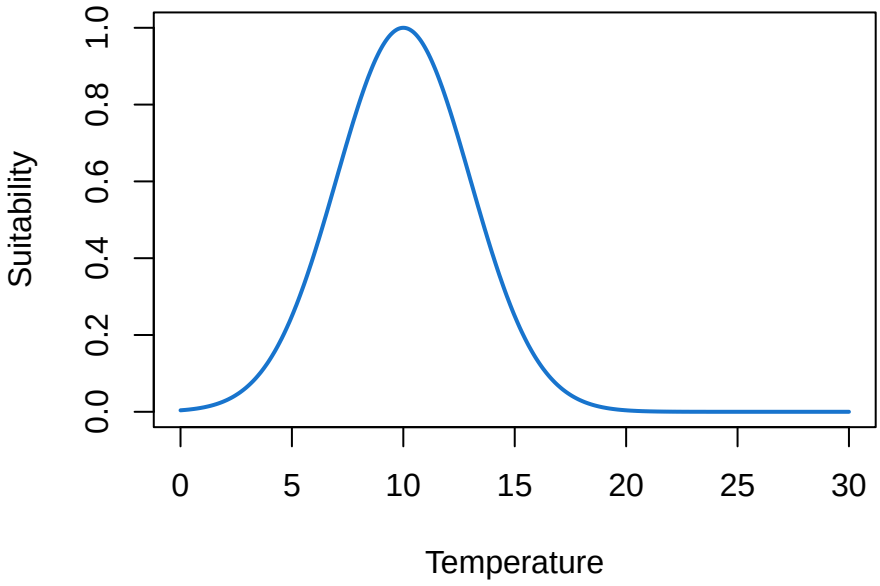
Theory of ENM/SDM

Emilio Berti

Ecological Niche Modeling (ENM)

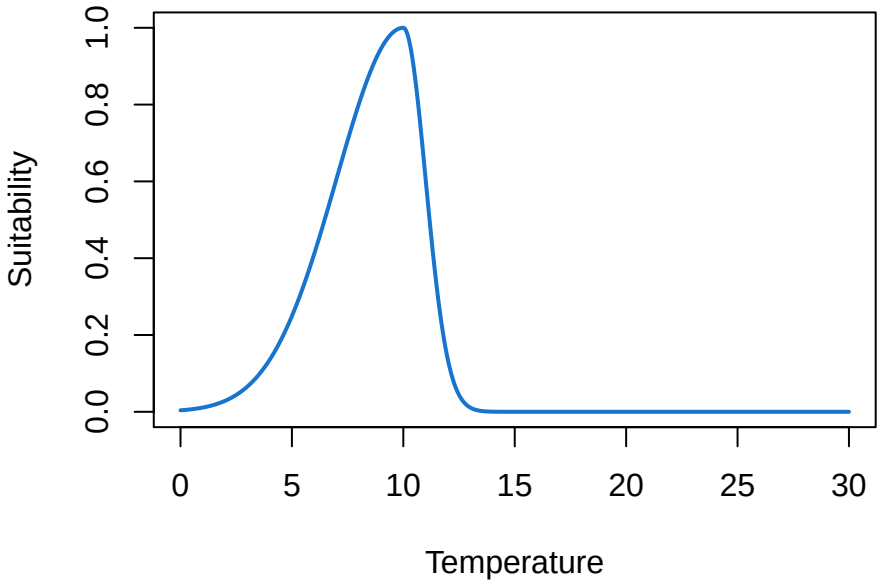
ENM relates one or more environmental variables, e.g. temperature and precipitation, to the suitability of species. The range and shape of how this suitability changes with the variables is called the *environmental niche* of the species. For example, the figure below shows the thermal niche of a species.

```
l <- function(x, mu, sigl, sigr) {
  x[is.na(x)] <- -9999
  out <- rep(NA, length(x))
  out[x < mu] <- exp(- ((x[x < mu] - mu) / sqrt(2) / sigl) ^ 2)
  out[x >= mu] <- exp(- ((x[x >= mu] - mu) / sqrt(2) / sigr) ^ 2)
  out[x == -9999] <- NA
  return(out)
}
x <- seq(0, 30, length.out = 1e3)
plot(
  x, l(x, 10, 3, 3),
  type = "l", lwd = 2, col = "dodgerblue3",
  xlab = "Temperature",
  ylab = "Suitability"
)
```



