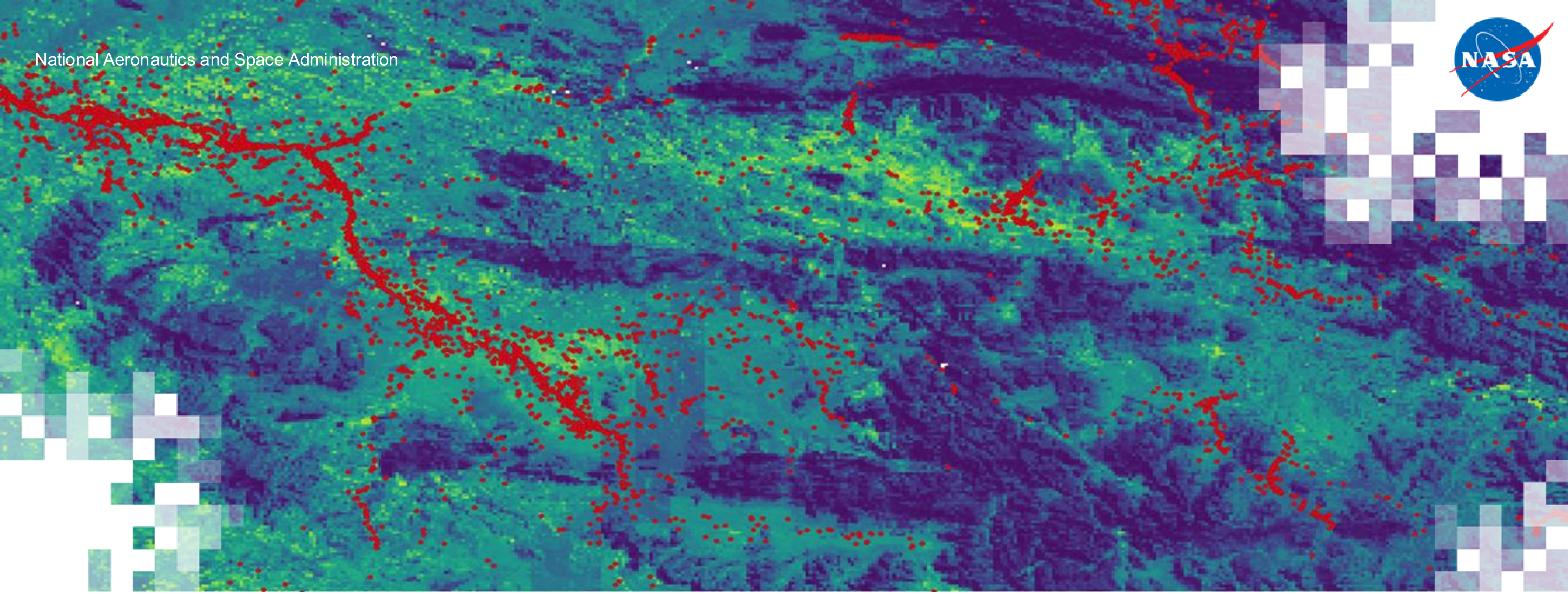
National Aeronautics and Space Administration



Species Distribution Modeling with Google Earth Engine

Part 2: Building, Evaluating, and Interpreting a SDM on GEE

ARSET Hosts: Justin Fain (NASA Ames/BAERI) & Sativa Cruz (NASA Ames/BAERI)
Guests: Ramiro Crego (University College Cork), Grant Connette (Smithsonian), & Jared Stabach (Smithsonian)

July 14, 2026

ARSET Eco Team

Justin Fain

Research Scientist
BAER/NASA Ames Research Center

Sativa Cruz

Applied Scientist
BAER/NASA Ames Research Center

Juan Torres-Perez

Research Scientist
NASA Ames Research Center

Savannah Cooley

Research Scientist
BAER/NASA Ames Research Center

Part 1 – Trainers

Ramiro D Crego

Lecturer in Animal Ecology

University College Cork

Grant Connette

Research Ecologist

Smithsonian's National Zoo & Conservation Biology Institute

Jared Stabach

Research Ecologist

Smithsonian's National Zoo & Conservation Biology Institute

Training Outline

Part 1

Foundations of Species Distribution Modeling with GEE

July 07, 2026

Session A: 9:00 a.m. to 10:30 a.m. EDT (UTC-4)

