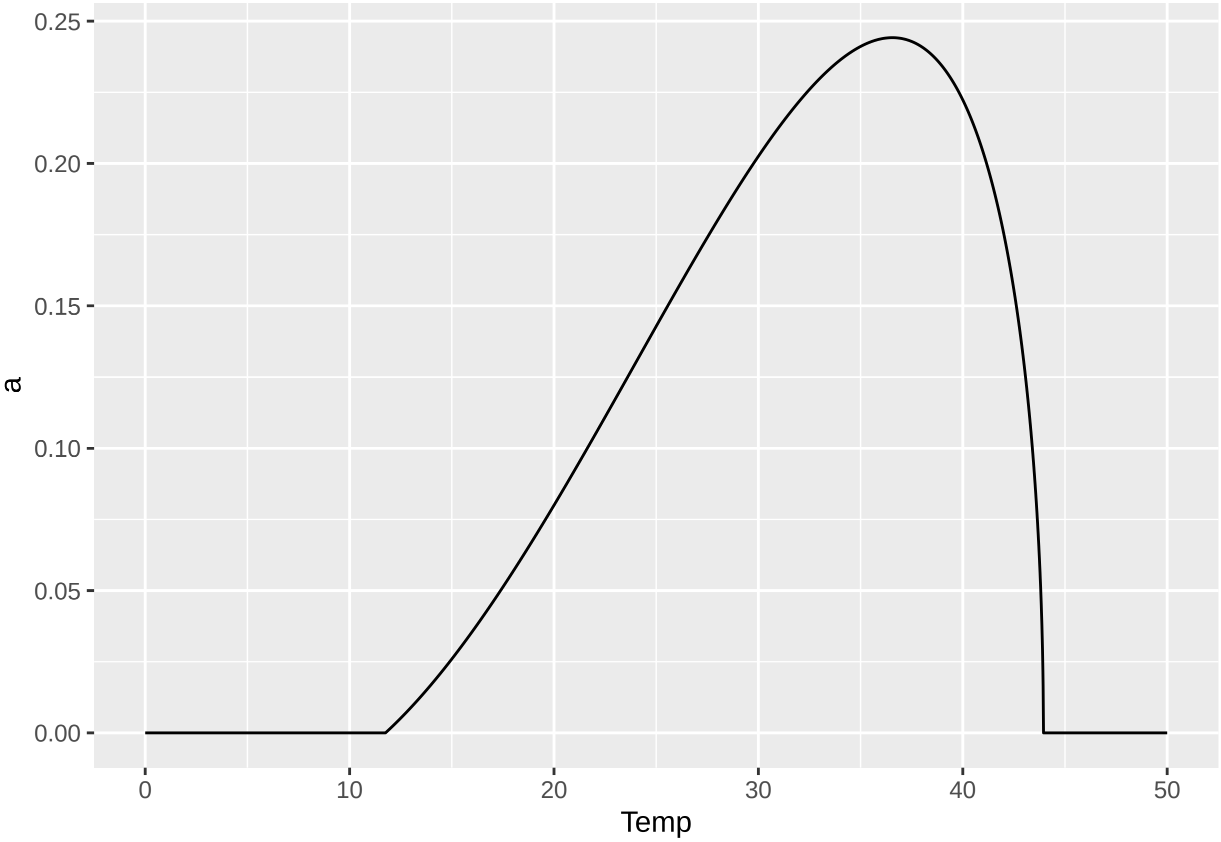
```
## [1] 11.75336
```

```
getpar("a", "p3", mal_est_par)
```