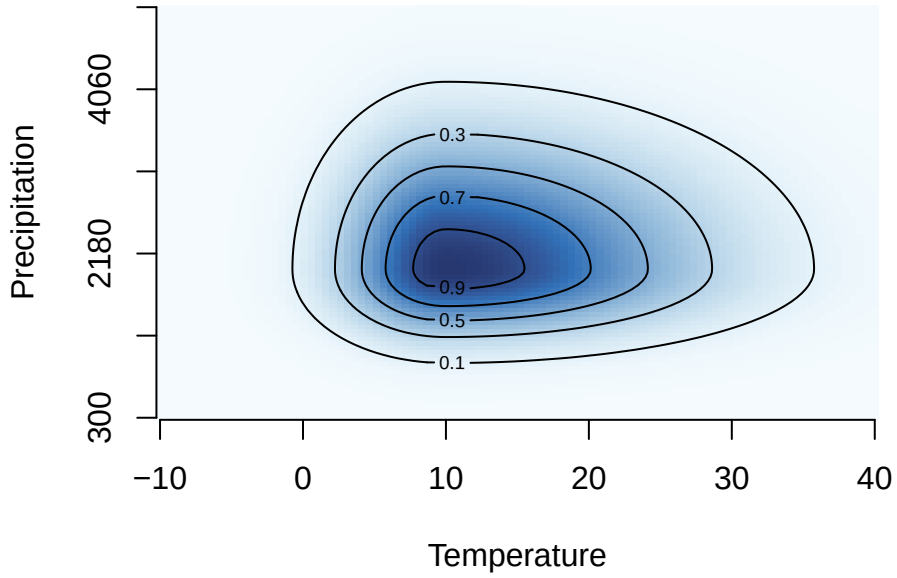
Theory suggests that the niche must be convex, i.e. that the suitability between two points along an environmental gradient must be larger or equal to the value of the points. The curve shown above is an example of a convex niche. The niche is usually also not symmetrical, as biological processes are affected differently by extreme cold and heat. An example of a convex asymmetric niche is shown below.

```
plot(
  x, l(x, 10, 3, 1),
  type = "l", lwd = 2, col = "dodgerblue3",
  xlab = "Temperature",
  ylab = "Suitability"
)
```



Species usually respond to more than one environmental covariate, making the niche multidimensional. An example of a non-symmetrical convex 2D niche is shown below.

```
x <- seq(-10, 40, length.out = 1e2)
y <- seq(300, 5000, length.out = 1e2)
z <- matrix(NA, nrow = length(unique(x)), ncol = length(unique(y)))
for (X in unique(x)) {
  for (Y in unique(y)) {
    i <- which(unique(x) == X)
    j <- which(unique(y) == Y)
    z[i, j] <- l(X, 10, 5, 12) * l(Y, 2000, 500, 1e3)
  }
}
image(z, axes = FALSE, col = hcl.colors(100, "Blues", rev = TRUE))
contour(z, add = TRUE, levels = seq(0.1, 1, by = 0.2))
axis(1, at = seq(0, 1, length.out = 6), labels = seq(-10, 40, length.out = 6))
axis(2, at = seq(0, 1, length.out = 6), labels = seq(300, 5000, length.out = 6))
title(xlab = "Temperature", ylab = "Precipitation")
```



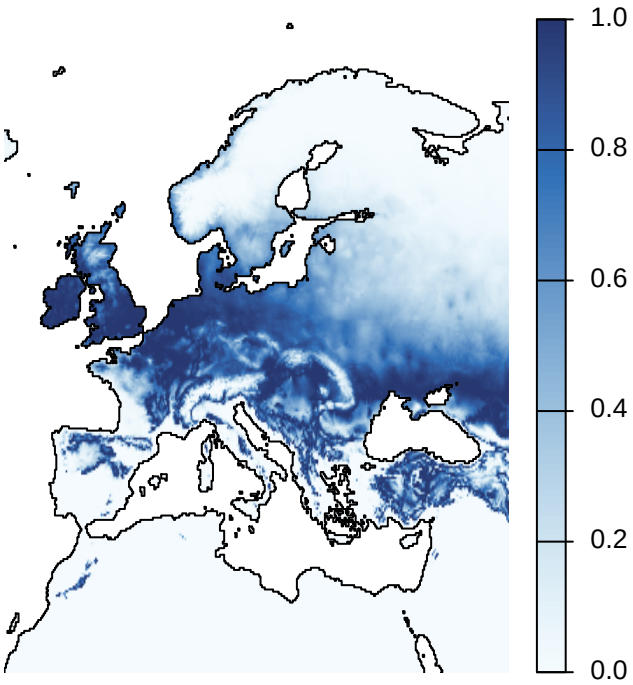
Species Distribution Modeling (SDM)