Session B: 4:00 p.m. to 5:30 p.m. EDT (UTC-4)

Part 2

Building, Evaluating, and Interpreting an SDM on GEE

July 14, 2026

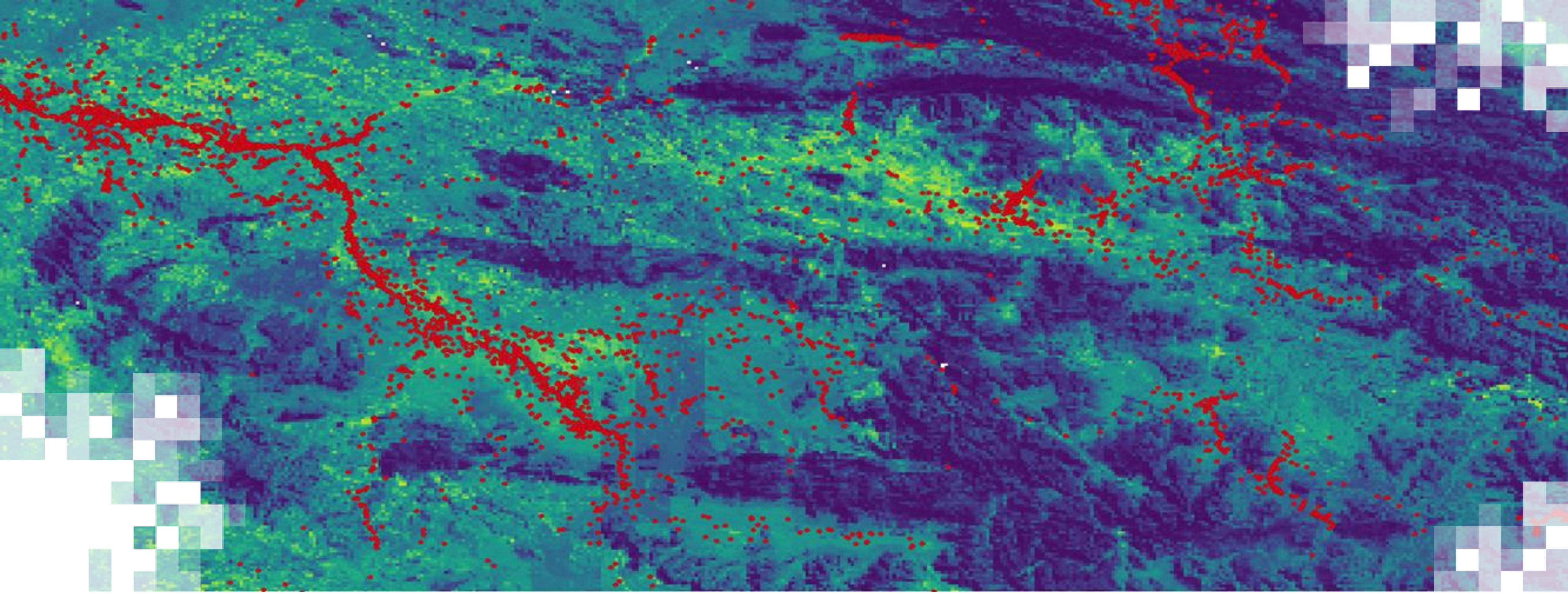
Session A: 9:00 a.m. to 10:30 a.m. EDT (UTC-4)

Session B: 4:00 p.m. to 5:30 p.m. EDT (UTC-4)

Homework

Opens July 14 – Due July 28– Posted on Training Webpage

A certificate of completion will be awarded to those who attend all live sessions and complete the homework assignment(s) before the given due date.



