Ecological Niche Models

Sebastian Fernandez
University of Florida

FLORIDA MUSEUM™ iDigBio Integrated Digitized Biocollections NSF UF UNIVERSITY of FLORIDA

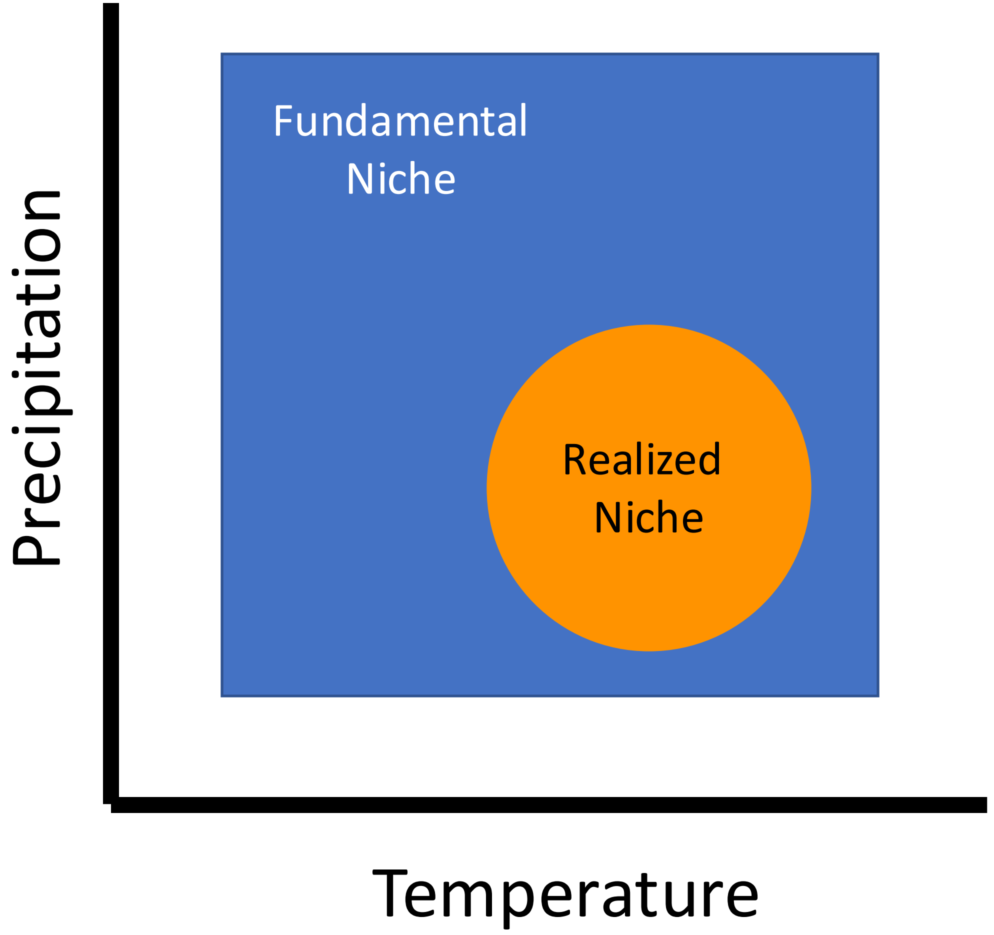
Ecological Niche

Fundamental Niche

- abiotic conditions a species could potentially occupy in the absence of biotic interactions

Realized Niche

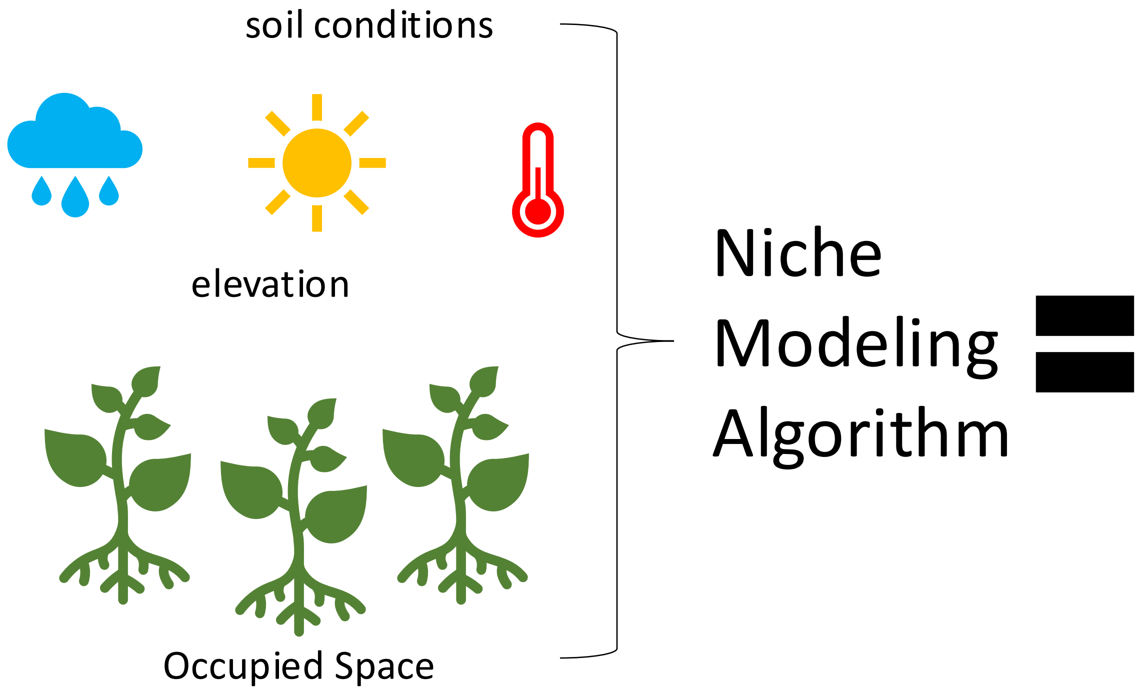
- abiotic conditions that a species can occupy with the presence of biotic interactions

