EVALUATING ENM PERFORMANCE WITH NULL MODELS

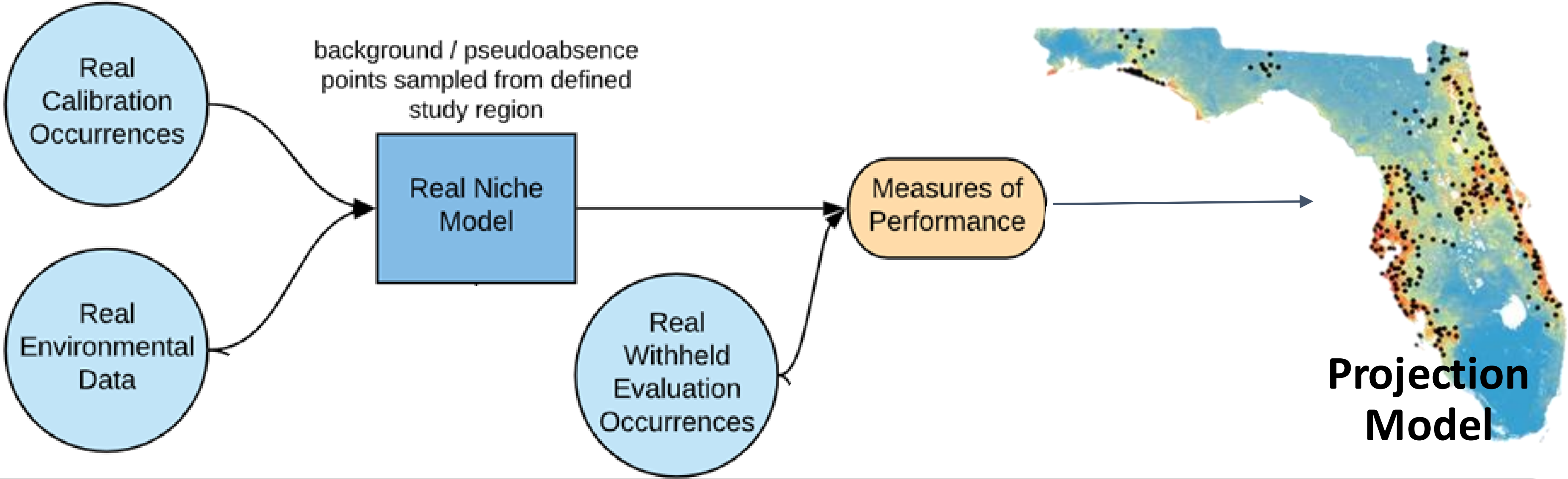
Tyler Radtke
University of Florida

FLORIDA MUSEUM

iDigBio Integrated Digitized Biocollections

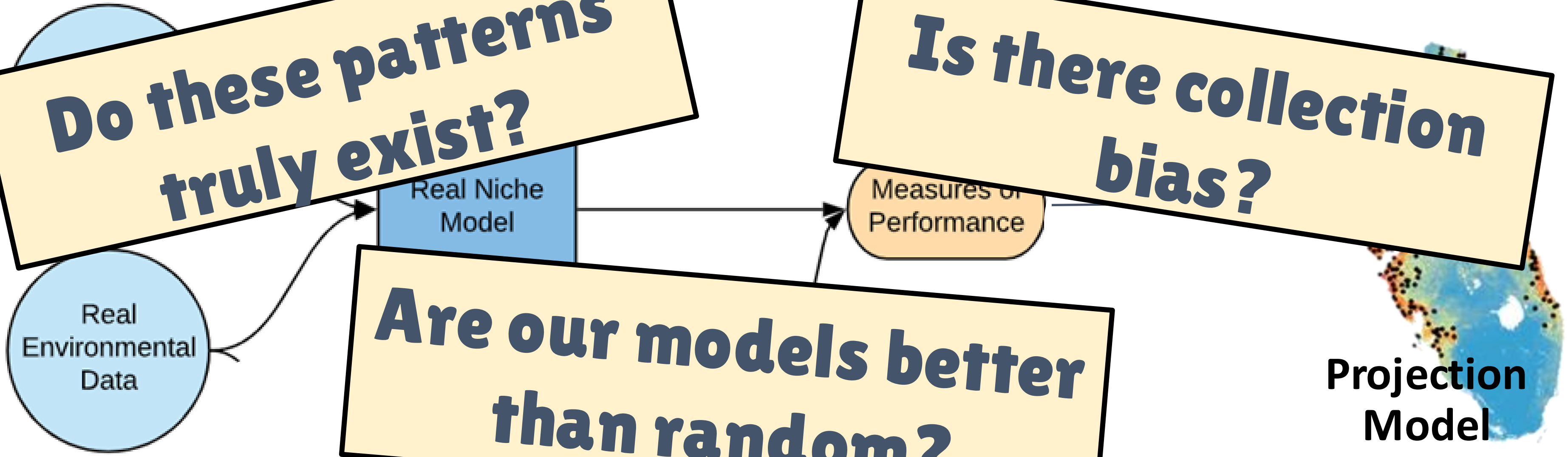
NSF

UF UNIVERSITY of FLORIDA



The Typical ENM Process

