Appendix S1

Buckley et al.

We thank Mike Kearney for making modular R functions corresponding to NicheMapR available and providing code comparing TrenchR and NicheMapR functions during the review process. Our adaptation of the provided code corresponds to the operative temperature estimation in the manuscript. The NicheMapR implementation relies on the micro_global environmental data available through NicheMapR. The modular NicheMapR functions are available in version 3.2.1 via GitHub (https://github.com/mrke/NicheMapR/releases).

We provide the input data corresponding to the manuscript example.

```
# Set up input data
lat  <- 35.69    # Latitude (degrees)
lon  <- -105.944 # Longitude (degrees)
elev <- 2121     # Elevation (meters)

# Assumptions
tau <- 0.7 # Atmospheric transmissivity
rho <- 0.6 # Albedo
Tb0 <- 25  # Initial assumption of body temperature (C)
doy <- day_of_year("2021-06-15", format = "%Y-%m-%d") # Day of year, changed
to mid June to correspond to micro_global data
z <- 0.02 # height of animal
psa_g <- 0.33 # fraction of animal surface contacting the ground
K_skin <- 0.5 # parameter for thermal conductivity of skin, W/mC
K_sub <- 0.5 # parameter for thermal conductivity of substrate, W/mC
```

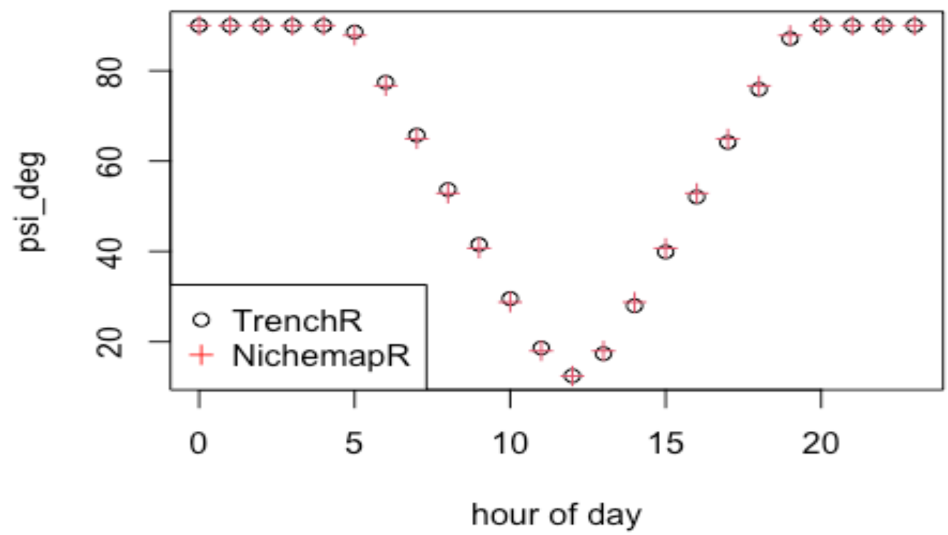
We then run the NicheMapR microclimate model for clear sky conditions at this site, passing elevation (would otherwise have been 2257m), soil albedo and animal height used in the example. The microclimate model output is prepared as model input.

```
#run microclimate model
micro <- micro_global(loc = c(lon, lat), elev = elev, clearsky = 1, REFL = rh
o, Usrhyt = z)

## If program is crashing, try run.gads = 2.

## extracting climate data

## extracting soil moisture data

## running microclimate model for 12 days by 1 years at site long -105.944 la
t 35.69

## Note: the output column `SOLR` in metout and shadmet is for unshaded horiz
ontal plane solar radiation
```

```
## runtime 0.0539999999999994 seconds

metout <- as.data.frame(micro$metout)
soil <- as.data.frame(micro$soil)

# choose mid-June
metout_sub <- subset(metout, DOY == 166)
soil_sub <- subset(soil, DOY == 166)

# use min and max values of air and soil temperature for this day
# and max wind speed
Tmin   <- min(metout_sub$TAREF) # Minimum air temperature (C)
Tmax   <- max(metout_sub$TAREF) # Maximum air temperature (C)
Tmin_s <- min(soil_sub$D0cm) # Minimum soil temperature (C)
Tmax_s <- max(soil_sub$D0cm) # Maximum soil temperature (C)
u      <- max(metout_sub$VREF) # Wind speed (m/s)

snoon <- solar_noon(lon = lon, doy = doy) # Estimate solar noon
dayl  <- daylength(lat = lat, doy = doy)  # Estimate day length

tr    <- snoon - dayl / 2 # Time of sunrise
ts    <- snoon + dayl / 2 # Time of sunset
```

We compare the estimation of zenith angles between TrenchR and NicheMapR.