Species Distribution Modeling with Google Earth Engine

Part 2: Building, Evaluating, and Interpreting a SDM on GEE

Training Learning Objectives

By the end of this training, participants will be able to:

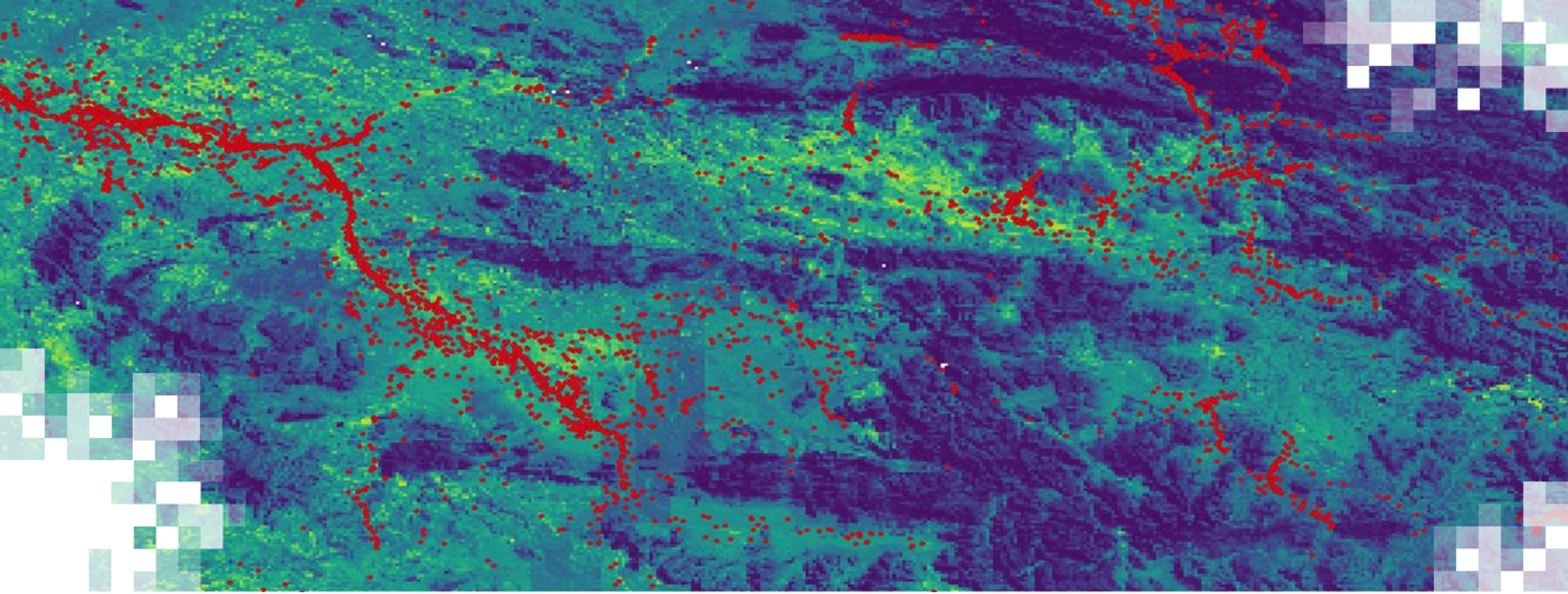
- Conduct basic operations and functions in the Google Earth Engine (GEE) web interface using JavaScript code.
- Identify core concepts and applications of species distribution modeling (SDM).
- Assess key SDM workflow decisions to ensure occurrence point independence, to select background points, and to choose ecologically relevant predictors.
- Access and manage analysis-ready spatial data from the GEE catalog as predictor variables.
- Fit species distribution models using machine learning models (e.g., Random Forest) in GEE using a reproducible code-based workflow.
- Critically evaluate the appropriateness of the model design and interpret predicted distribution and habitat suitability results.

Prerequisites

- Part 2 of Visualizing Land Cover and Land Use Change with NASA Satellite Imagery
- General knowledge of GIS
- Some coding experience

How to Ask Questions

- Please put your questions in the Questions box and we will address them at the end of the webinar.
- Feel free to enter your questions as we go. We will try to get to all of the questions during the Q&A session after the webinar.
- The remainder of the questions will be answered in the Q&A document, which will be posted to the training website about a week after the training.