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

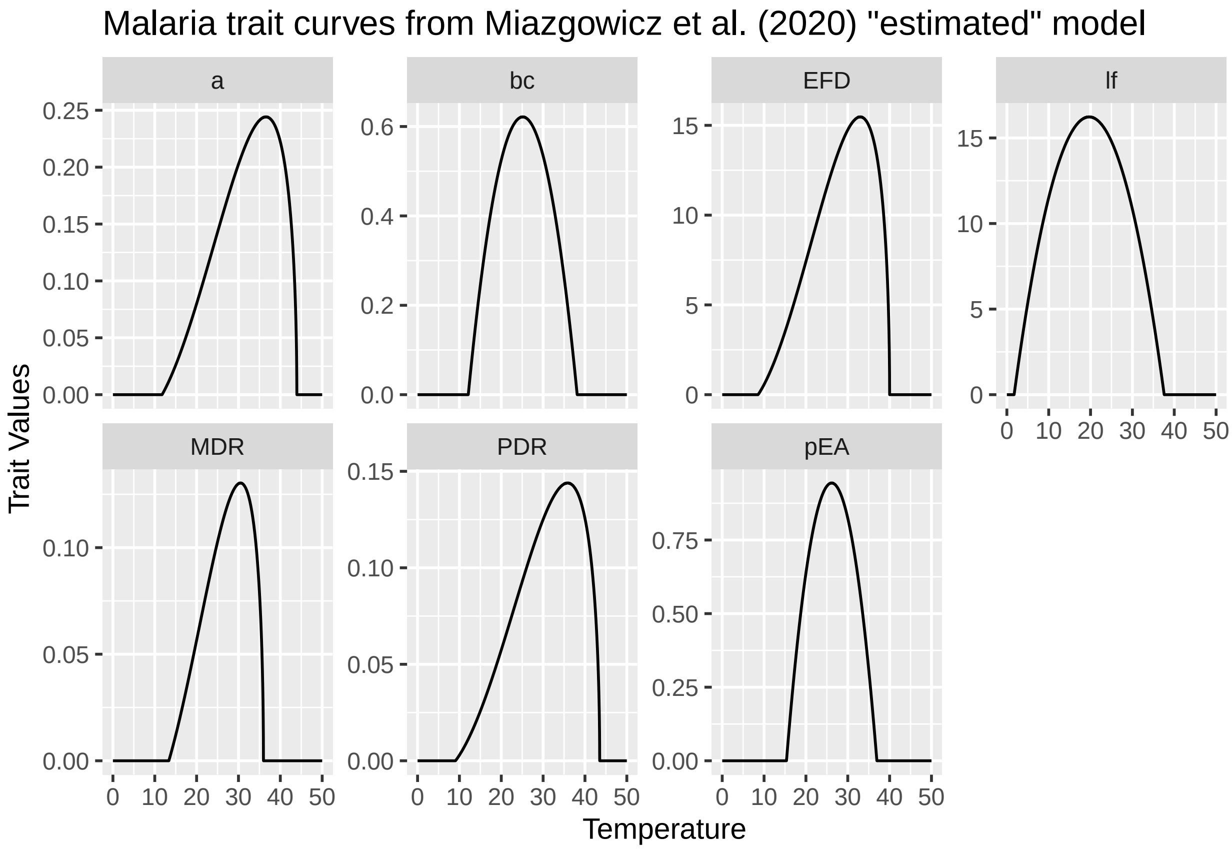
The `tempcurve()` function is used to calculate the trait values as a function of temperature for each variable. The function is vectorized, so a range of temperatures can be provided.

```
Temp <- seq(0, 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. The resulting trait estimates are then combined into a data frame and graphed.