Bohl, C. L., Kass, J. M., & Anderson, R. P. (2019). A new null model approach to quantify performance and significance for ecological niche models of species distributions. *Journal of Biogeography*, *46*(6), 1101–1111. https://doi.org/10.1111/jbi.13573



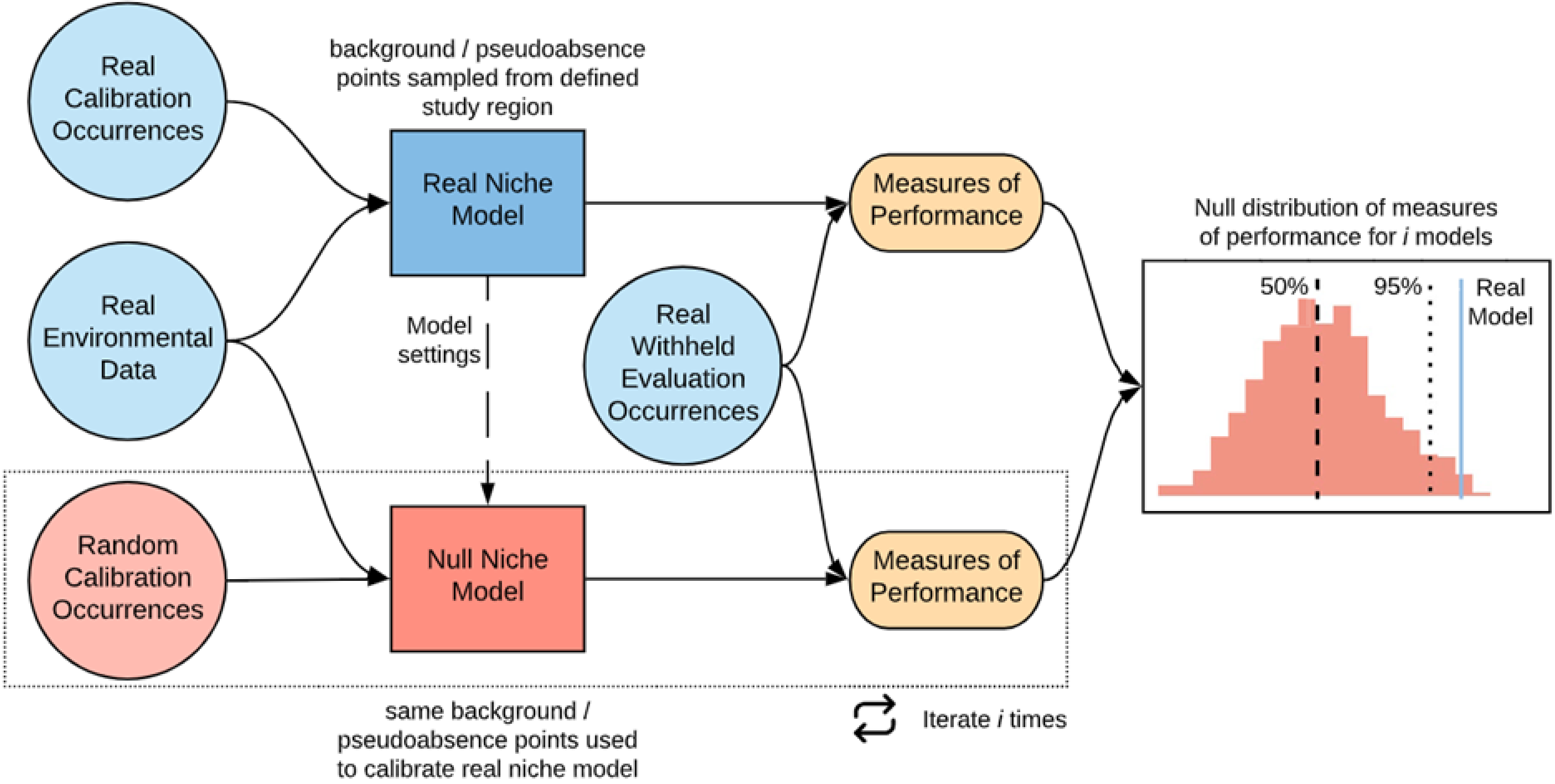
Do these patterns truly exist?