SDMs project ENMs from an environmental space to a geographical area. Its nuances are mainly in achieving this efficiently using GIS software and highlighting uncertainty of projections due to model extrapolation. For example, projecting the second niche shown in this page onto Europe gives the projected suitability distribution shown below.

```
library(terra)

tavg <- rast("../data/wc_tavg.tif")
niche <- app(tavg, fun = function(x) l(x, 10, 3, 1))
w <- niche
w[!is.na(w)] <- 1
w <- as.polygons(w)

# SDM
plot(
  crop(niche, ext(-15, 45, 25, 75)),
  col = hcl.colors(100, "Blues", rev = TRUE),
  fun = \() lines(w),
  axes = FALSE,
  mar = c(2, 0, 1, 0)
)
```



Generally speaking, the hard part of ENM/SDM is the ENM. Once the niche of inferred, projecting in geographic space is relatively trivial.

Modeling the niche

Ecological Niche Modeling is a general term that applies to many types of models, all of which try to infer the niche of species. In this course, we will focus on two ENM frameworks, one relying on a generalized linear model (GLM) and one on the Random Forest (RF) algorithm [@breiman2001random]. I chose GLM and RF because they represent two extremes of algorithms used in ENM, highlighting their different advantages and limitations. To explain these two procedures, I will use simulated data for a virtual species, *Equus unicornis*.

```
set.seed(123)
tavg <- seq(-20, 50, length.out = 1e2)
prec <- seq(0, 5e3, length.out = 1e2)
suit <- matrix(NA, nrow = length(unique(tavg)), ncol = length(unique(prec)))
for (x in tavg) {
  for (y in prec) {
    i <- which(tavg == x)
    j <- which(prec == y)
    suit[i, j] <- l(x, 10, 5, 12) * l(y, 2000, 500, 1000)
  }
```

```
}
occ <- rbinom(suit, size = 1, prob = suit)
tavg <- rep(tavg, each = 1e2)
prec <- rep(prec, 1e2)
d <- data.frame(tavg, prec, occ)
d <- d[!is.na(d$occ), ]
d <- unique(d)
d <- d[sample(nrow(d), 500), ]
head(d)
```

```
          tavg     prec occ
3971  7.575758 3535.354   0
4884 13.939394 4191.919   0
7881 35.151515 4040.404   0
5156 16.060606 2777.778   1
8238 37.979798 1868.687   0
6962 28.787879 3080.808   0
```

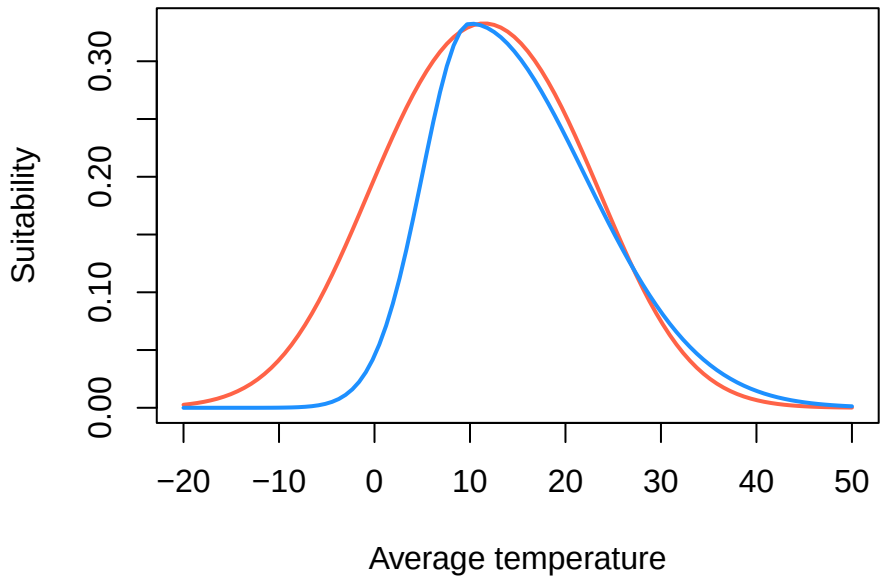
This dataframe has three columns. `tavg` and `prec` are the average temperature and precipitation, respectively. `occ` is the occurrence status, namely if the species has been observed at that environmental conditions (`1`) or not (`0`).