```
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")
```

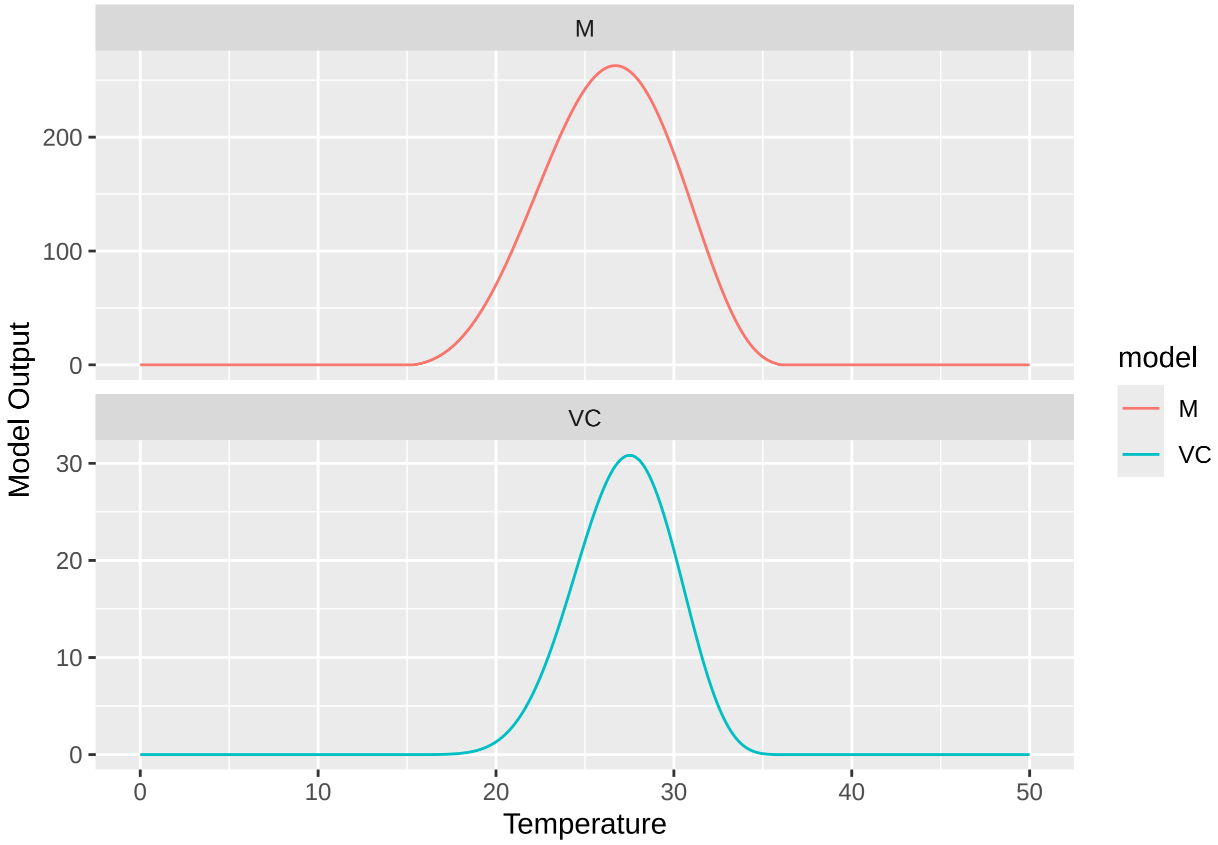


Malaria Vectorial Capacity

The `vc_rossmac()` function implements the temperature-dependent vectorial capacity (VC) equation derived from the Ross-Macdonald 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 can 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 also a `vc_life()` function for calculating VC based on the "lifetime" model from Miazgowicz et al. (2020). Mosquito abundance is estimated separately using the `m_trait()` function for the Ross-Macdonald model or the `m_life()`function for the lifetime model.

The following example is based on the "estimated" model for malaria transmission by *Anopheles stephensi* based using parameters from Miazgowicz et al. (2020).

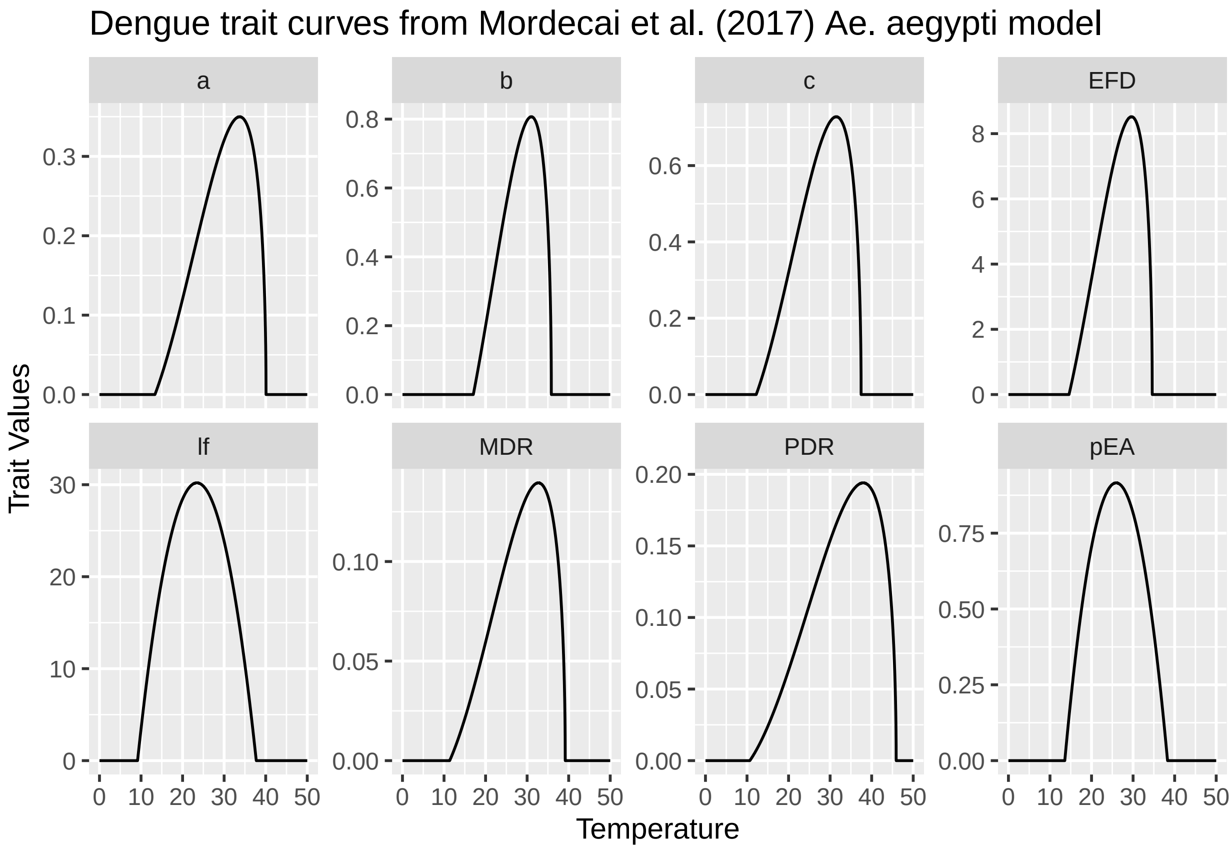
```
mal_est_m <- m_trait(lf = lf,
                     EFD = EFD,
                     pEA = pEA,
                     MDR = MDR)