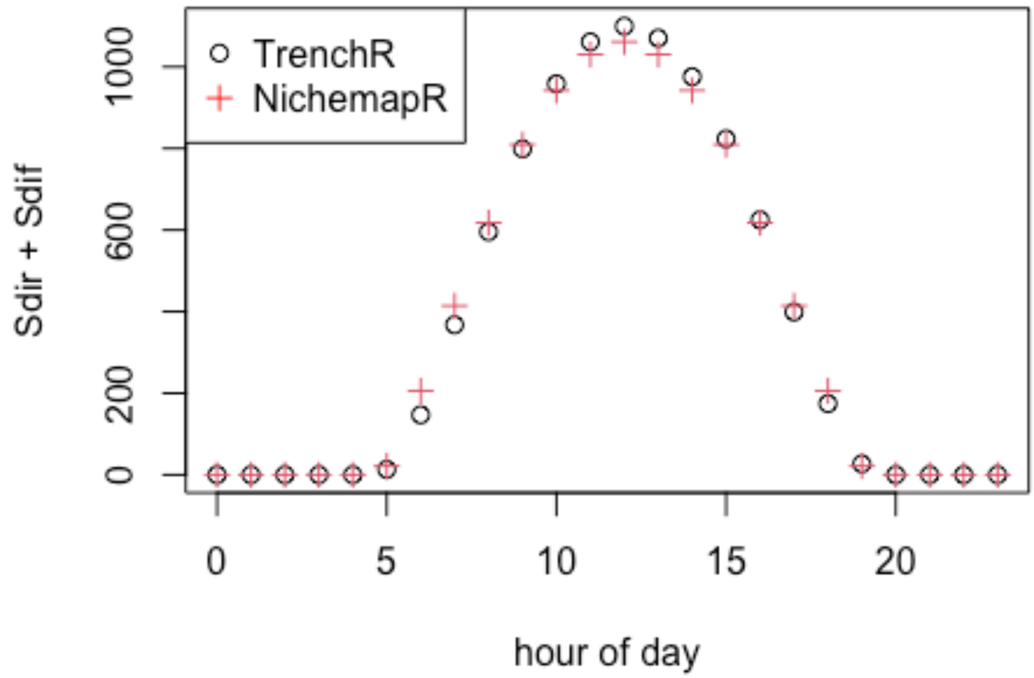
```
# Estimate zenith angle (degrees)
psi_deg <- sapply(0:23, FUN = zenith_angle, doy = doy, lat = lat, lon = lon)

# compare TrenchR and NMR
plot(seq(0, 23), psi_deg, xlab = 'hour of day')
points(seq(0, 23), metout_sub$ZEN, pch = 3, col = 2)
legend("bottomleft",
       legend = c("TrenchR", "NichemapR"),
       lty = NULL, pch = c(1,3), col = c("black","red"))
```




```
# Convert to radians
psi_rad <- degrees_to_radians(psi_deg)
```

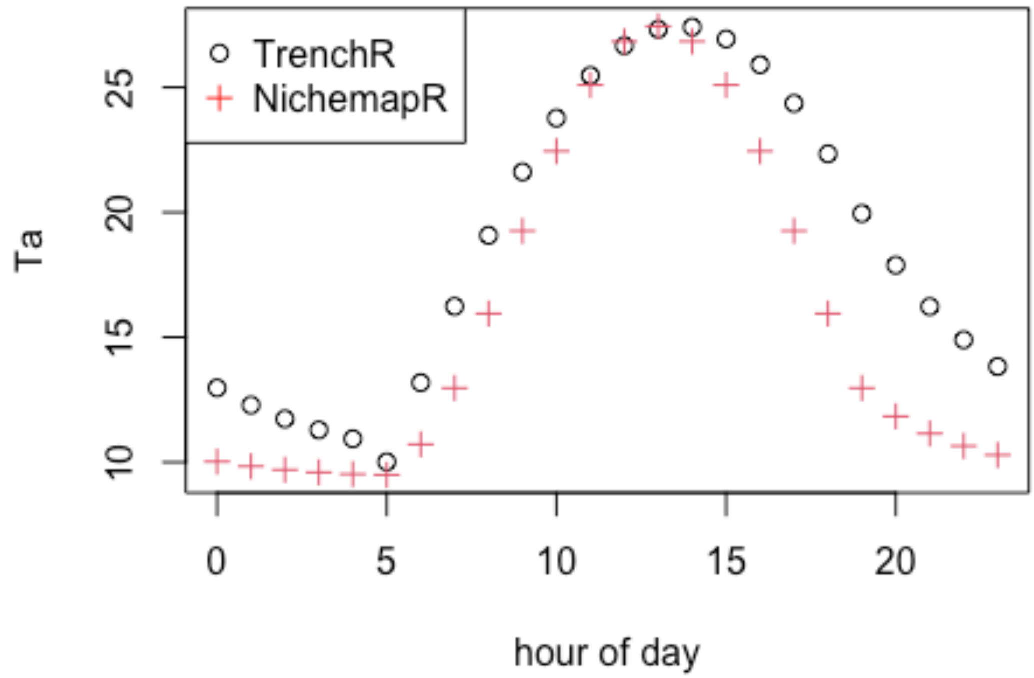
We next compare the estimation of radiation, air temperatures, and radiation between the packages.

```
# Estimate radiation
Srad <- sapply(psi_rad, FUN = solar_radiation, doy = doy, tau = tau, elev = e
lev, rho = rho)
# Separate solar radiation into direct, diffuse, and reflected components
Sdir <- Srad[1,] # Direct solar radiation (W/m2)
Sdif <- Srad[2,] # Diffuse solar radiation  (W/m2)
Sref <- Srad[3,] # Reflected solar radiation  (W/m2)

# compare TrenchR and NMR
plot(seq(0, 23), Sdir + Sdif, xlab = 'hour of day')
points(seq(0, 23), metout_sub$SOLR, pch = 3, col = 2)
legend("topleft",
       legend = c("TrenchR", "NichemapR"),
       lty = NULL, pch = c(1,3), col = c("black","red"))
```