Is there collection bias?

Are our models better than random?

The Typical ENM Process

Bohl, C. L., Kass, J. M., & Anderson, R. P. (2019). A new null model approach to quantify performance and significance for ecological niche models of species distributions. *Journal of Biogeography*, *46*(6), 1101–1111. https://doi.org/10.1111/jbi.13573



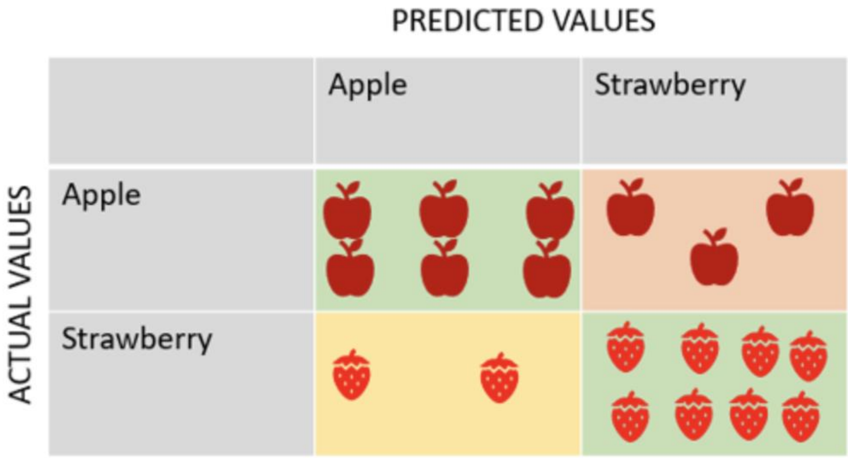
Bohl, C. L., Kass, J. M., & Anderson, R. P. (2019). A new null model approach to quantify performance and significance for ecological niche models of species distributions. *Journal of Biogeography*, *46*(6), 1101–1111. https://doi.org/10.1111/jbi.13573