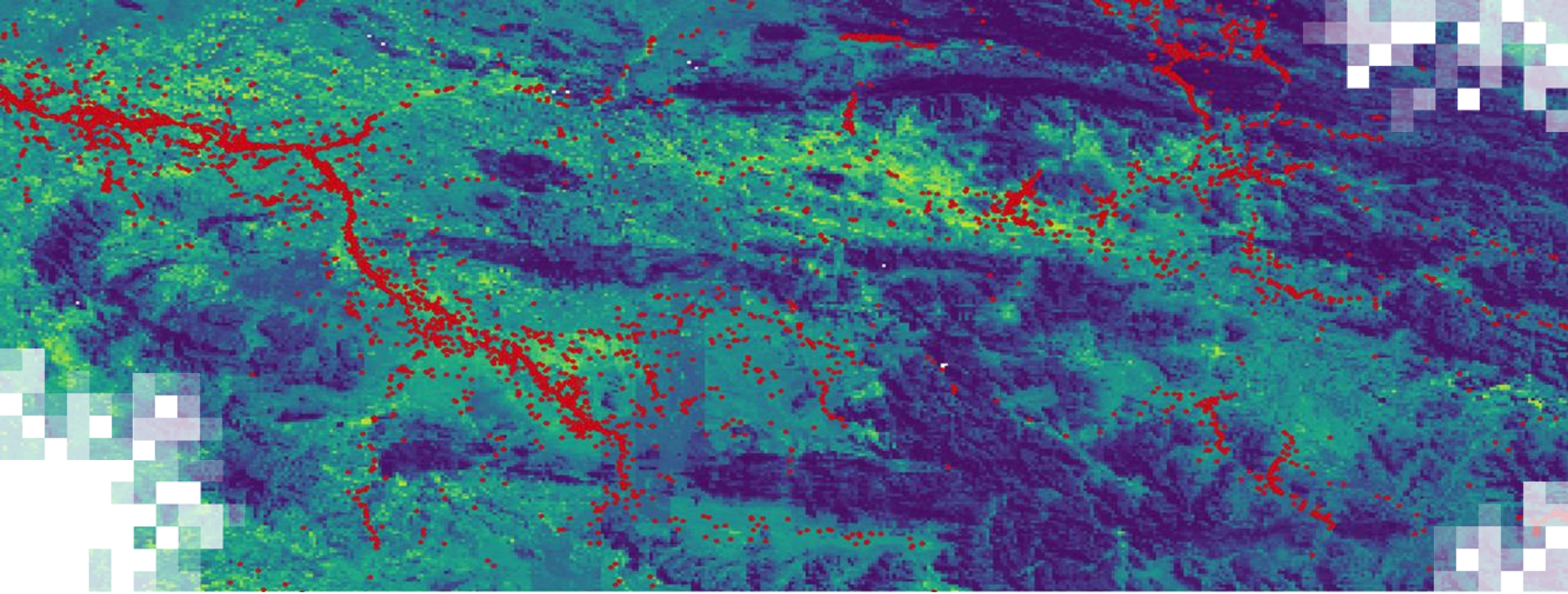
Species Distribution Modeling with Google Earth Engine