GLM

The simples GLM uses a quadratic curve and a binary (binomial) response. The quadratic curve assures the niche is convex, even though it does not allow asymmetric curves. In R, this is achieved using the following specifications.

```
enm <- glm(occ ~ tavg + I(tavg^2), data = d, family = "binomial")
plot(
  sort(d$tavg),
  predict(enm, type = "response")[order(d$tavg)],
  type = "l", lwd = 2, col = "tomato",
  xlab = "Average temperature",
  ylab = "Suitability"
)
lines(
  sort(d$tavg),
  1(sort(d$tavg), 10, 5, 12) * max(predict(enm, type = "response")),
  lwd = 2,
```

```
  col = "dodgerblue"
)
```



> **Note**
>
> The syntax `I(tavg^2)` makes R consider the quadratic term explicitly, rather than squaring the values of tavg.

You can see that the actual niche (blue) is not symmetrical, but the inferred niche (red) is. This is a common bias of GLM ENM. However, for its simplicity, a GLM ENM performs reasonably well at capturing the optimal environmental value and the general decline of suitability away from it.

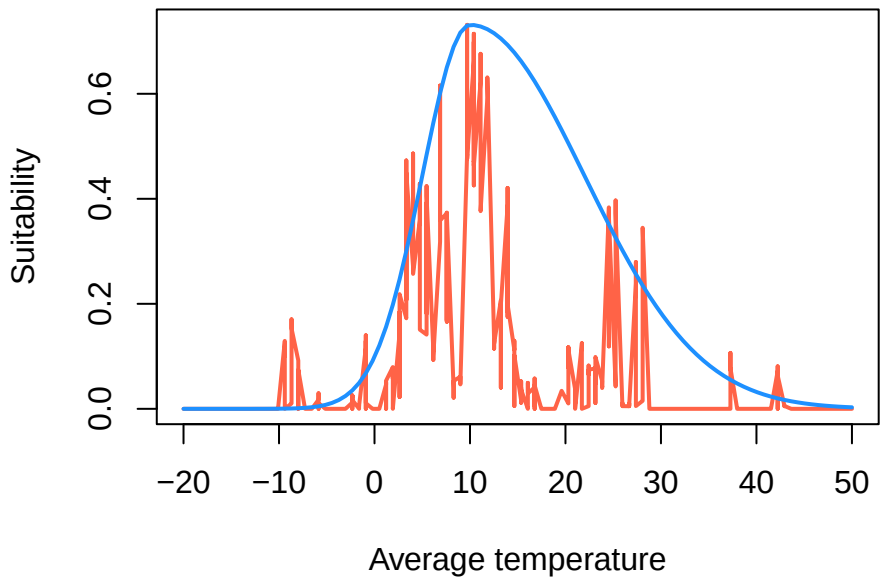
Random Forest (RF)

RF is a powerful machine-learning algorithm that has been applied to ENM for several decades. RF has its advantages, but has a major issue: the niche is rarely convex. In fact, in most cases, the niche will be concave (the opposite of convex). In R, RF is achieved using the package `randomForest` with the following specifications.

```
library(randomForest)

enm <- randomForest(
```

```
  as.factor(occ) ~ tavg,
  nodesize = 20,
  data = d
)
plot(
  sort(d$tavg),
  predict(enm, type = "prob")[order(d$tavg), 2],
  type = "l", lwd = 2, col = "tomato",
  xlab = "Average temperature",
  ylab = "Suitability"
)
lines(
  sort(d$tavg),
  1(sort(d$tavg), 10, 5, 12) * max(predict(enm, type = "prob")[order(d$tavg), 2]),
  lwd = 2,
  col = "dodgerblue"
)
```



> **Note**
>
> The syntax `as.factor(occ)` makes `randomForest()` to consider a classification, rather than a regression, problem.

You can see that the niche is very jagged. This is due to how RF works internally. We will not explain why this happens in this course, but we will highlight this feature of RF and discuss its

implications several times. However, you can see that the RF model (red) follows, more or less, the shape of the actual niche (blue).