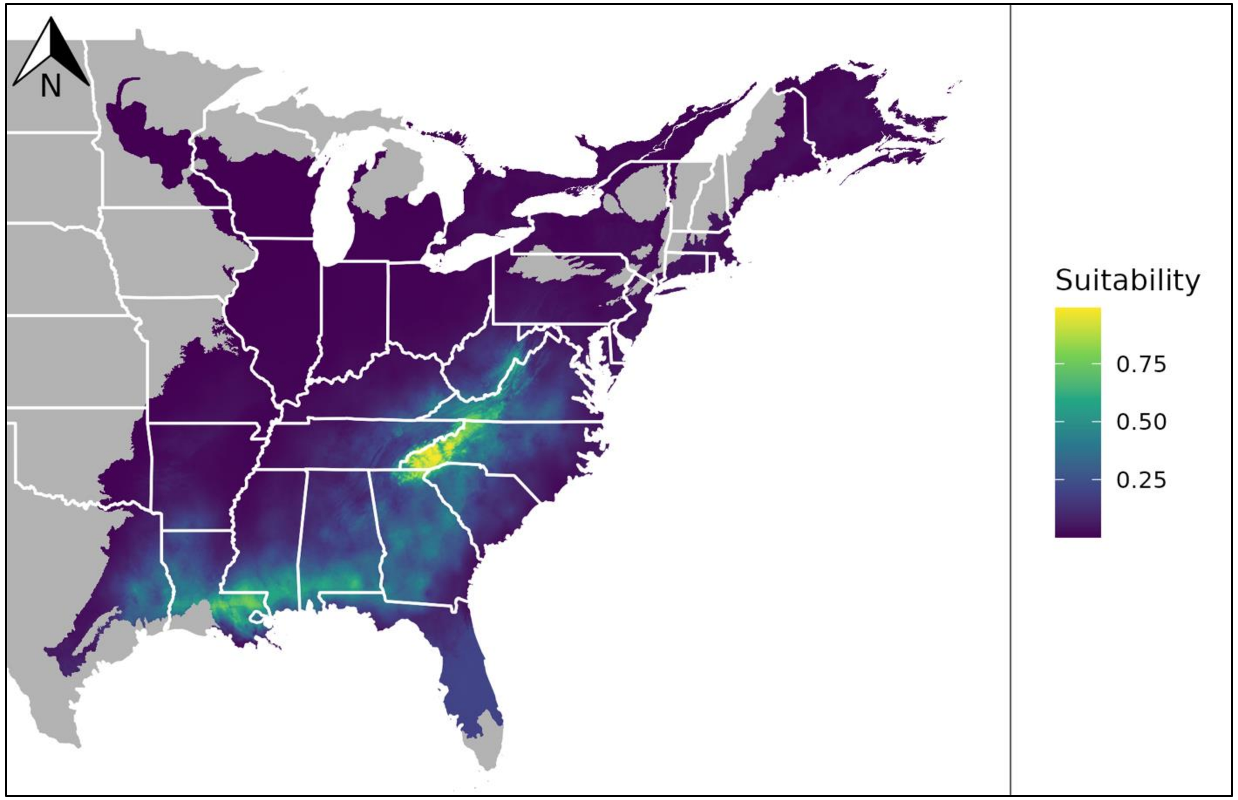


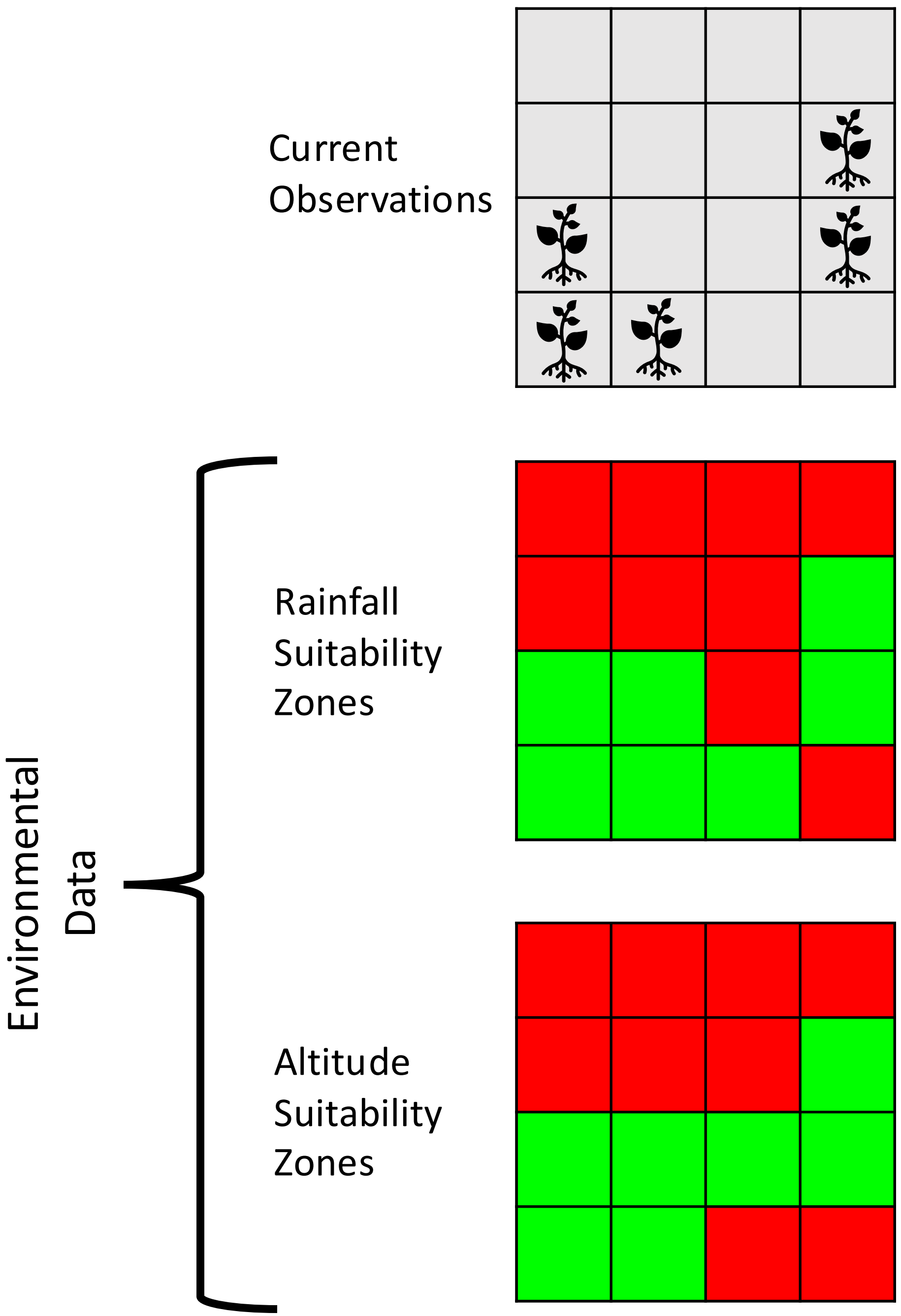
Ecological Niche Modeling

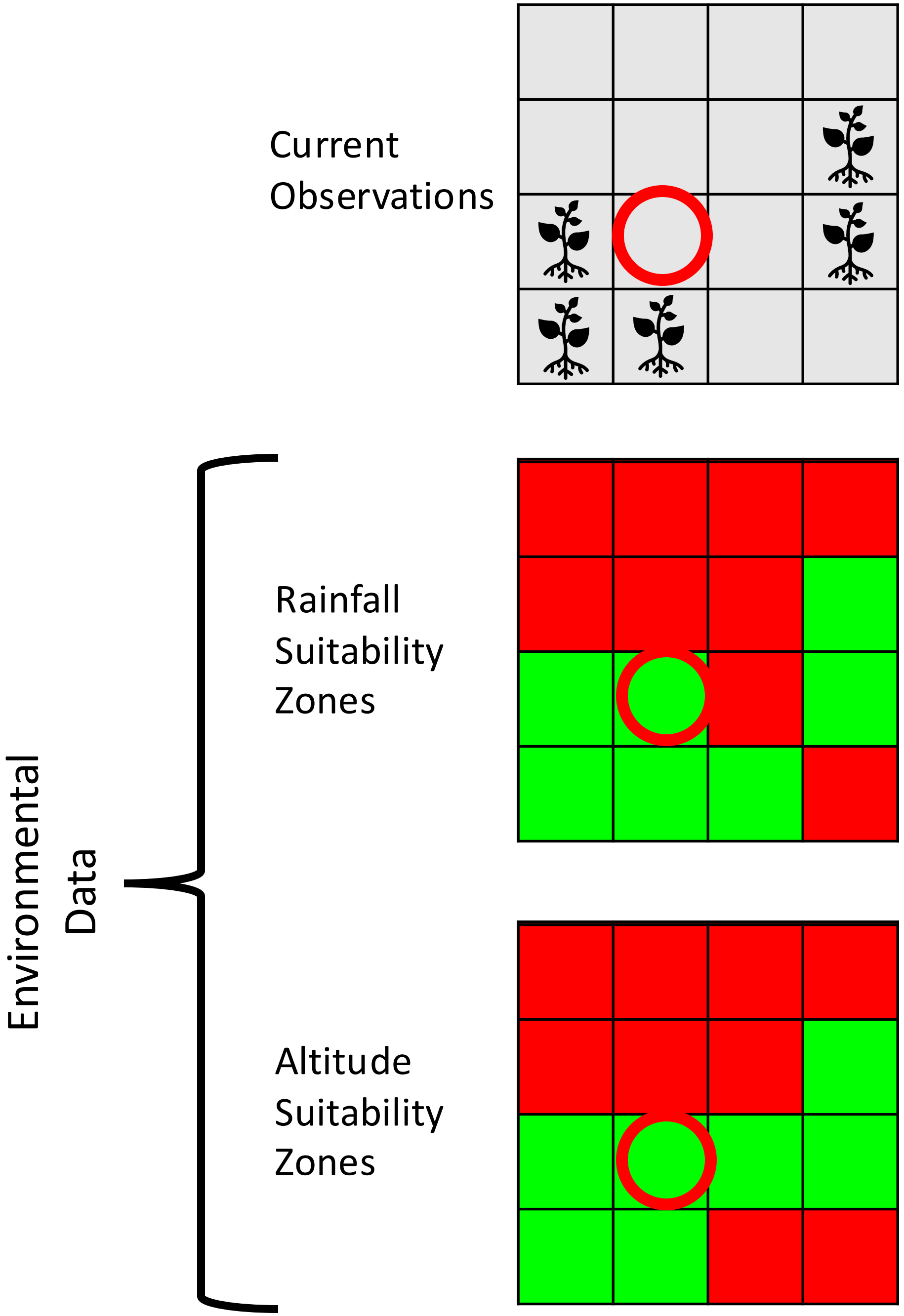
Fundamental Niche

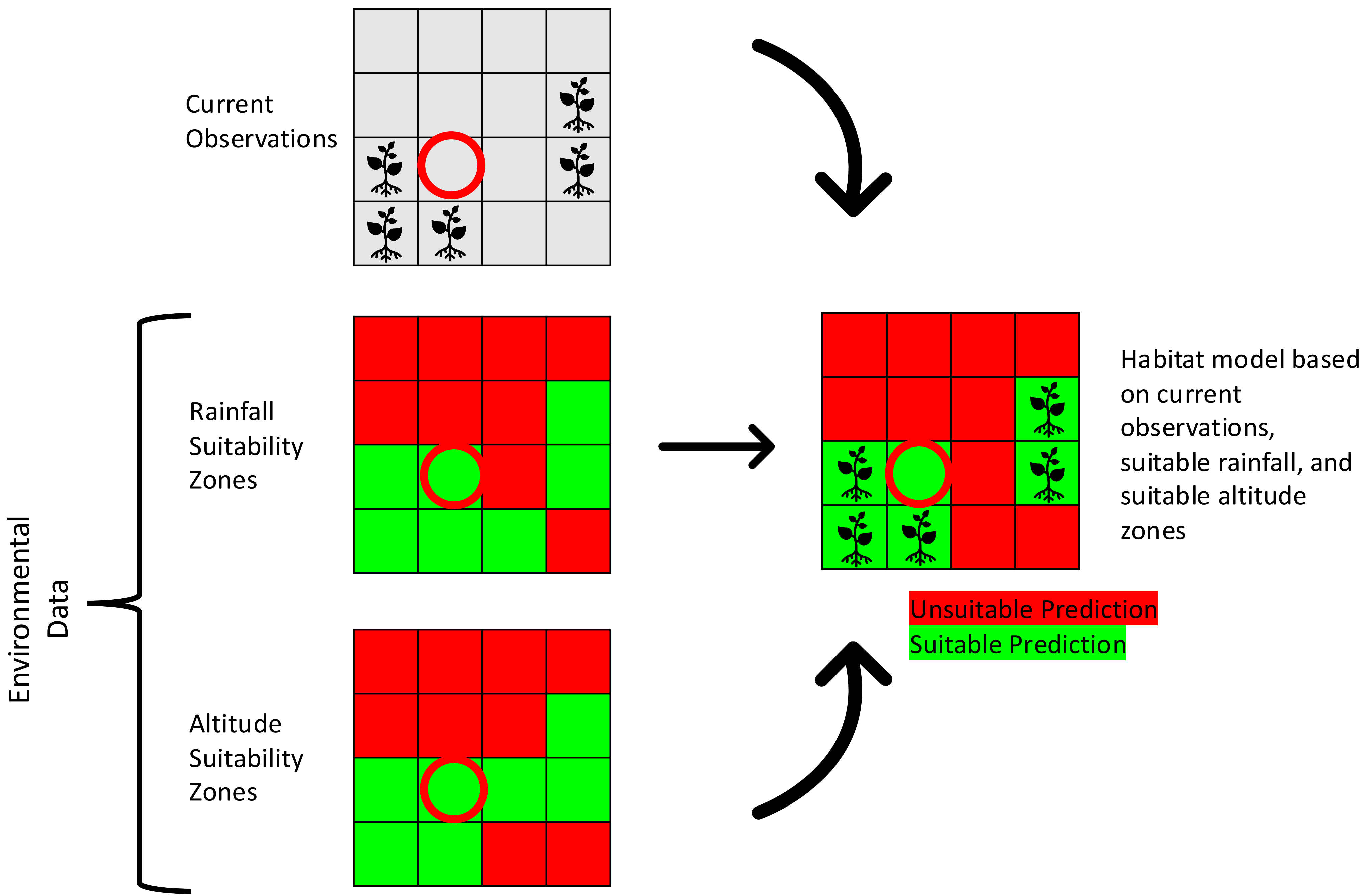
- abiotic conditions a species could potentially occupy in the absence of biotic interactions






Current
Observations
Rainfall
Suitability
Zones
Altitude
Suitability
Zones
Environmental
Data

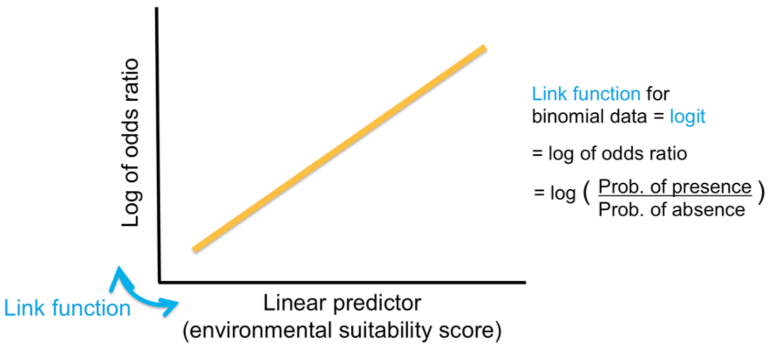

Current Observations
Rainfall Suitability Zones
Altitude Suitability Zones
Environmental Data



Current Observations
Environmental Data
Rainfall Suitability Zones
Altitude Suitability Zones
Habitat model based on current observations, suitable rainfall, and suitable altitude zones
Unsuitable Prediction
Suitable Prediction