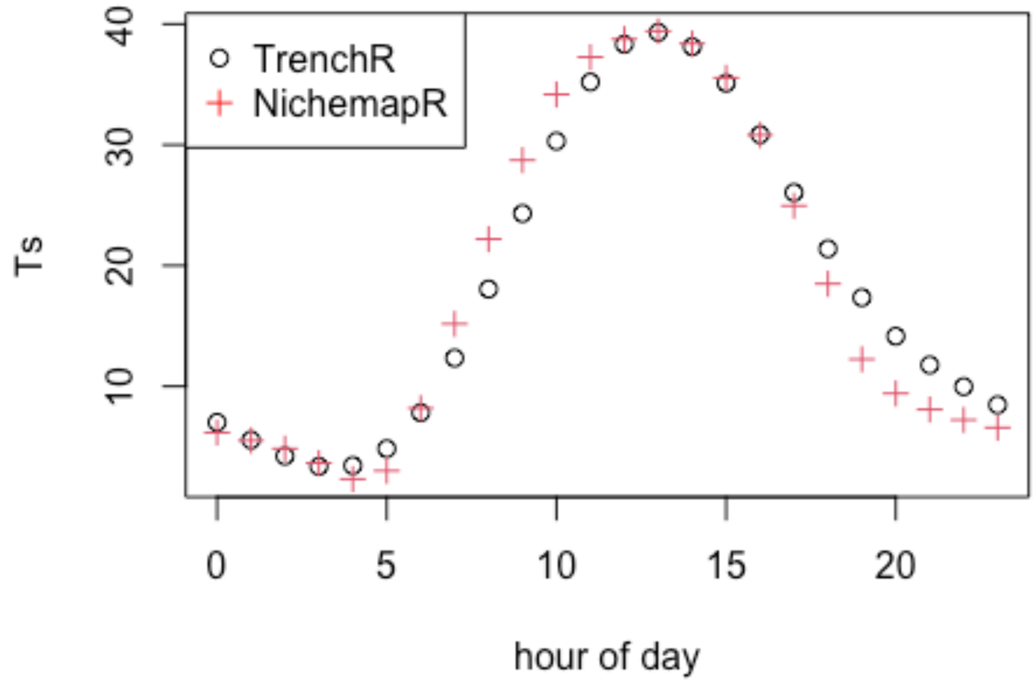


We then compare air and soil temperatures. We note that TrenchR offers several different functions for estimating diurnal temperature variation and also offers more sophisticated functions for estimating soil temperatures than that used here.

```
# Air temperature (C)
Ta <- sapply(1:24, diurnal_temp_variation_sineexp, T_max = Tmax, T_min = Tmin
, t_r = tr, t_s = ts, alpha = 2.59, beta = 1.55, gamma = 2.2)
plot(seq(0, 23), Ta, ylim = c(Tmin, Tmax), xlab = 'hour of day')
points(seq(0, 23), metout_sub$TAREF, pch = 3, col = 2)
legend("topleft",
       legend = c("TrenchR", "NichemapR"),
       lty = NULL, pch = c(1,3), col = c("black","red"))
```

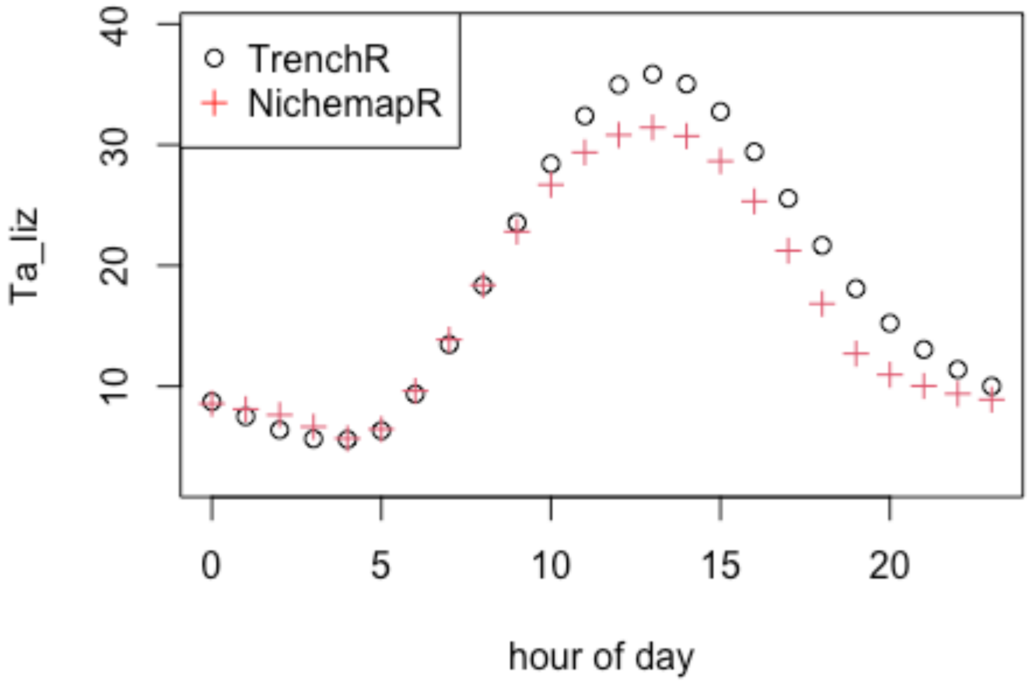



```
# Soil surface temperature (C)
Ts <- sapply(1:24, diurnal_temp_variation_sine, T_max = Tmax_s, T_min = Tmin_
s)
plot(seq(0, 23), Ts, ylim = c(Tmin_s, Tmax_s), xlab = 'hour of day')
points(seq(0, 23), soil_sub$D0cm, pch = 3, col = 2)
legend("topleft",
       legend = c("TrenchR", "NichemapR"),
       lty = NULL, pch = c(1,3), col = c("black","red"))
```