Actual vs. predicted habitat suitability values

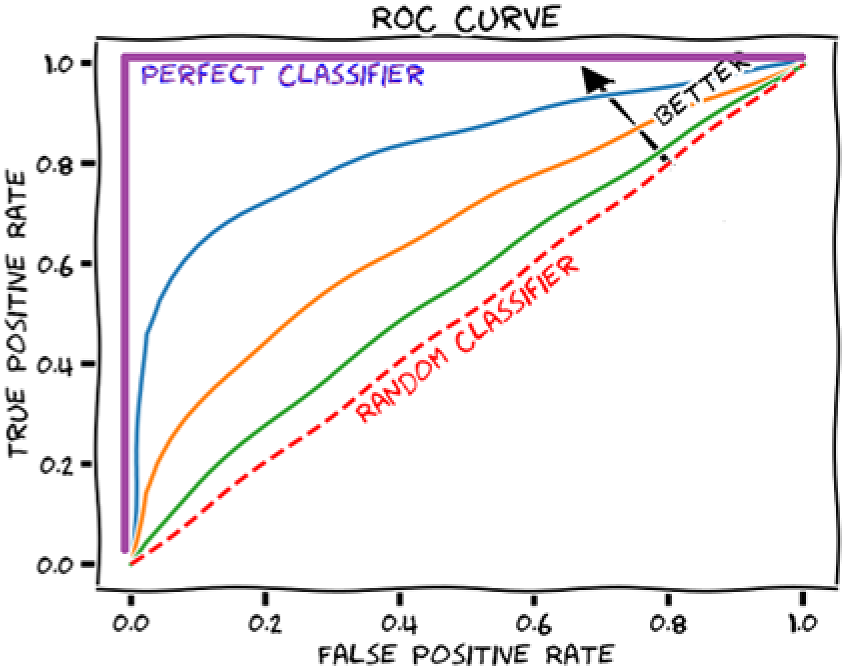
PREDICTED VALUES

ACTUAL VALUES

	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)



Area Under the Curve (AUC)



False positive rate (specificity) vs. true positive rate (sensitivity)

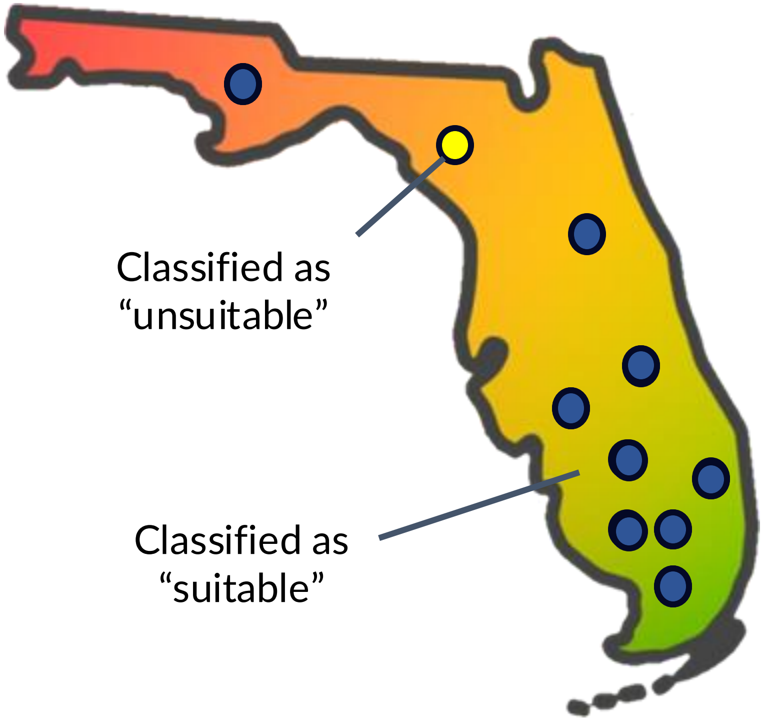
AUC = 1.0 -> perfect model
AUC = 0.5 -> model performs no better than random
AUC = <0.5 -> a sign of error

AUC evaluates a model's ability to distinguish between presence/background points

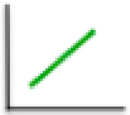
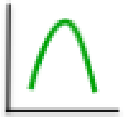
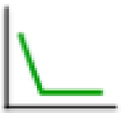
Omission Rate (OR)

OR = the proportion of known presence points that the model fails to predict as suitable