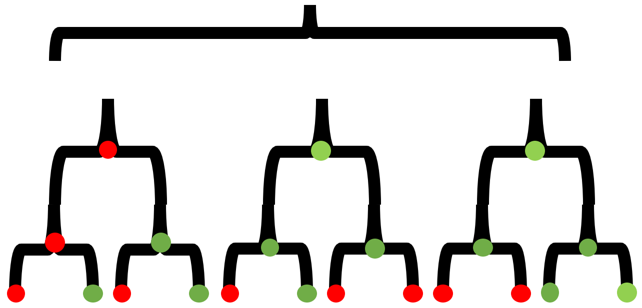
Niche Modeling Algorithms

Generalized Linear Model



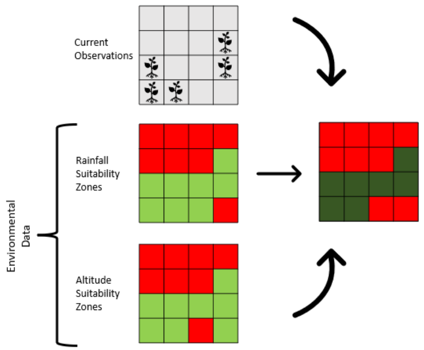
Requires absence data

Random Forest Analysis



Requires absence data

Maxent



Does not require absence data

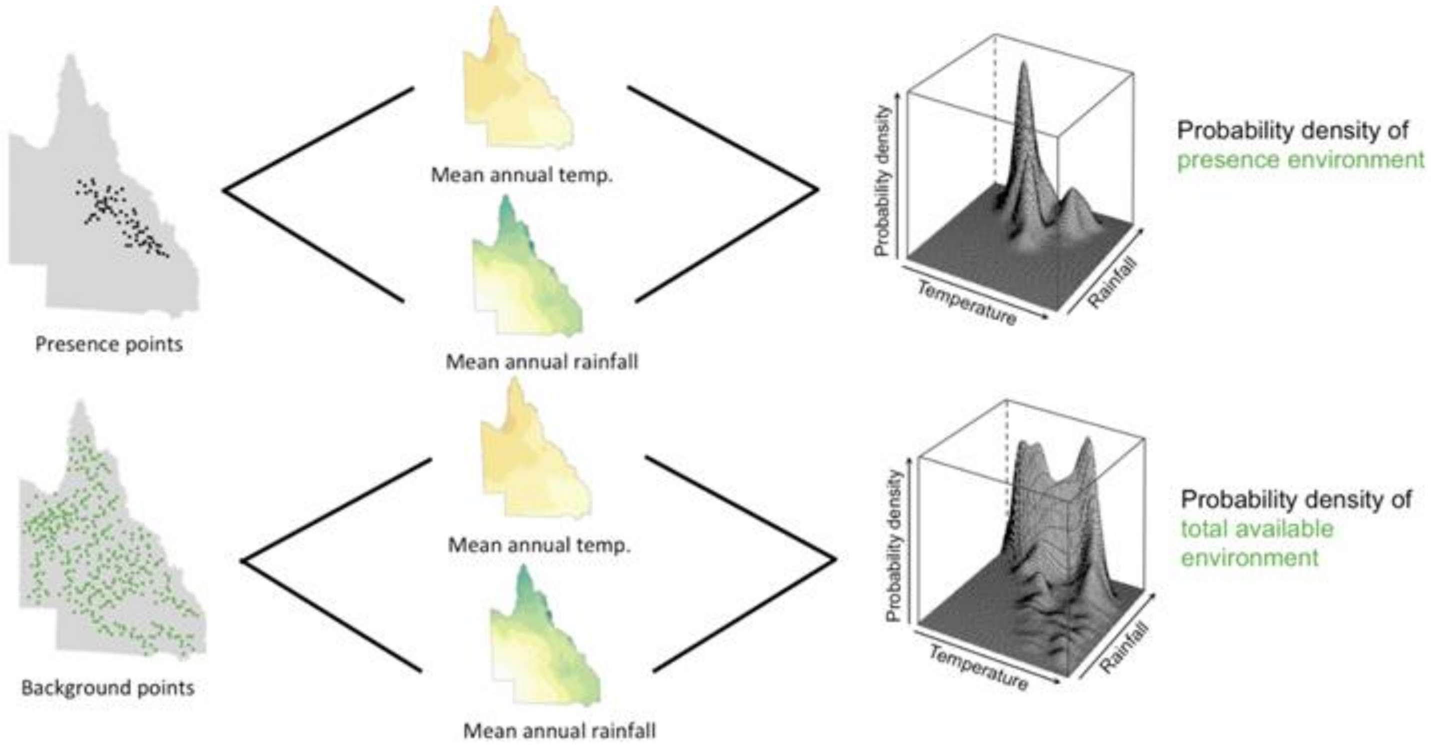
Generalized linear model (GLM) SDM explained. Biosecurity Commons Support Portal. (2023). https://support.biosecuritycommons.org.au/support/solutions/articles/6000262275-generalized-linear-model-glm-sdm-explained

Background Points

Maxent only uses presence data and the algorithm compares the locations of where a species has been found to all the environments that are available in the study region.

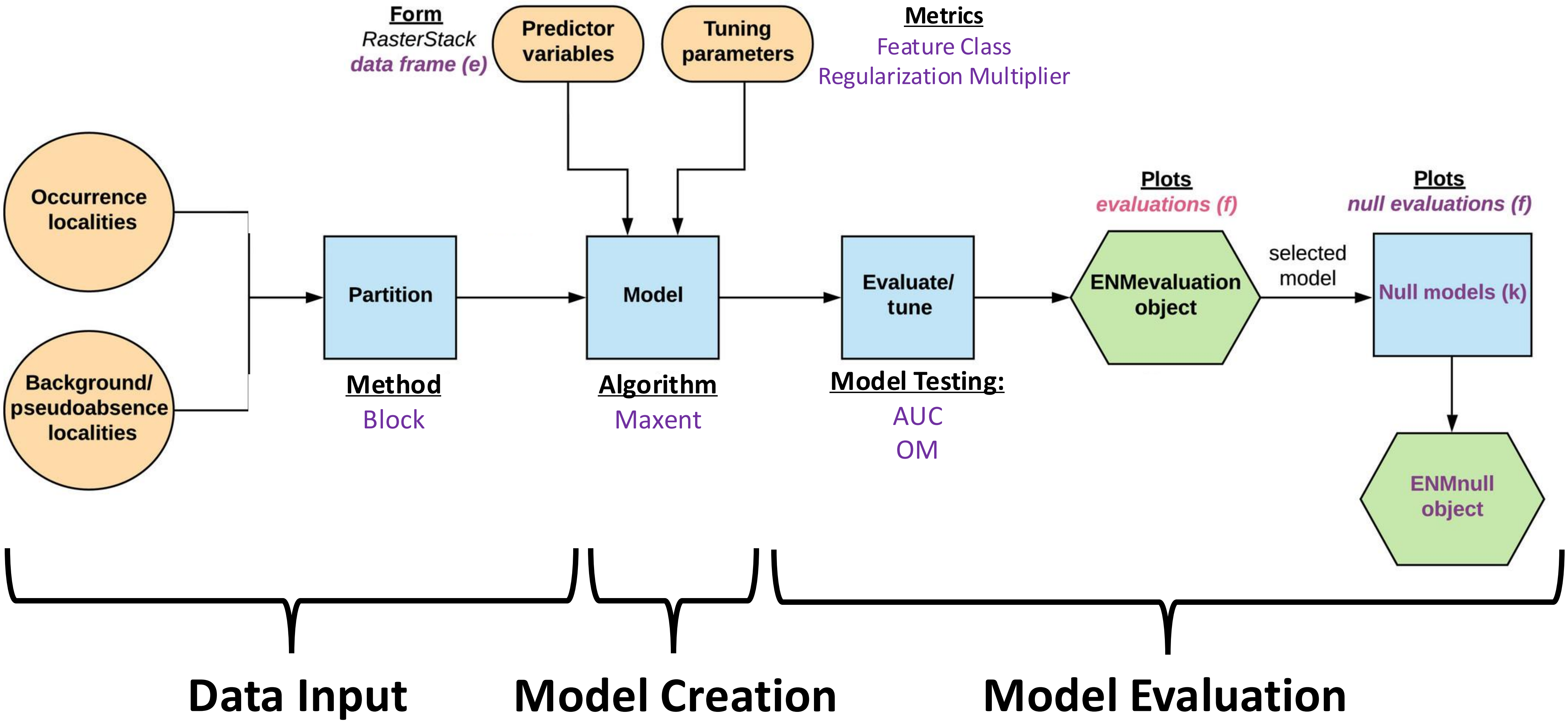
- Most niche modeling algorithms use presence data to weigh the influence of environmental conditions
- Background points define the available environment
- Background points can include areas where species occur
- Maxent then calculates the ratio between these two probability densities, which gives the relative environmental suitability for presence of a species for each point in the study area.

```
eval <- ENMevaluate(
  occs = Galax_urceolata[, c("longitude", "latitude")],
  envs = vifStack,
  tune.args = list(fc = c("L", "Q"), rm = 1:2),
  partitions = "block",
  n.bg = 10000,
  parallel = FALSE,
  algorithm = 'maxent.jar',
)
```



Adapted from Elith et al. (2011) *A statistical explanation of MaxEnt for ecologists*. Diversity and Distributions, 17, 43-57.