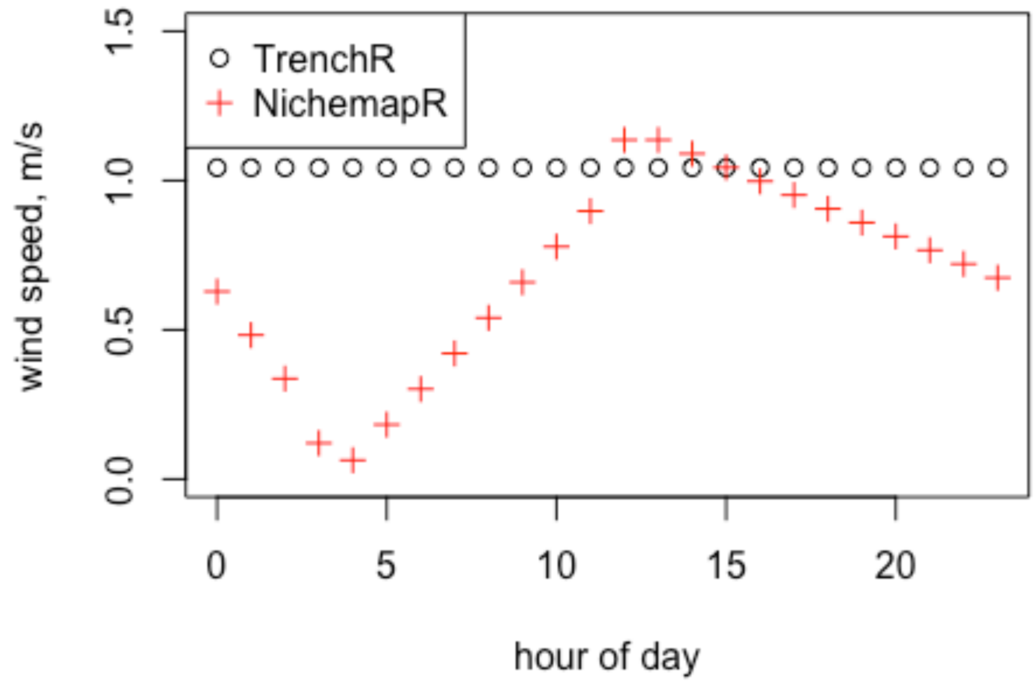


We then compare air temperature and windspeed profiles associated with free convection. We note that TrenchR assumed fixed windspeed for simplicity in the manuscript.

```
# Neutral air temperature profile
Ta_liz  <- air_temp_profile_neutral(T_r = Ta, zr = 2, z0 = 0.004, z = z, T_s
= Ts) # MRK used z0=0.004 instead of 0.2
plot(seq(0, 23), Ta_liz, ylim = c(Tmin_s, Tmax_s), xlab = 'hour of day')
points(seq(0, 23), metout_sub$TALOC, pch = 3, col = 2)
legend("topleft",
       legend = c("TrenchR", "NichemapR"),
       lty = NULL, pch = c(1,3), col = c("black","red"))
```



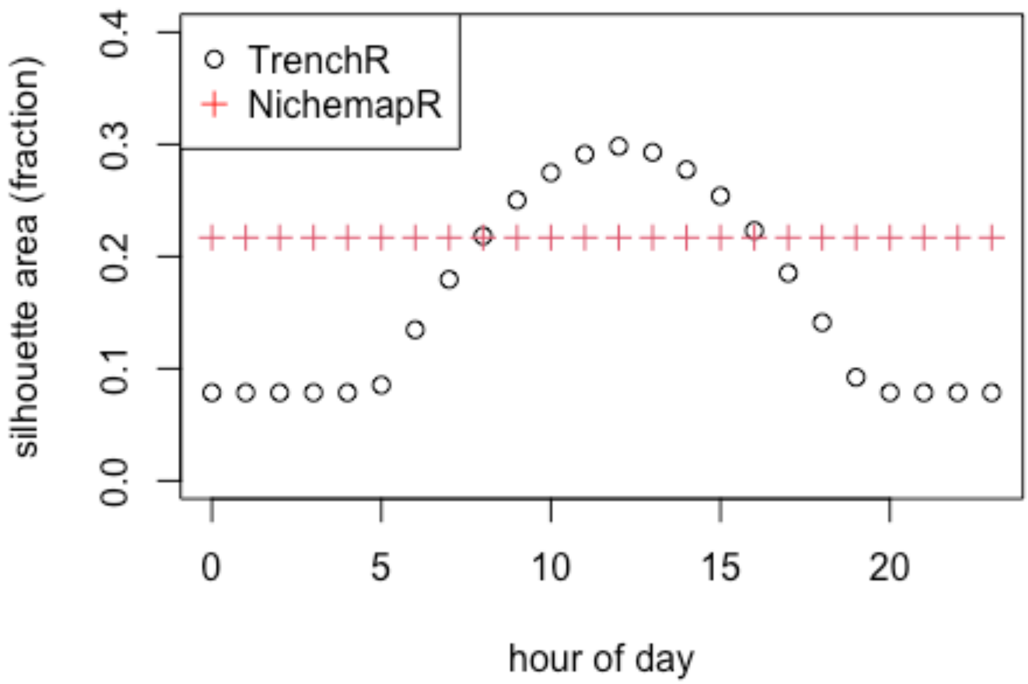

```
# Neutral wind speed profile
V_liz <- wind_speed_profile_neutral(u_r = u, zr = 2, z0 = 0.004, z = z) # MRK
used z0=0.004 instead of 0.2
plot(seq(0, 23), rep(V_liz, 24), xlab = 'hour of day', ylab = 'wind speed, m/
s', ylim = c(0, 1.5))
points(seq(0, 23), metout_sub$VLOC, pch = 3, col = "red")
legend("topleft",
       legend = c("TrenchR", "NichemapR"),
       lty = NULL, pch = c(1,3), col = c("black","red"))
```