Example:
OR = 0.1

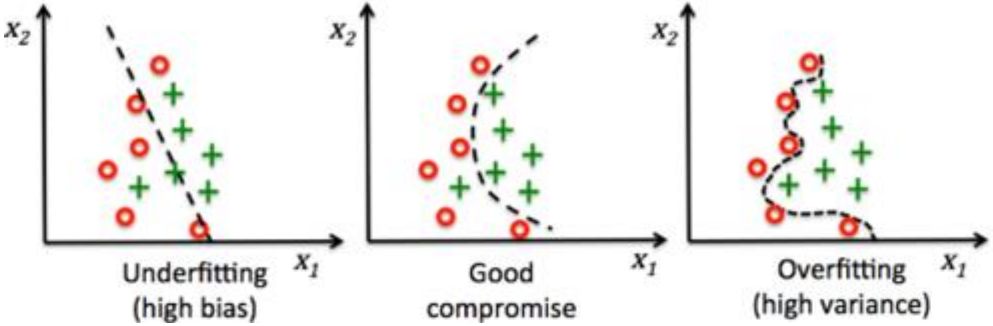


Feature classes and regularization multiplier

Feature type	Interpretation	Shape
Linear	Continuous variable	
Quadratic	Square of the variable	
Hinge	As threshold type, but response after the threshold (knot) is linear	

<u>Balance complexity and fit</u>

- Feature classes: shape of the expected response curve
- Response curve: the relationship between environmental variables and suitability
- Regularization multiplier: penalty applied to reduce overfitting



Sample code

Using tools in the ENMeval package

ENMevaluate generates models with different combinations of feature classes and RMs

```
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',
)
```

Sample code

Using tools in the ENMeval package

Filter the resulting models by selecting for the one with the lowest omission rate and highest area under the curve value

```
opt.seq <- results %>%
  filter(!is.na(AICc)) %>%                          # Exclude models with NA AICc
  filter(AICc == min(AICc)) %>%                     # Minimum AICc
  filter(or.10p.avg != 0) %>%                       # Exclude zero omission
  filter(or.10p.avg == min(or.10p.avg)) %>%         # Minimum omission
  filter(auc.val.avg == max(auc.val.avg))           # Maximum AUC
```

How does ENMevaluate generate null models?

1 - Use the same number of presence points as the empirical model, but sample them randomly from the background

2 - Fit models with those random points using the same algorithm and environmental predictors

3 - Evaluate performance using real withheld presence data

4 - Repeat many times to generate a **null distribution** of performance metrics

5 - Compare the **empirical model's AUC and omission rate** to the null distribution to assess significance