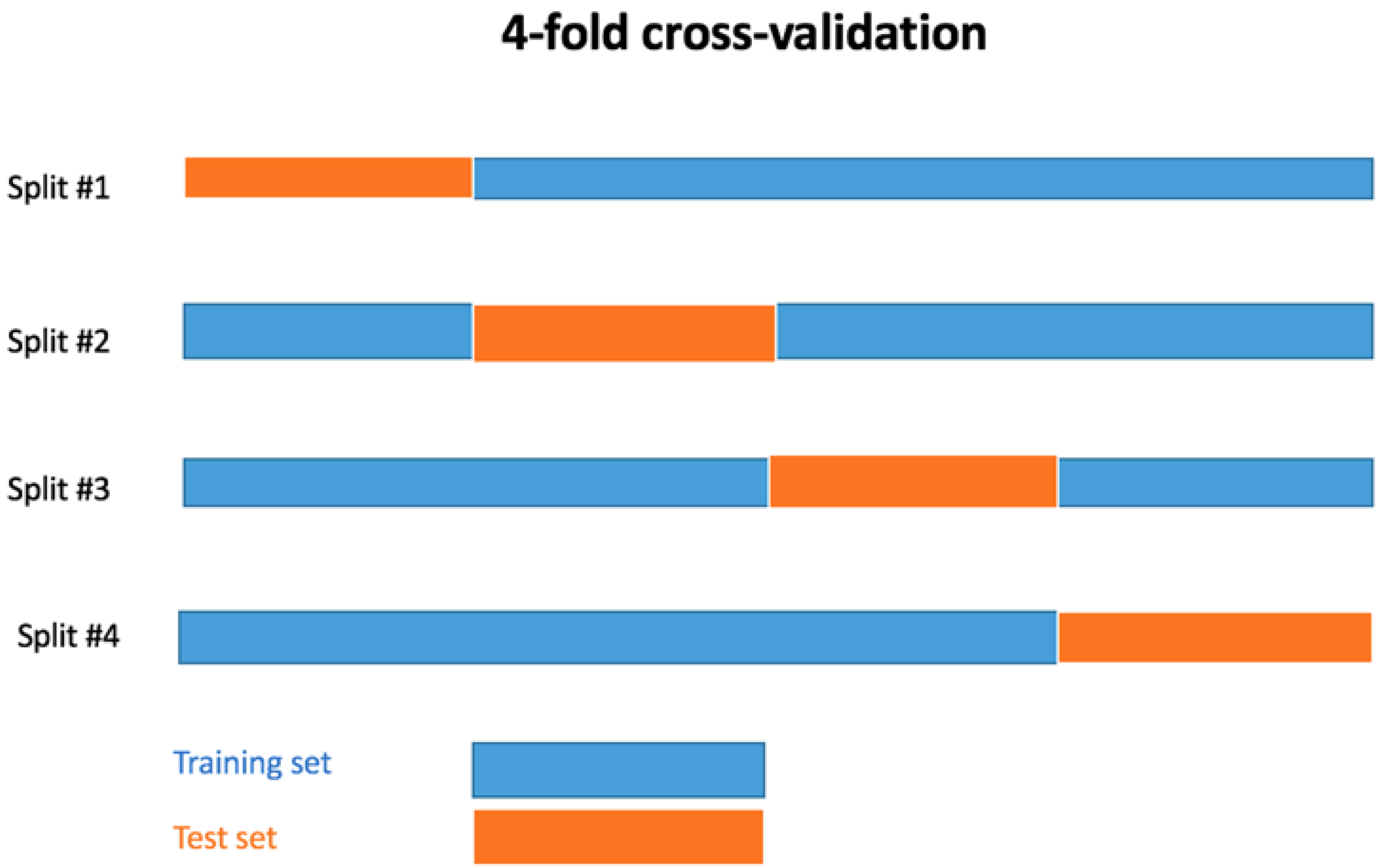
ENMeval



Form
RasterStack
data frame (e)
Predictor variables
Tuning parameters
Metrics
Feature Class
Regularization Multiplier
Occurrence localities
Background/ pseudoabsence localities
Partition
Method
Block
Model
Algorithm
Maxent
Evaluate/ tune
Model Testing:
AUC
OM
Plots
evaluations (f)
ENMevaluation object
selected model
Plots
null evaluations (f)
Null models (k)
ENMnull object
Data Input
Model Creation
Model Evaluation

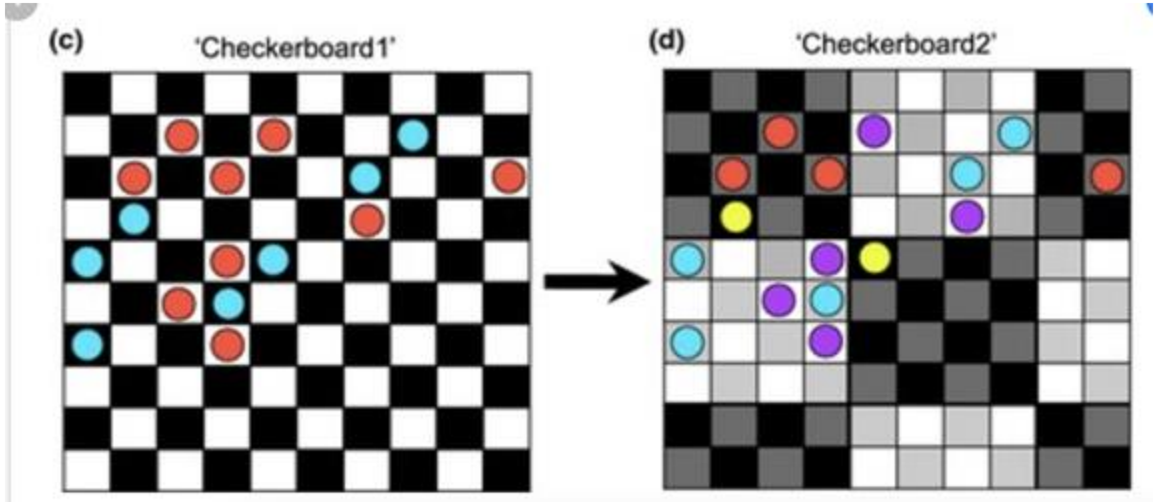
ENMeval: Partitions

- Cross validation
    - a resampling procedure used to evaluate machine learning models on a limited data sample.
- Model trained on three blocks and tested on the fourth
- Process rotates between blocks