Next, set up input for estimating solar radiation and estimate silhouette area

```
mass <- 10  # Mass (g)
a    <- 0.9 # Solar absorptivity (proportion)
# Estimate surface area (m^2) and the proportion sihouette area
A   <- surface_area_from_mass(mass, "lizard")

# use NicheMapR GEOM function
GEOM.out <- GEOM_ecto(AMASS = mass/1000, GEOMETRY = 3, PTCOND = psa_g, PMOUTH
= 0)
A_NMR <- GEOM.out$AREA
ASILN <- GEOM.out$ASILN
ASILP <- GEOM.out$ASILP
psa_NMR <- 0.5 * (ASILN + ASILP) / A_NMR

# compare outputs
A

## [1] 0.004578742

A_NMR

## [1] 0.005105057
```

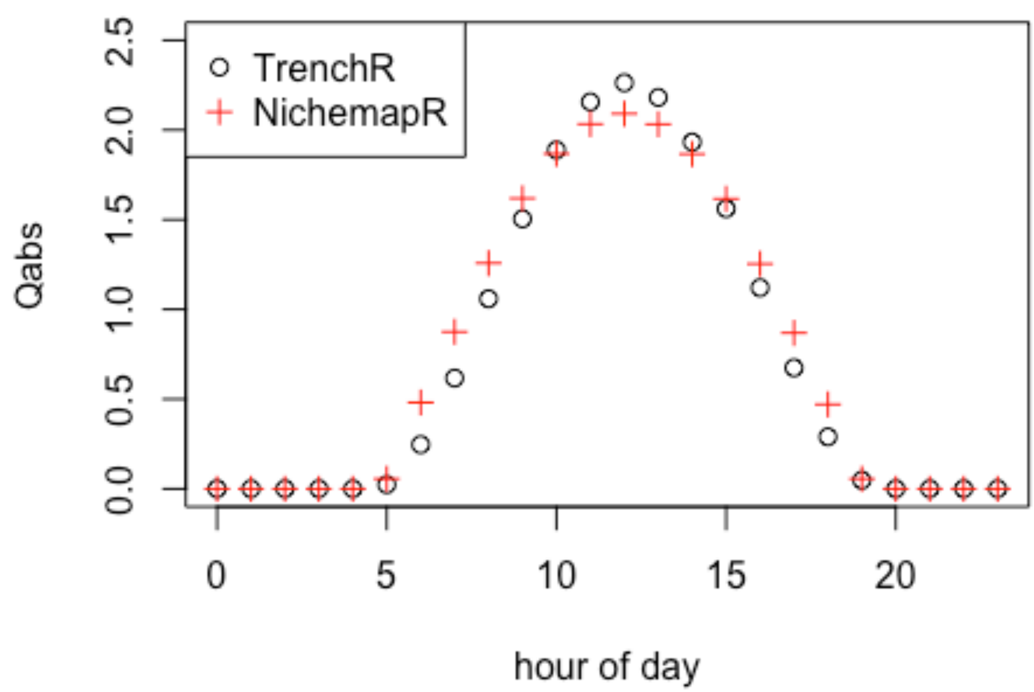
```r
psa <- sapply(psi_deg, proportion_silhouette_area, taxon = "lizard", posture
= "prostrate")
# Change negative values to zero
psa[psa < 0] = 0
```



We then estimate and compare absorption of solar radiation

```r
Qabs <- rep(NA, 24)
Qabs_NMR <- Qabs
p_dif <- Sdif / (Sdir + Sdif)
# MRK note that psa_dif has been added as a parameter, set to half, and psa_r
ef has been set to the bottom half minus fraction in contact with ground
Qradiation_absorbed2 <- function (a = 0.9, A, psa_dir, psa_dif, psa_ref, S_di
r, S_dif, S_ref = NA,
                                  a_s = NA)
{
  stopifnot(a >= 0, a <= 1, A > 0, psa_dir >= 0, psa_dir <=
            1, psa_ref >= 0, psa_ref <= 1, psa_dif >= 0, psa_dif <= 1,
            S_dir > 0, S_dif > 0)
  if (is.na(S_ref)) {
    S_ref <- a_s * S_dir
  }
  A_dir <- A * psa_dir
```