Sample code

Using tools in the ENMeval package

Generate null models using parameters from optimal model

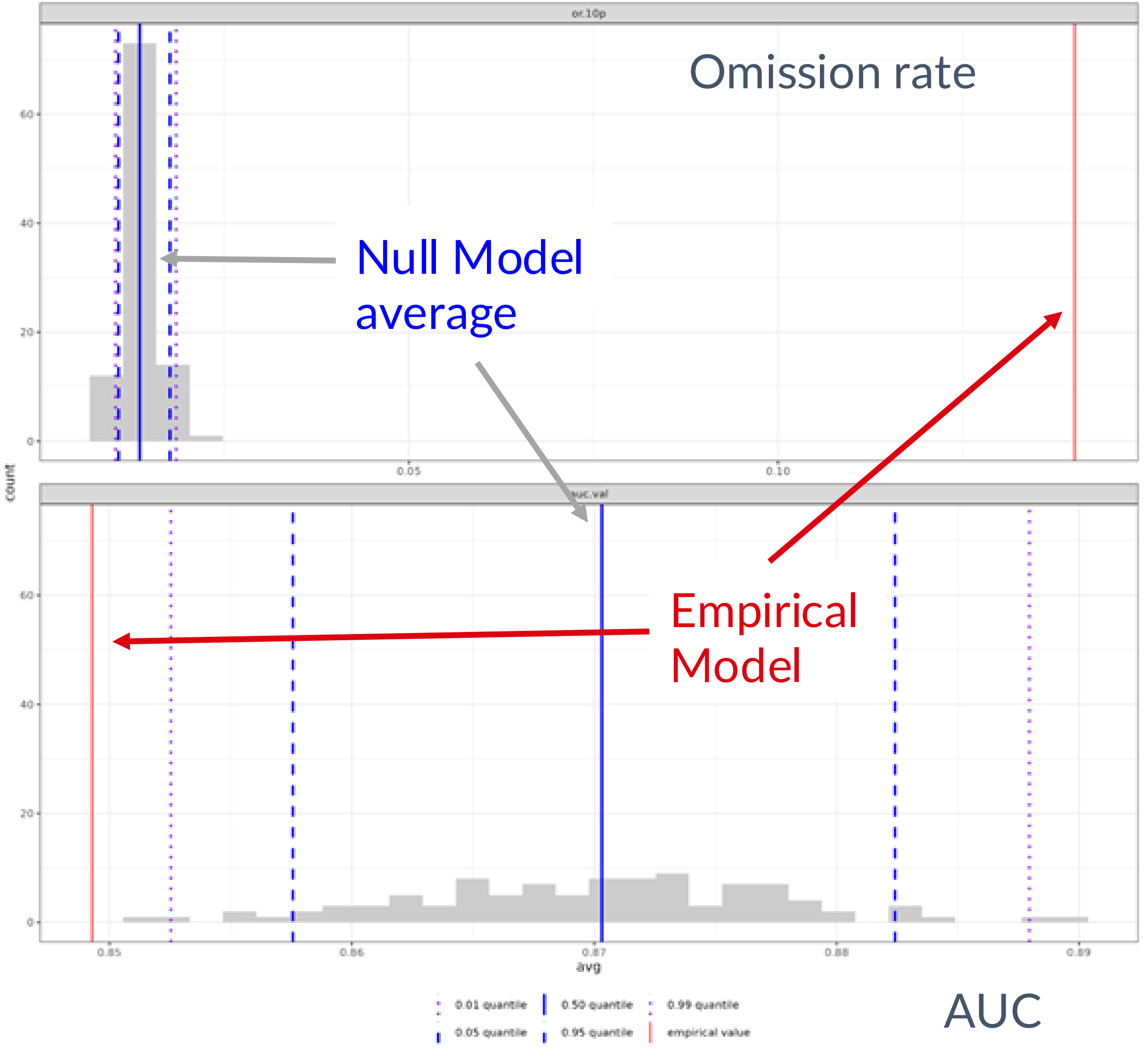
```
# Load optimal feature class (fc) and regularization multiplier (rm)
opt.seq <- read.delim("data/05_ENMs/Galax_urceolata_OptModel.txt")

# Extract parameters
fc <- opt.seq$fc
rm <- opt.seq$rm


# Run ENMnulls with optimal parameters and 100 iterations
spec.mod.null <- ENMnulls(
  eval,
  mod.settings = list(fc = fc, rm = rm),
  no.iter = 100
)
```

Output:

Plot shows the median AUC value and OR for the empirical model (red) against the null distribution values (blue)