Ramiro Crego, Grant Connette, & Jared Stabach

July 14, 2026

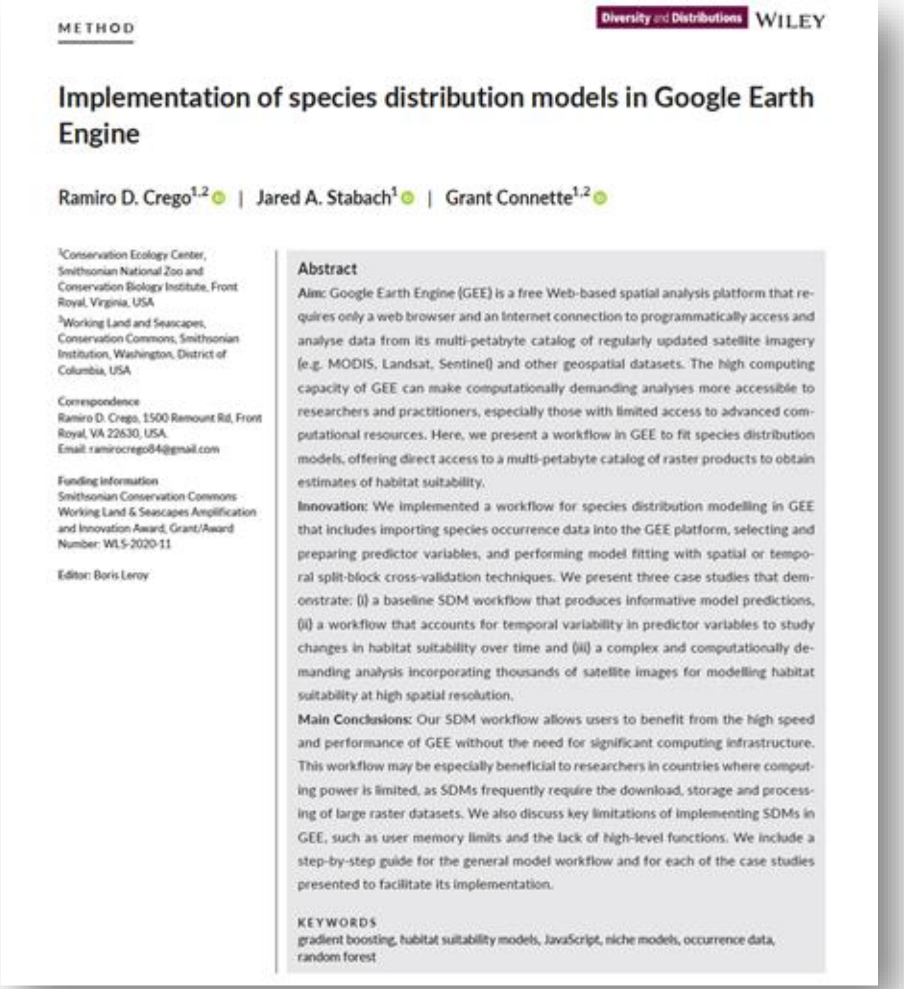
Learning Objectives

- Conduct basic operations and functions in Google Earth Engine (GEE) web interface using JavaScript code
- Identify core concepts and applications of Species Distribution Modeling (SDM)
- Assess key SDM workflow decisions to ensure occurrence point independence, select background points, and choose ecologically relevant predictors.
- Access and manage analysis-ready spatial data from the GEE catalog as predictor variables
- Fit species distribution models using machine learning models (e.g., Random Forest) in GEE using a reproducible code-based workflow
- Critically evaluate the appropriateness of the model design and interpret predicted distribution and habitat suitability results



Building a Machine Learning Model

The Workflow


METHOD

Diversity and Distributions WILEY

Implementation of species distribution models in Google Earth Engine

Ramiro D. Crego[1,2] | Jared A. Stabach[1] | Grant Connette[1,2]

[1]Conservation Ecology Center, Smithsonian National Zoo and Conservation Biology Institute, Front Royal, Virginia, USA

[2]Working Land and Seascapes, Conservation Commons, Smithsonian Institution, Washington, District of Columbia, USA

Correspondence
Ramiro D. Crego, 1500 Remount Rd, Front Royal, VA 22630, USA.
Email: ramirocrego84@gmail.com

Funding information
Smithsonian Conservation Commons Working Land & Seascapes Amplification and Innovation Award, Grant/Award Number: WLS-2020-11

Editor: Boris Leroy

Abstract

Aim: Google Earth Engine (GEE) is a free Web-based spatial analysis platform that requires only a web browser and an Internet connection to programmatically access and analyse data from its multi-petabyte catalog of regularly updated satellite imagery (e.g. MODIS, Landsat, Sentinel) and other geospatial datasets. The high computing capacity of GEE can make computationally demanding analyses more accessible to researchers and practitioners, especially those with limited access to advanced computational resources. Here, we present a workflow in GEE to fit species distribution models, offering direct access to a multi-petabyte catalog of raster products to obtain estimates of habitat suitability.