mal_est_vc <- vc_rossmac(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")
```



Dengue Temperature-Trait Relationships

This example shows the temperature-trait curves for dengue transmission by *Aedes aegypti* based on the parameters reported in Mordecai et al. (2017).

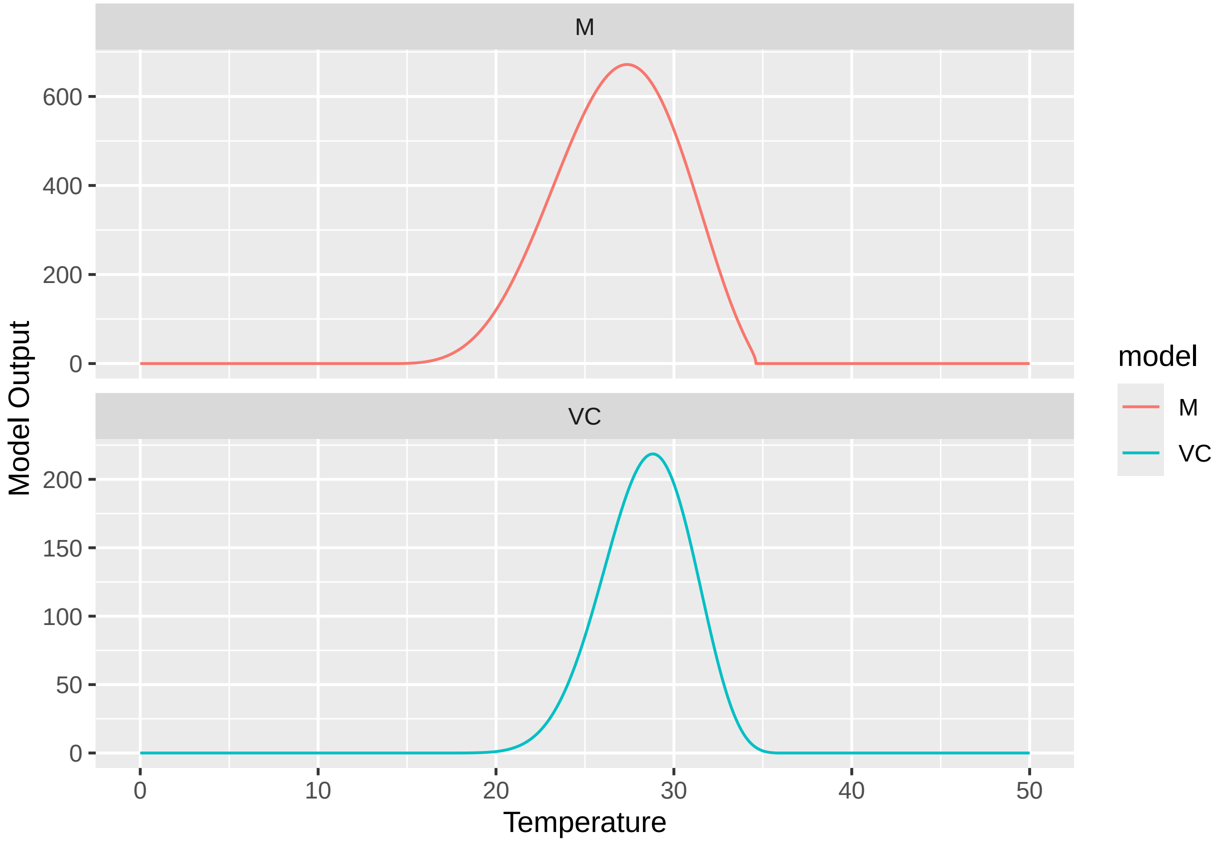
```
a <- tempcurve(Temp, "a", den_aegypti_par)
lf <- tempcurve(Temp, "lf", den_aegypti_par)
EFD <- tempcurve(Temp, "EFD", den_aegypti_par)
PDR <- tempcurve(Temp, "PDR", den_aegypti_par)
b <- tempcurve(Temp, "b", den_aegypti_par)
c <- tempcurve(Temp, "c", den_aegypti_par)
pEA <- tempcurve(Temp, "pEA",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