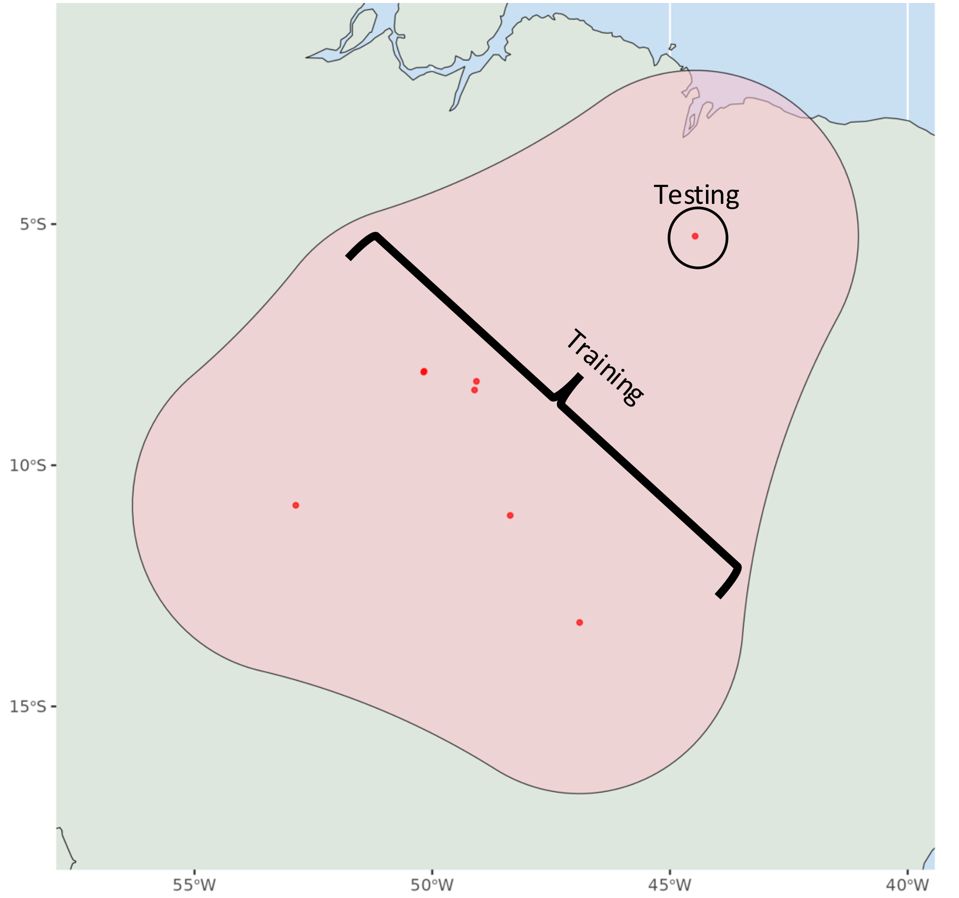

```
eval <- ENMevaluate(
  occs = Galax_urceolata[, c("longitude", "latitude")],
  envs = vifStack,
  tune.args = list(fc = c("L", "Q"), rm = 1:2),
  partitions = "block",
  n.bg = 10000,
  parallel = FALSE,
  algorithm = 'maxent.jar',
)
```

ENMeval: Partitions

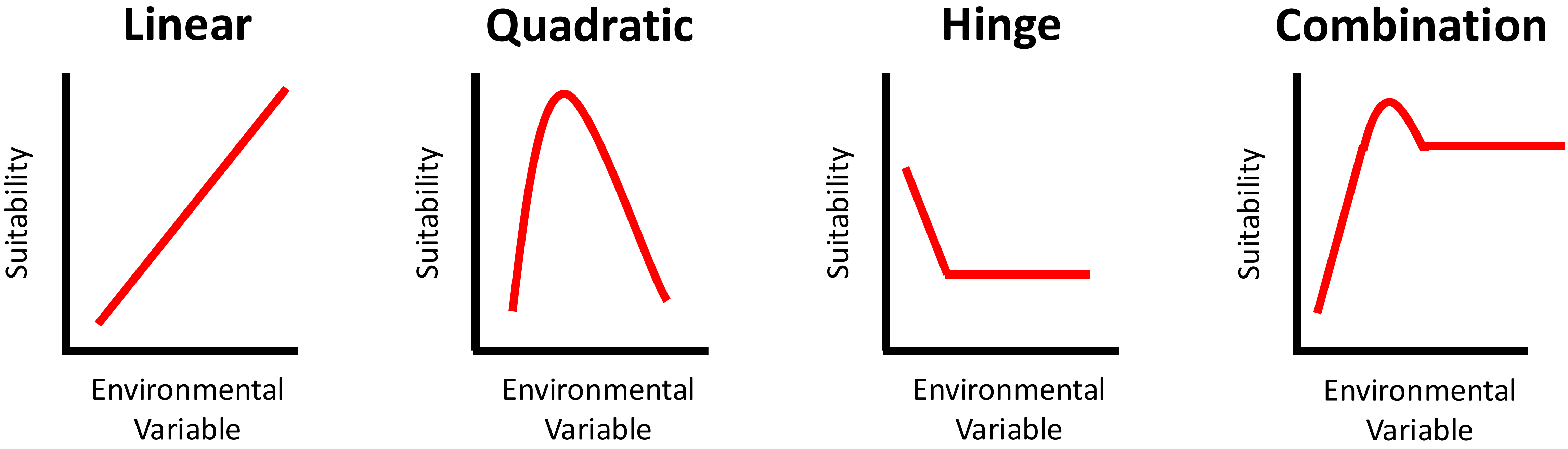



```
eval <- ENMevaluate(
  occs = Galax_urceolata[, c("longitude", "latitude")],
  envs = vifStack,
  tune.args = list(fc = c("L", "Q"), rm = 1:2),
  partitions = "block",
  n.bg = 10000,
  parallel = FALSE,
  algorithm = 'maxent.jar',
)
```

Jackknife



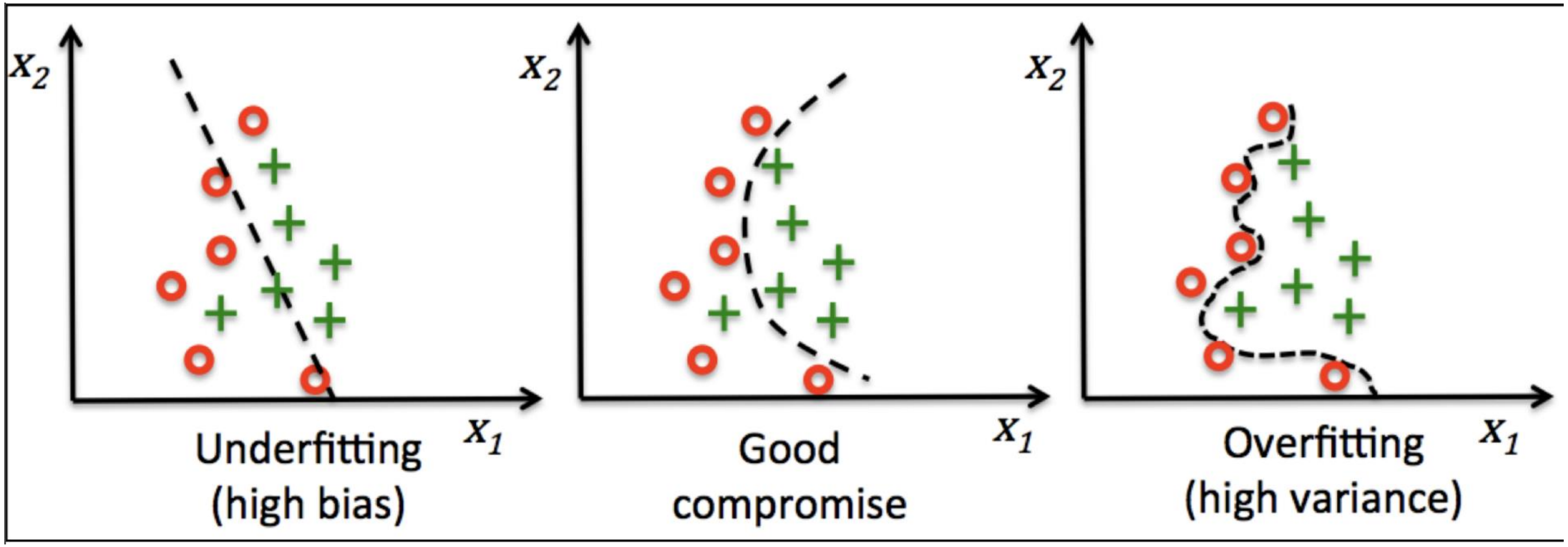
ENMeval: Feature Classes




```r
eval <- ENMevaluate(
  occs = Galax_urceolata[, c("longitude", "latitude")],
  envs = vifStack,
  tune.args = list(fc = c("L", "Q"), rm = 1:2),
  partitions = "block",
  n.bg = 10000,
  parallel = FALSE,
  algorithm = 'maxent.jar',
)
```