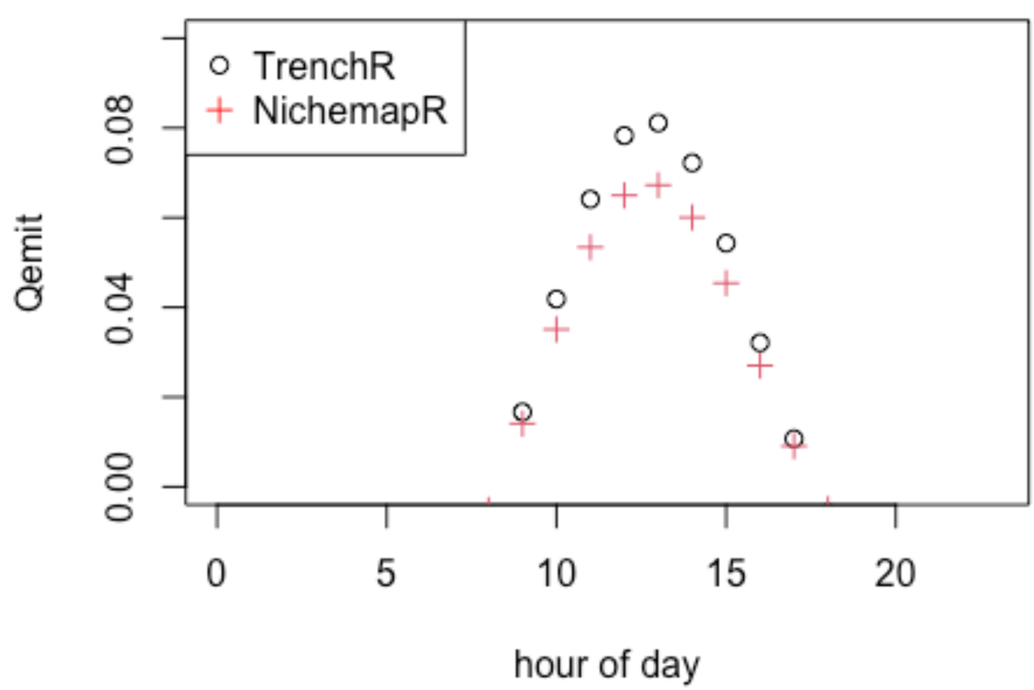
```
  A_dif <- A * psa_dif
  A_ref <- A * psa_ref
  a * A_dir * S_dir + a * A_dif * S_dif + a * A_ref * S_ref
}
for (hour in 1:24) {
  Qabs[hour] <- Qradiation_absorbed(a = a, A = A, psa_dir = psa[hour], psa_di
f = 0.5, psa_ref = 0.5 * (1 - psa_g), S_dir = Sdir[hour], S_dif = Sdif[hour],
rho = rho)
  # use SOLAR_ecto for NMR prediction, assuming halfway between perpendicular
and parallel posture
 Qabs_NMR[hour] <- SOLAR_ecto(QSOLR = metout_sub$SOLR[hour], ATOT = A, FATOSK
= 0.5, FATOSB = 0.5, ASIL = (ASILN + ASILP) / 2, ABSAN = a, ABSSB = 1 - rho,
ZEN = metout_sub$Z[hour] / 180 * pi, postur = 0, PDIF = p_dif[hour])$QSOLAR
}
```



We then estimate and compare net emitted thermal radiation.

```
epsilon_s <- 0.965 # Surface emmissivity of lizards
Qemit <- rep(NA, 24)
Qemit_NMR <- rep(NA, 24)
for (hour in 1:24) {
  Qemit[hour] <- Qemitted_thermal_radiation(epsilon = epsilon_s, A = A, psa_d
ir = 0.5, psa_ref = 0.5 * (1 - psa_g), T_b = Ta_liz[hour] + 273.15, T_g = Ts[
```

```
hour] + 273.15, T_a = Ta[hour] + 273.15, enclosed = TRUE)
  # use RADOUT_ecto for NMR prediction
  Qemit_NMR[hour] <- RADOUT_ecto(TSKIN = Ta_liz[hour], ATOT = A, EMISAN = eps
ilon_s) - RADIN_ecto(ATOT = A, EMISAN = epsilon_s, EMISSB = 1, EMISSK = 1, TS
KY = Ta[hour], TGRD = Ts[hour])
}
```



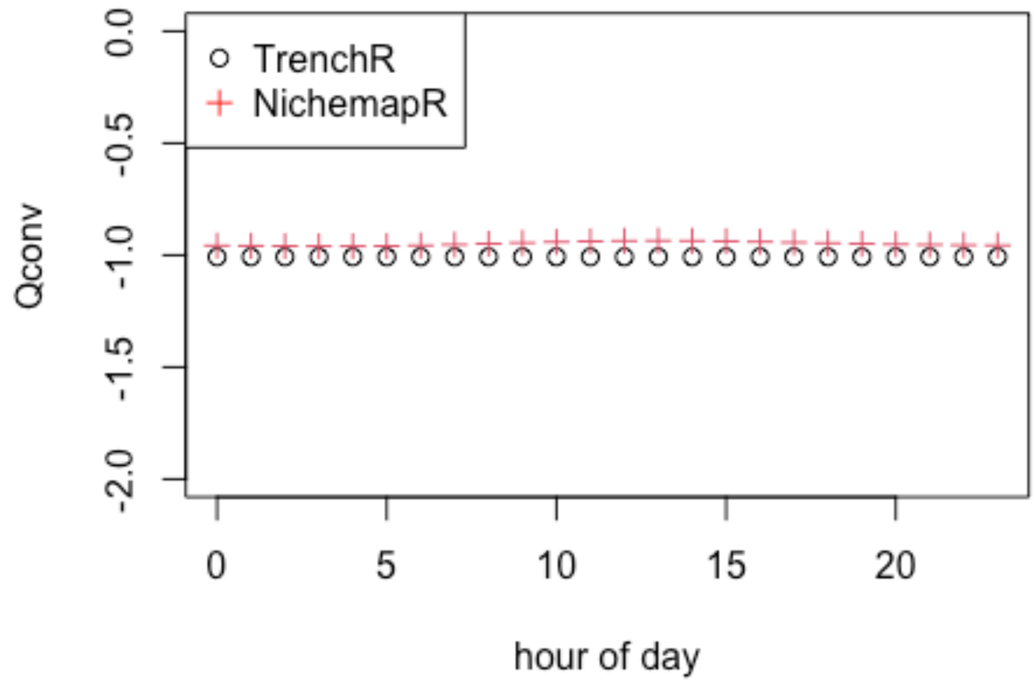
We then estimate and compare heat transfer coefficients.

```
# Use DRYAIR from NicheMapR to estimate the thermal conductivity of air and k
inematic viscosity
ap <- airpressure_from_elev(elev) * 1000 # Barometric pressure (pascal)
DRYAIRout <- DRYAIR(db = Ta, bp = ap, alt = elev)

K  <- mean(DRYAIRout$thcond) # Thermal conductivity (Wm^-2K^-1)
nu <- mean(DRYAIRout$viskin) # Kinematic viscosity (m2 s-1)

# Estimate the characteristic dimension as cube root of volume, assuming dens
ity of water as 1000kg/m^3

D <- ((mass / 1000) / 1000) ^ (1 / 3)

# Estimate the heat transfer coefficient using an empirical relationship for
```

```
lizards
H_L <- heat_transfer_coefficient(u = V_liz, D = D, K = K, nu = nu, taxon = "l
izard_surface")
# Estimate the heat transfer coefficient using a spherical approximation
H_L2 <- heat_transfer_coefficient_approximation(u = V_liz, D = D, K = K, nu =
nu, taxon = "lizard")
# Estimate the heat transfer coefficient using a simplified version of the ap
proximation
H_L3 <- heat_transfer_coefficient_simple(u = V_liz, D = D, type = "Gates")

# run NMR CONV_ecto to get characteristic dimension
CONV.out <- CONV_ecto(GEOMETRY = 3, AL = GEOM.out$AL)

# compare heat transfer coefficients
H_L_NMR <- CONV.out$HC #NicheMapR
H_L

## [1] 25.27445

H_L_NMR

## [1] 32.61385
```