Dengue Vectorial Capacity

Finally, this example shows the estimates of mosquito abundance and vectorial capacity for dengue transmission by *Aedes aegypti* based on the temperature-trait curves shown in the last section. Note the `bcvar = FALSE` argument used in the `vc_rossmac()` function.

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

den_est_vc <- vc_rossmac(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")
```

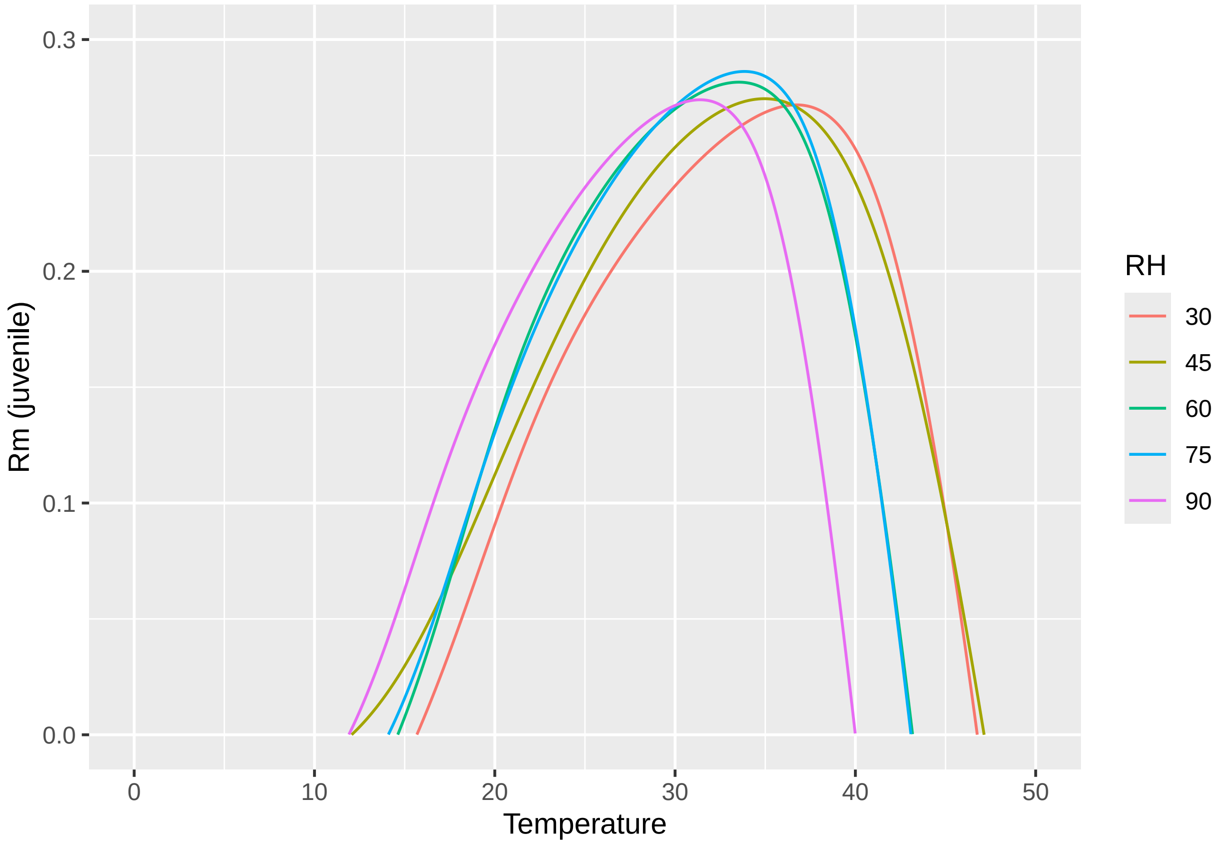


Temperature and Humidity Effects

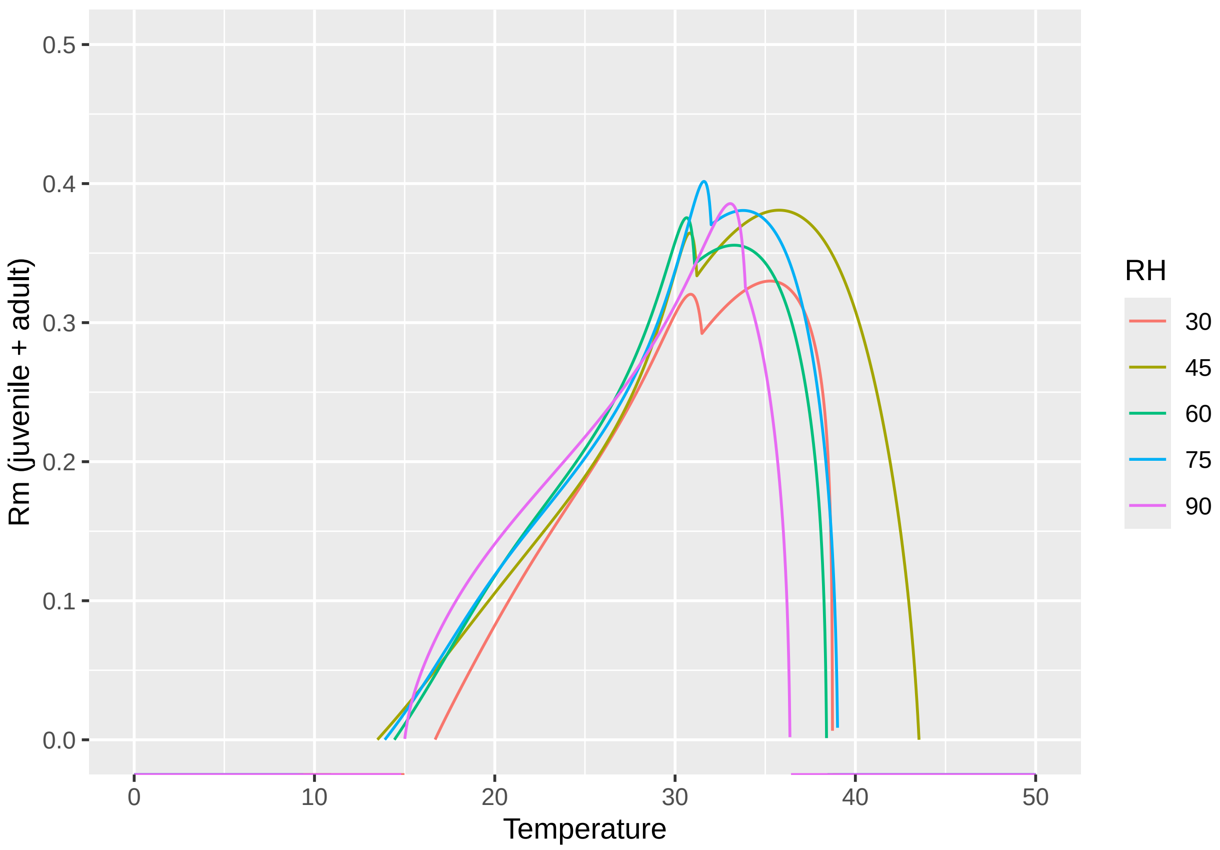
The thcurve() function is used to calculate temperature-dependent trait curves where the parameters vary with humidity. A file of parameters is loaded that contains parameters for temperature-trait curves estimated at five levels of relative humidity (30%, 45%, 60%, 75%, 90%). The rm_trait() function and the m_trait() are then used to calculate the maximum growth rate (rmj based only on temperature-dependent juvenile traits and rm based on temperature-dependent juvenile and adult traits) and the equilibrium mosquito abundance (m based on temperature-dependent juvenile and adult traits) as indices of environmental suitability.