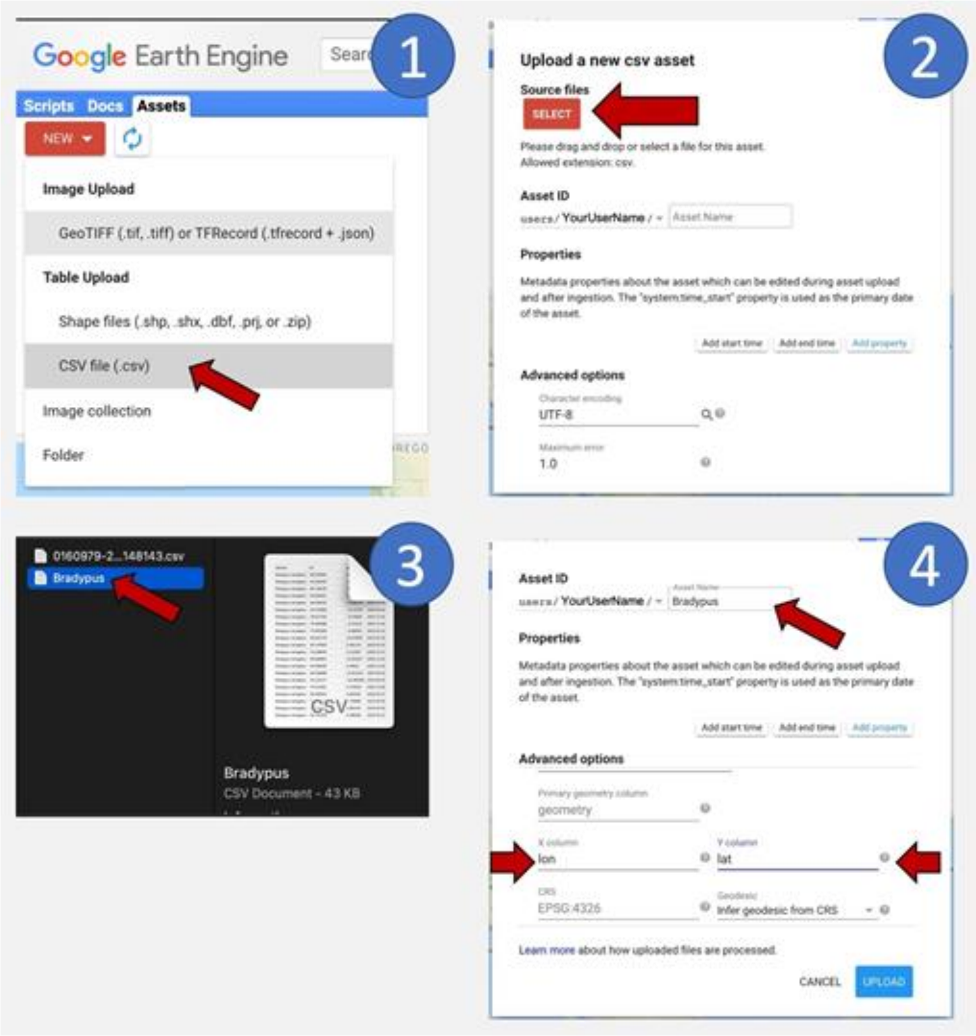
Innovation: We implemented a workflow for species distribution modelling in GEE that includes importing species occurrence data into the GEE platform, selecting and preparing predictor variables, and performing model fitting with spatial or temporal split-block cross-validation techniques. We present three case studies that demonstrate: (i) a baseline SDM workflow that produces informative model predictions, (ii) a workflow that accounts for temporal variability in predictor variables to study changes in habitat suitability over time and (iii) a complex and computationally demanding analysis incorporating thousands of satellite images for modelling habitat suitability at high spatial resolution.