Sample code

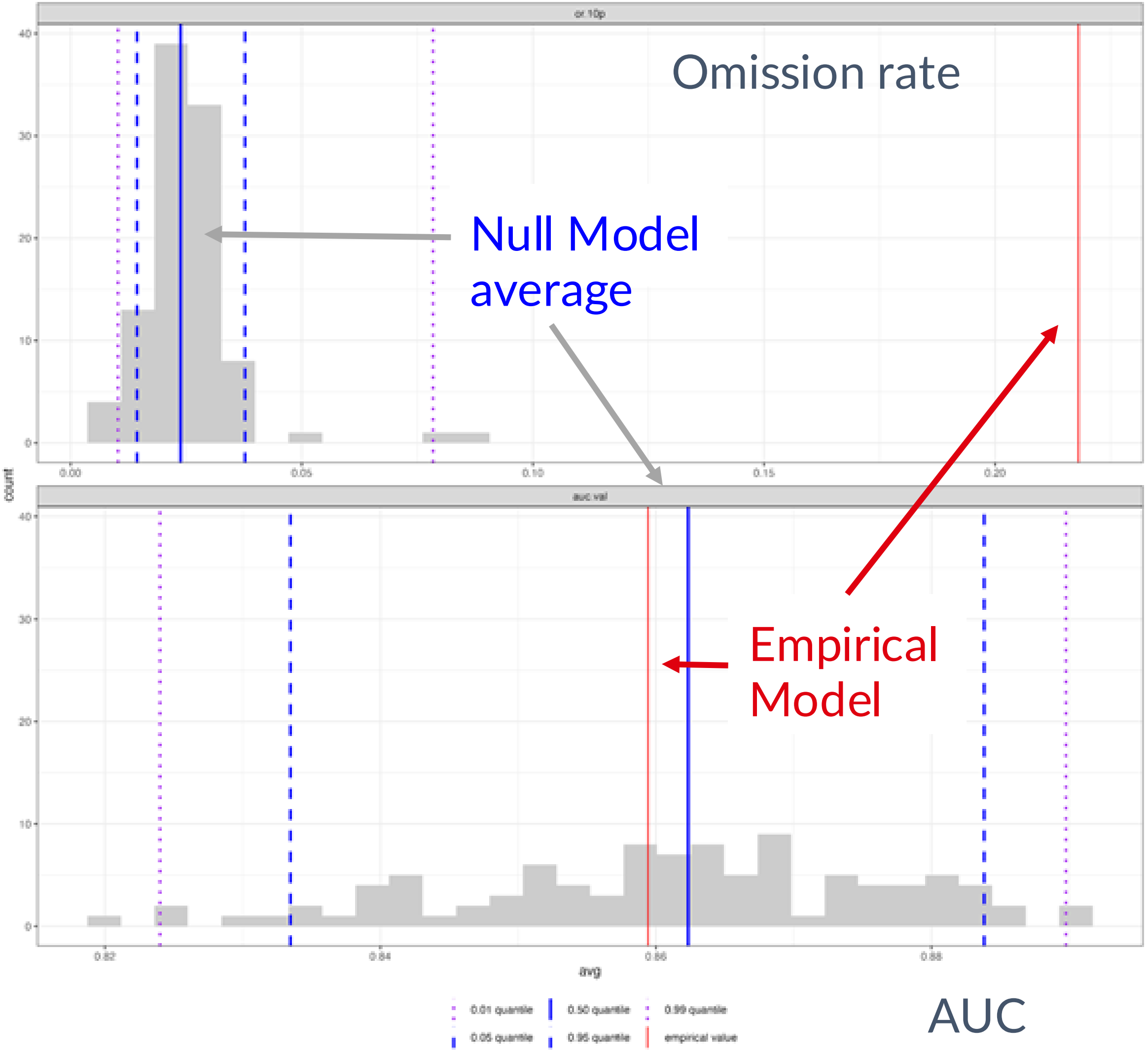
Using tools in the ENMeval package

Generate a new suite of models with a wider range of parameters

```
eval <- ENMevaluate(
  occs = Galax_urceolata[, c("longitude", "latitude")],
  envs = vifStack,
  tune.args = list(fc = c("L", "Q", "H", "LQ", "QH", "LH", "LQH"), rm = 1:5),
  partitions = "block",
  n.bg = 10000,
  parallel = FALSE,
  algorithm = 'maxent.jar',
)
```

Output:

New optimal model:
fc=H, rm=1



Why create null models?

- Determine whether model performance is better than expected by chance.
- Quantify how much better your model is than a null expectation.
- Avoid false confidence by evaluating for bias, spatial autocorrelation and overfitting
- Add rigor to model selection and interpretation