```
pars_th <- read.csv(file.path(mypath, "Anstephensi_th_mal.csv"))

for(rh in c(30, 45, 60, 75, 90)) {
  assign(paste0("a", rh), thcurve(Temp, rh, "a", pars_th))
  assign(paste0("bmax", rh), thcurve(Temp, rh, "bmax", pars_th))
  assign(paste0("z", rh), thcurve(Temp, rh, "z", pars_th))
  assign(paste0("zj", rh), thcurve(Temp, rh, "zj", pars_th))
  assign(paste0("rmj", rh), rm_trait(zj = get(paste0("zj", rh)),
                                     a = get(paste0("a", rh)),
                                     bmax = 11.4,
                                     z = 1/0.12))
  assign(paste0("rm", rh), rm_trait(zj = get(paste0("zj", rh)),
                                    a = get(paste0("a", rh)),
                                    bmax = get(paste0("bmax", rh)),
                                    z = get(paste0("z", rh))))
  assign(paste0("mdens", rh), m_trait(pEA = get(paste0("zj", rh)),
                                      MDR = 1/get(paste0("a", rh)),
                                      EFD = get(paste0("bmax", rh)),
                                      lf = get(paste0("z", rh))))
}

rh <- rep(c(30, 45, 60, 75, 90), each = length(Temp))
temp <- rep(Temp, 5)
a <- c(a30, a45, a60, a75, a90)
bmax <- c(bmax30, bmax45, bmax60, bmax75, bmax90)
z <- c(z30, z45, z60, z75, z90)
zj <- c(zj30, zj45, zj60, zj75, zj90)
rmj <- c(rmj30, rmj45, rmj60, rmj75, rmj90)
rm <- c(rm30, rm45, rm60, rm75, rm90)
mdens <- c(mdens30, mdens45, mdens60, mdens75, mdens90)

rmj_sum <- tibble(rmj, rh, temp)
rm_sum <- tibble(rm, rh, temp)
mdens_sum <- tibble(mdens, rh, temp, a, bmax, z, zj)

ggplot(data = rmj_sum) +
  geom_line(aes(x = temp, y = rmj, col = as.factor(rh))) +
  labs(y = "Rm (juvenile)", x = "Temperature", color = "RH") +
  ylim(0, 0.3)
```

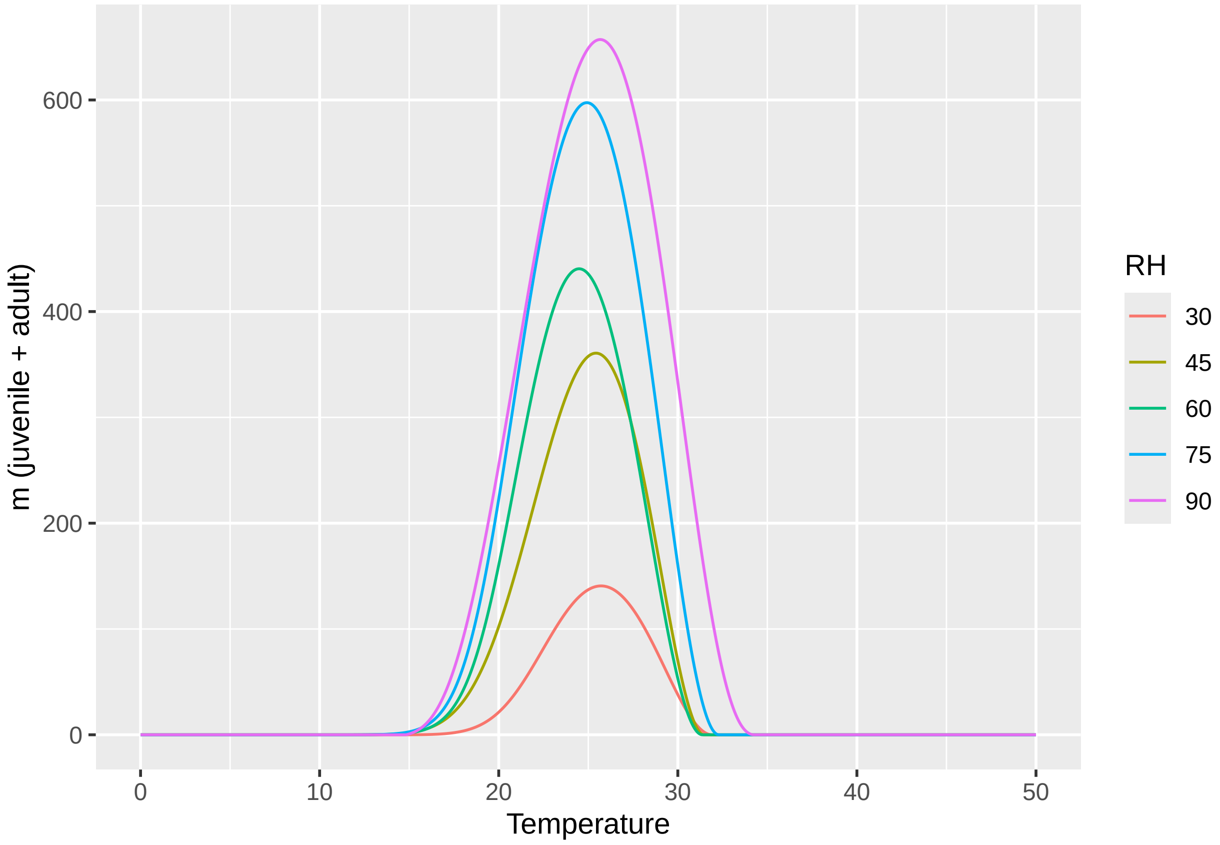
```
## Warning: Removed 9823 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
```