Main Conclusions: Our SDM workflow allows users to benefit from the high speed and performance of GEE without the need for significant computing infrastructure. This workflow may be especially beneficial to researchers in countries where computing power is limited, as SDMs frequently require the download, storage and processing of large raster datasets. We also discuss key limitations of implementing SDMs in GEE, such as user memory limits and the lack of high-level functions. We include a step-by-step guide for the general model workflow and for each of the case studies presented to facilitate its implementation.

KEYWORDS
gradient boosting, habitat suitability models, JavaScript, niche models, occurrence data, random forest

1. Species data upload
2. Preprocessing
3. Predictor variables
4. Presence and absence
    a. Presence and absence
    b. Presence and pseudoabsence
    c. Presence and absence and pseudoabsence
5. Model fitting and model evaluation
6. Model predictions

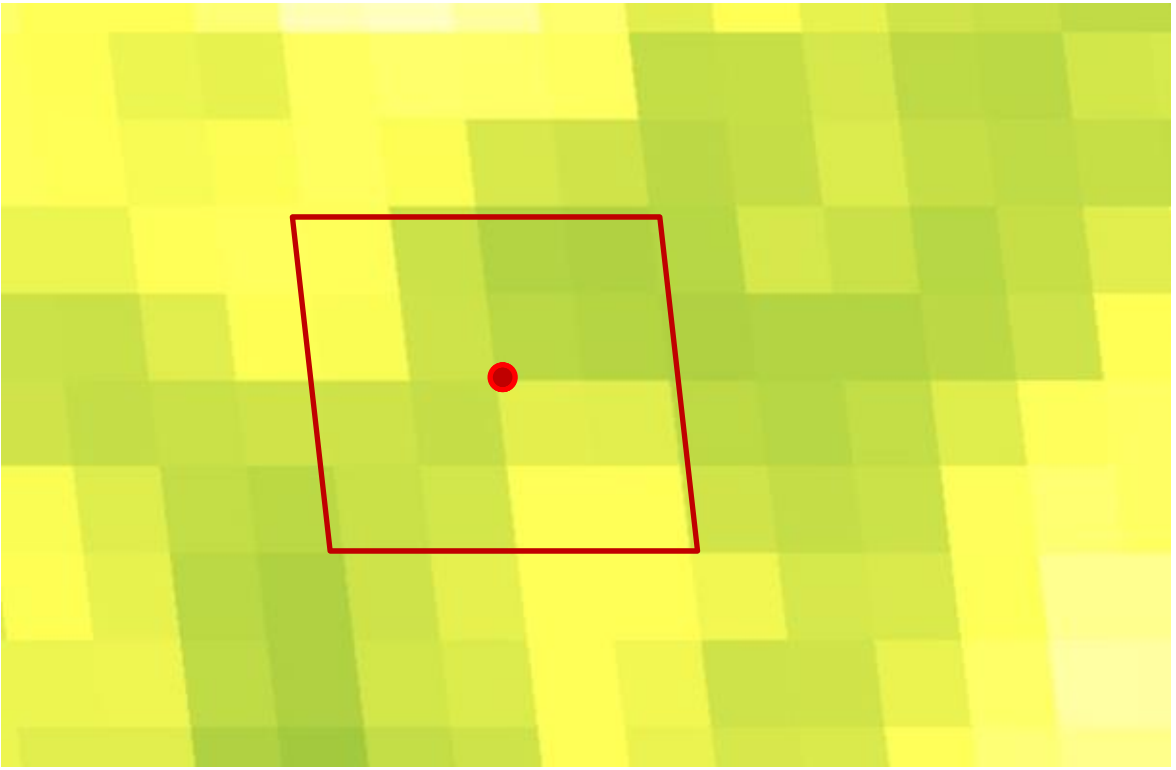
Uploading Your Own Data as an Asset




```javascript
// Load presence data
var Data = ee.FeatureCollection('users/ramirocrego84/BradypusVariegatus');
print('Original data size:', Data.size());

// Visualize presence points on the map
Map.addLayer(Data, {color:'red'}, 'Presence', 1);
```