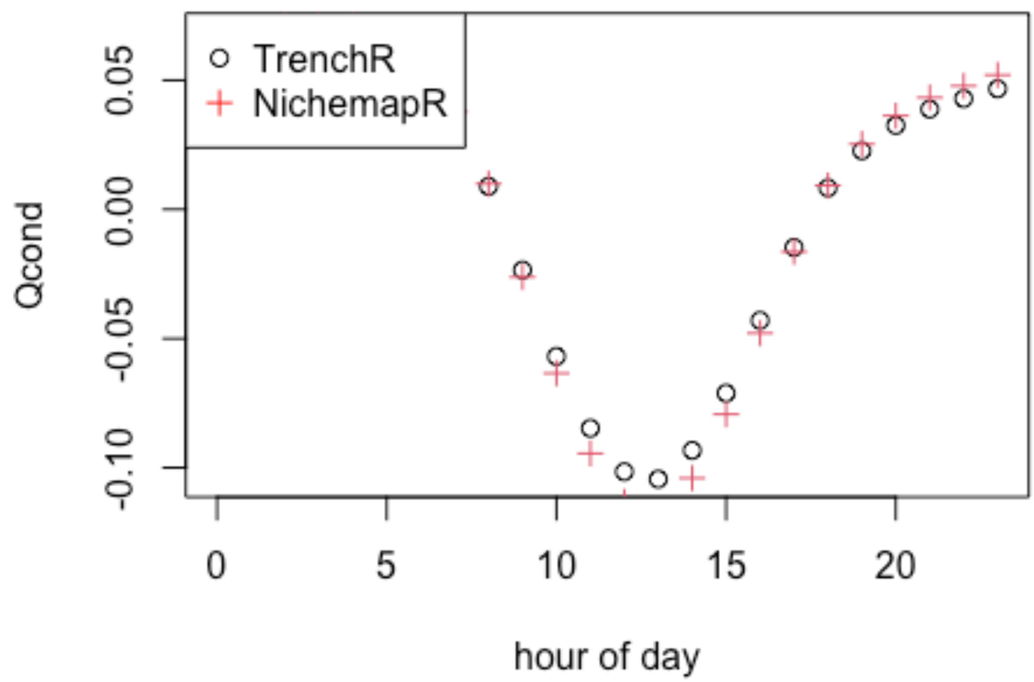
We then estimate and compare convection. NichemapR estimates a higher convection coefficient and thus a greater magnitude of convection.

```
Qconv <- rep(NA, 24)
Qconv_NMR <- Qconv

offset <- 10 # adding an offset to air temperature, otherwise just get zero
for (hour in 1:24) {
  Qconv[hour] <- Qconvection(T_a = Ta_liz[hour] + 273.15 + offset, T_b = Ta_l
iz[hour] + 273.15, H = H_L, A = A, proportion = 1 - psa_g, ef = 1.3)
  # use CONV_ecto for NMR prediction
  Qconv_NMR[hour] <- CONV_ecto(GEOMETRY = 3, AL = GEOM.out$AL, AV = GEOM.out$
AV, ATOT = GEOM.out$AREA, BP = ap, ALT = elev, TA = Ta_liz[hour] + offset, TS
KIN = Ta_liz[hour])$QCONV
}
```



We then estimate and compare conduction.

```
Qcond <- rep(NA, 24)
Qcond_NMR <- Qcond

for(hr in 1:24) {
  Qcond[hr] <- Qconduction_animal(T_g = Ts[hr] + 273.15, T_b = Ta_liz[hr] + 2
73.15, d = 0.025, K = K_skin, A = A, proportion = psa_g)

  # use COND_ecto for NMR prediction
  Qcond_NMR[hr] <- COND_ecto(AV = GEOM.out$AV, TSKIN = Ta_liz[hr], TSUBST = T
s[hr], SUBTK = K_sub)
}
```