ENMeval: Regularization Multiplier

Complexity Penalization



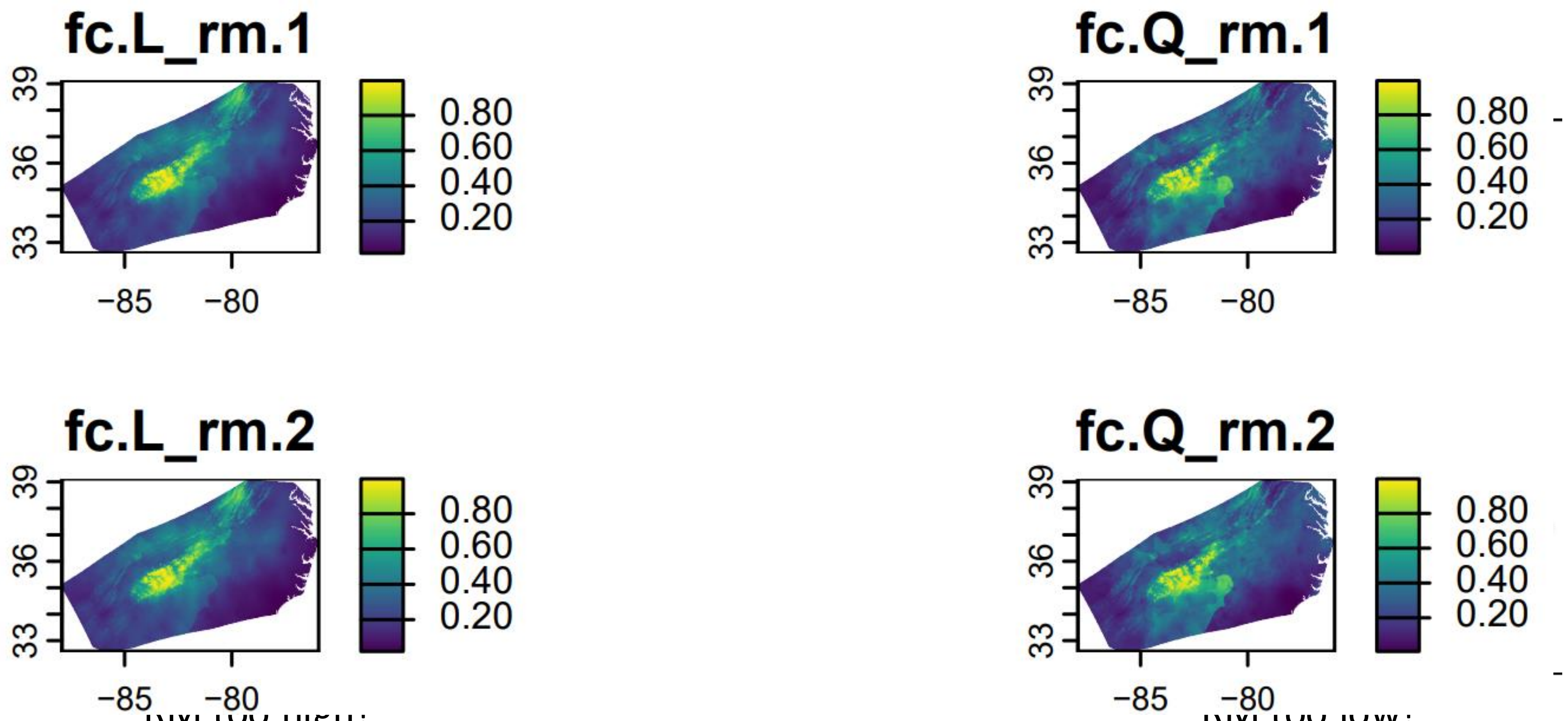
RM too high!

RM too low!

```
eval <- ENMevaluate(
  occs = Galax_urceolata[, c("longitude", "latitude")],
  envs = vifStack,
  tune.args = list(fc = c("L", "Q"), rm = 1:2),
  partitions = "block",
  n.bg = 10000,
  parallel = FALSE,
  algorithm = 'maxent.jar',
)
```

ENMeval: Models

Complexity




```
eval <- ENMevaluate(
  occs = Galax_urceolata[, c("longitude", "latitude")],
  envs = vifStack,
  tune.args = list(fc = c("L", "Q"), rm = 1:2),
  partitions = "block",
  n.bg = 10000,
  parallel = FALSE,
  algorithm = 'maxent.jar',
)
```

ENMeval: Model Evaluation

Akaike Information Criterion

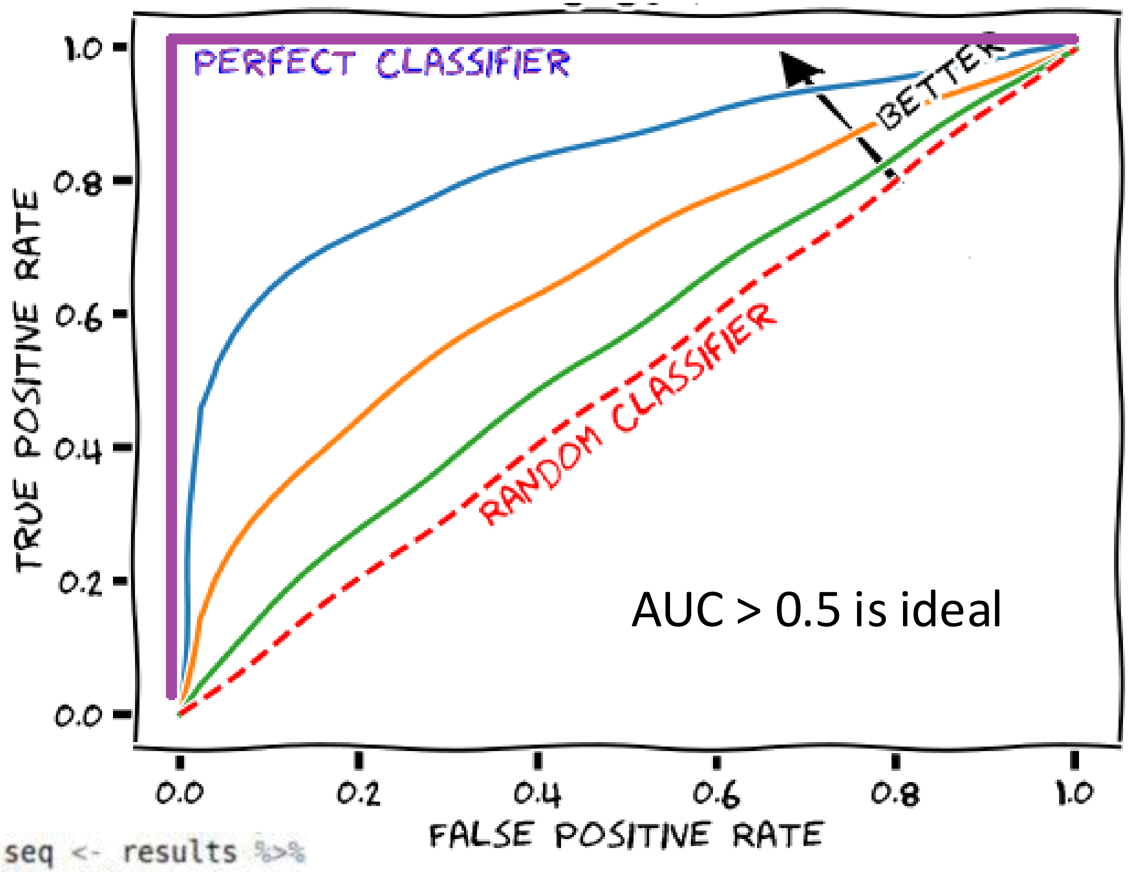
$$AIC = -2\ln(\underbrace{L}_{\text{Maximum likelihood of model}}) + 2\underbrace{k}_{\text{\# of feature coefficients}}$$

```
opt.seq <- results %>%
  filter(!is.na(AICc)) %>%
  filter(AICc == min(AICc)) %>%
  filter(or.10p.avg != 0) %>%
  filter(or.10p.avg == min(or.10p.avg)) %>%
  filter(auc.val.avg == max(auc.val.avg))
```