```
ggplot(data = rm_sum) +
  geom_line(aes(x = temp, y = rm, col = as.factor(rh))) +
  labs(y = "Rm (juvenile + adult)", x = "Temperature", color = "RH") +
  ylim(0, 0.5)
```



```
ggplot(data = mdens_sum) +
  geom_line(aes(x = temp, y = mdens, col = as.factor(rh))) +
  labs(y = "m (juvenile + adult)", x = "Temperature", color = "RH")
```



Calculating Temperature and Humidity Dependent Traits

This example shows how to calculate weather-dependent traits and environmental suitability indices using a table of temperature humidity values. An `rhvalues` parameter is used with the `thcurve()` function to specify a vector of relative humidity value (30%, 45%, 60%, 75%, and 90%) that are present in the parameter table. The function then extracts the value from this vector that is closest to the input value and uses it to extract the appropriate parameters from the lookup table. The `th_curve()`, `rm_trait()`, and `m_trait()` functions are implemented using the `mutate()` function so that the trait values and environmental suitability indices they return are added as new columns to the data frame.

```
temp <- rep(c(20, 25, 30, 35), each = 4)
rh <- rep(c(30, 50, 70, 90), times = 4)
climate <- tibble(temp, rh)

rhvals <- c(30, 45, 60, 75, 90)

climate2 <- climate %>%
  mutate(a = thcurve(temp, rh, "a", pars_th, rhvals),
         bmax = thcurve(temp, rh, "bmax", pars_th, rhvals),
         zj = thcurve(temp, rh, "zj", pars_th, rhvals),
         z = thcurve(temp, rh, "z", pars_th, rhvals),
         rmj = rm_trait(zj, a, bmax = 11.4, z = 1/0.12),
         rm = rm_trait(zj, a, bmax, z),
         mdens = m_trait(pEA = zj, MDR = 1/a, EFD = bmax, lf = z))
climate2
```

```
## # A tibble: 16 x 9
##     temp    rh     a  bmax    zj     z    rmj     rm mdens
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl> <dbl>
##  1    20    30 23.7   8.23 0.195 17.7  0.0906 0.0823  21.4
##  2    20    50 24.4  11.9  0.416 22.5  0.112  0.106  102.
##  3    20    70 21.2  14.0  0.494 26.1  0.130  0.118  223.
##  4    20    90 17.5  10.9  0.795 22.7  0.168  0.141  255.
##  5    25    30 15.1  17.6  0.707 12.9  0.181  0.187  137.
##  6    25    50 13.5  20.5  0.735 17.9  0.197  0.189  358.
##  7    25    70 12.1  22.0  0.876 19.4  0.219  0.203  597.
##  8    25    90 10.9  22.3  0.930 18.5  0.236  0.218  649.
##  9    30    30 10.6  25.2  0.865  4.29 0.237  0.306   38.0
## 10    30    50  9.10 28.6  0.805  5.26 0.253  0.336   70.0
## 11    30    70  8.46 27.8  0.910  7.32 0.271  0.337  160.
## 12    30    90  8.33 30.0  0.885 10.2  0.272  0.313  333.
## 13    35    30  8.23 25.7  0.821  0    0.269  0.330    0
## 14    35    50  7.33 33.6  0.704  0    0.274  0.379    0
## 15    35    70  6.99 27.0  0.739  0    0.284  0.374    0
## 16    35    90  7.29 21.9  0.422  0    0.241  0.267    0
```