We estimate operative temperatures as in the manuscript, but add the biophysical model from NichemapR.

```
# General biophysical model from Gates (1980)
Te <- rep(NA, 24)
for (hour in 1:24) {
  Te[hour] <- Tb_Gates(A = A, D = D, psa_dir = psa[hour], psa_ref = 0.5 * (1
- psa_g), psa_air = 1 - psa_g, psa_g = psa_g, T_g = Ts[hour], T_a = Ta_liz[ho
ur], Qabs = Qabs[hour], epsilon = epsilon_s, H_L = H_L, ef = 1.3, K = K_sub)
# MRK made D = 2.5 cm and passed K_sub
}

# General biophysical model from Campbell and Norman (2000)
Te2 <- rep(NA, 24)
for (hr in 1:24) {
  # S is solar radiation flux (W m^-2), so we divide by surface area, A
  Te2[hr] <- Tb_CampbellNorman(T_a = Ta_liz[hr], T_g = Ts[hr], S = Qabs[hr] /
A, a_l = 0.96, epsilon = epsilon_s, c_p = 29.3, D = D, u = V_liz)
}

# Lizard specific biophysical model from Buckley (2008)
Te3 <- rep(NA, 24)
for (hour in 1:24) {
```

```
  Te3[hour] <- Tb_lizard(T_a = Ta_liz[hour], T_g = Ts[hour], u = V_liz, svl =
D * 1000, m = mass, psi = psi_deg[hour], rho_s = rho, elev = elev, doy = doy,
sun = TRUE, surface = TRUE, a_s = a, a_l = 0.965, epsilon_s = epsilon_s, F_d
= 0.8, F_r = 0.5, F_a = 0.5, F_g = 0.5)
}

# General biophysical model from NicheMapR, using modular function ectoR_deve
l()
TC <- unlist(lapply(1:length(Ta_liz), function(x){ectoR_devel(
  Ww_g = mass,
  shape = 3,
  alpha = a,
  M_1 = 0, # turns of metabolism
  pct_wet = 0, # turns of cutaneous evap
  pct_eyes = 0, # turns off ocular evap
  pantmax = 0, # turns of respiration
  alpha_sub = 1 - rho,
  elev = elev,
  pct_cond = psa_g * 100,
  epsilon = epsilon_s,
  epsilon_sub = 1,
  epsilon_sky = 1,
  postur = 0,
  pres = ap,
  K_sub = K,
  fatosk = 0.5,
  fatosb = 0.5,
  TA = Ta_liz[x],
  TGRD = Ts[x],
  TSKY = 1.22 * Ta_liz[x] - 20.4,
  VEL = V_liz,
  RH = 10,
  QSOLR = (Sdir + Sdif)[x],
  PDIF = Sdif[x] / (Sdir + Sdif)[x],
  Z = psi_deg[x]
)$TC})) # run ectotherm_simple across environments
```

We then recreate the plot in the manuscript, adding output from the NicheMapR biophysical model.

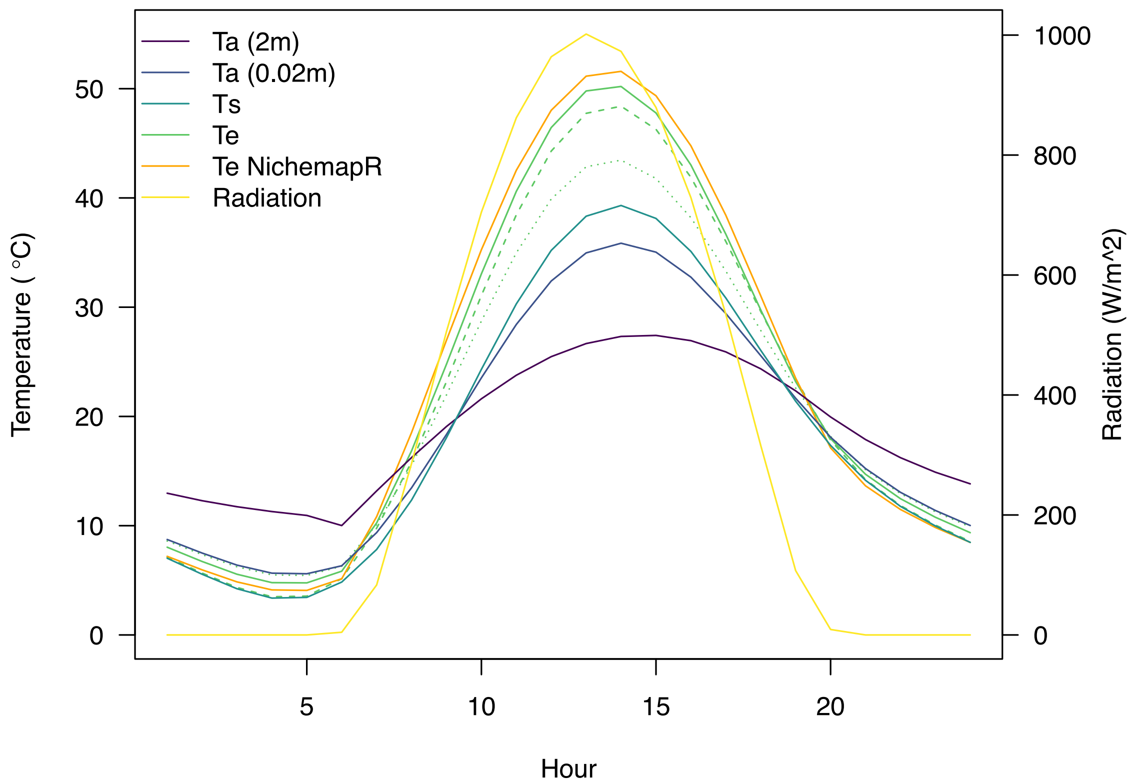


Figure S1.1. Body temperatures (Te) are predicted to drastically exceed air temperature when lizards are exposed to high levels of solar radiation. Air temperatures (Ta, °C) at lizard height (0.02 m) are predicted to exceed air temperatures at 2 m and to be similar to surface temperatures (Ts). We estimate body temperatures using two general energy budgets [solid: Tb_Gates(); dotted: Tb_CampbellNorman()] and a lizard specific biophysical model [dashed: Tb_lizard()] that differ in how they model heat exchanges. The orange line is NichemapR output.