Low AIC = Better Model Fit + Less Overfitting

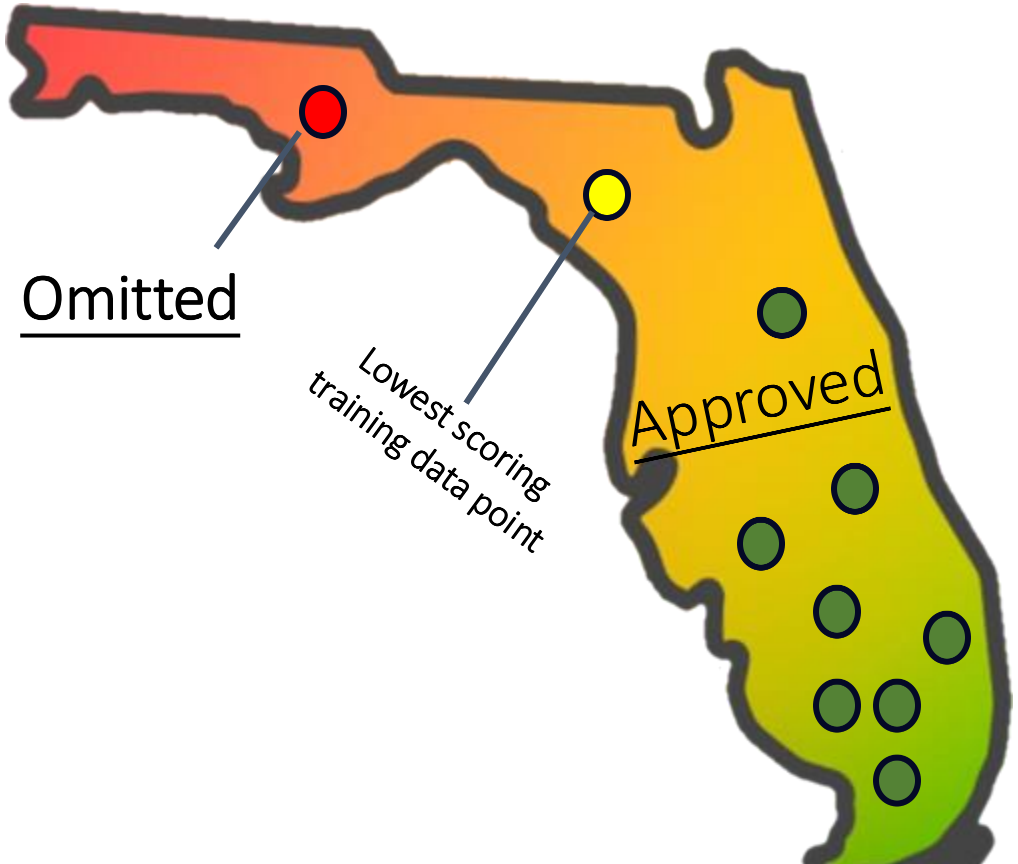
ENMeval: Model Evaluation

Area Under the Curve (AUC)




```
opt.seq <- results %>%
  filter(!is.na(AICc)) %>%
  filter(AICc == min(AICc)) %>%
  filter(or.10p.avg != 0) %>%
  filter(or.10p.avg == min(or.10p.avg)) %>%
  filter(auc.val.avg == max(auc.val.avg))
```

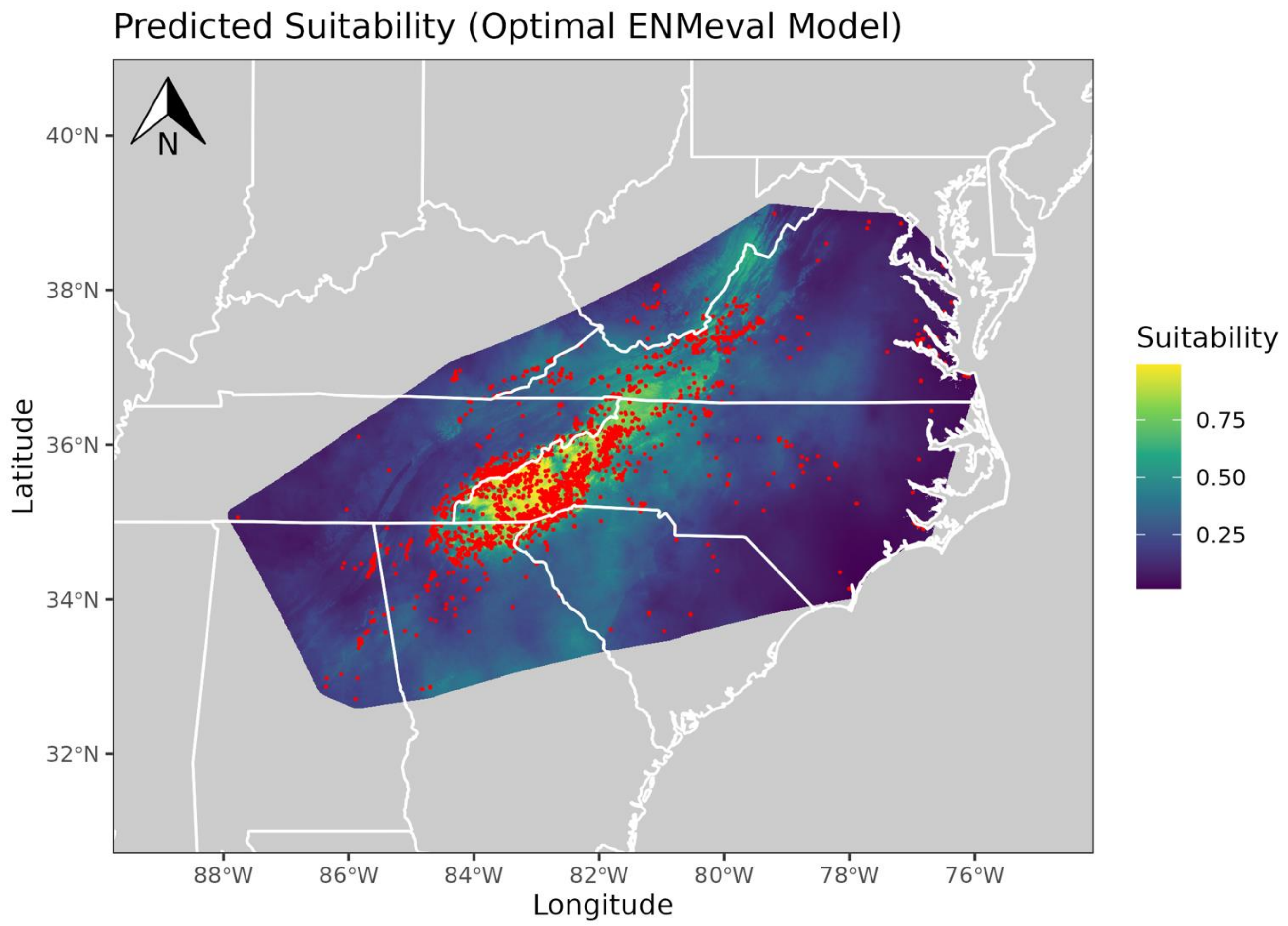
Omission Rate (OM)



OM close to 0 is ideal
OM = 0 is an overfit model

ENMeval: Optimal Model

Lowest AIC
Lowest OM
Highest AUC



ENMeval: Optimal Model