Define the Spatial Resolution of the Analysis

```
// Define spatial resolution to work with (m)
var GrainSize = 1000;
```

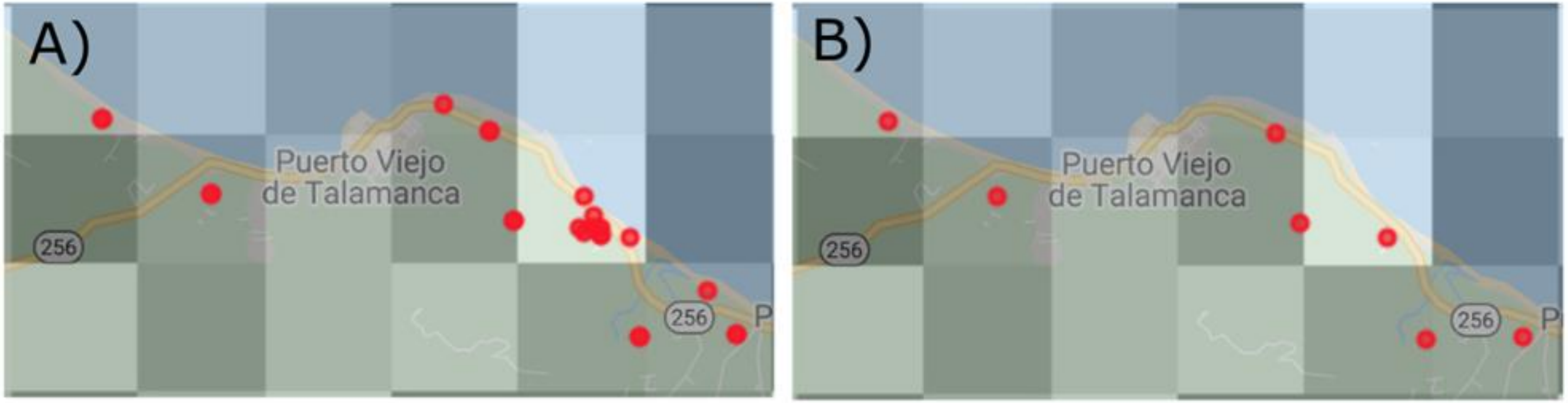


Spatial Thinning of Presence Data

- People often record a species multiple times in the exact same location (e.g., near roads or in parks). We can thin presence data to mitigate effects of sampling bias on our model.
- We can thin our location data to one randomly selected occurrence record per pixel at the chosen spatial resolution (the raster pixel or grain size of the analysis).

```
function RemoveDuplicates(data){
  var randomraster = ee.Image.random().reproject('EPSG:4326', null, GrainSize);
  var randpointvals = randomraster.sampleRegions({collection:ee.FeatureCollection(data), scale: 10, geometries: true});
  return randpointvals.distinct('random');
}

var Data = RemoveDuplicates(Data);
print('Final data size:', Data.size());
```

```
Use print(...) to write
to this console.

Original data size:  JSON
571

Final data size:     JSON
323
```



Define Your Area of Interest for Modelling

- The extent of the analysis should be carefully selected and constrained to a geographic region that is actually accessible to the species.

- Avoid massive analysis extents - comparing against distant absence points inflates accuracy metrics and focuses on obvious macro-scale patterns (e.g., forest vs. desert).

```
// Load region boundary from data catalog if working at a larger scale
var AOI = ee.FeatureCollection('USDOS/LSIB_SIMPLE/2017').filter(ee.Filter.eq('country_co', 'CO'));
Map.addLayer(AOI)

// Load country boundary from data catalog if working at a country scale
var AOI2 = ee.FeatureCollection('USDOS/LSIB_SIMPLE/2017').filter(ee.Filter.eq('wld_rgn', 'South America')).union().geometry();
Map.addLayer(AOI2)
```

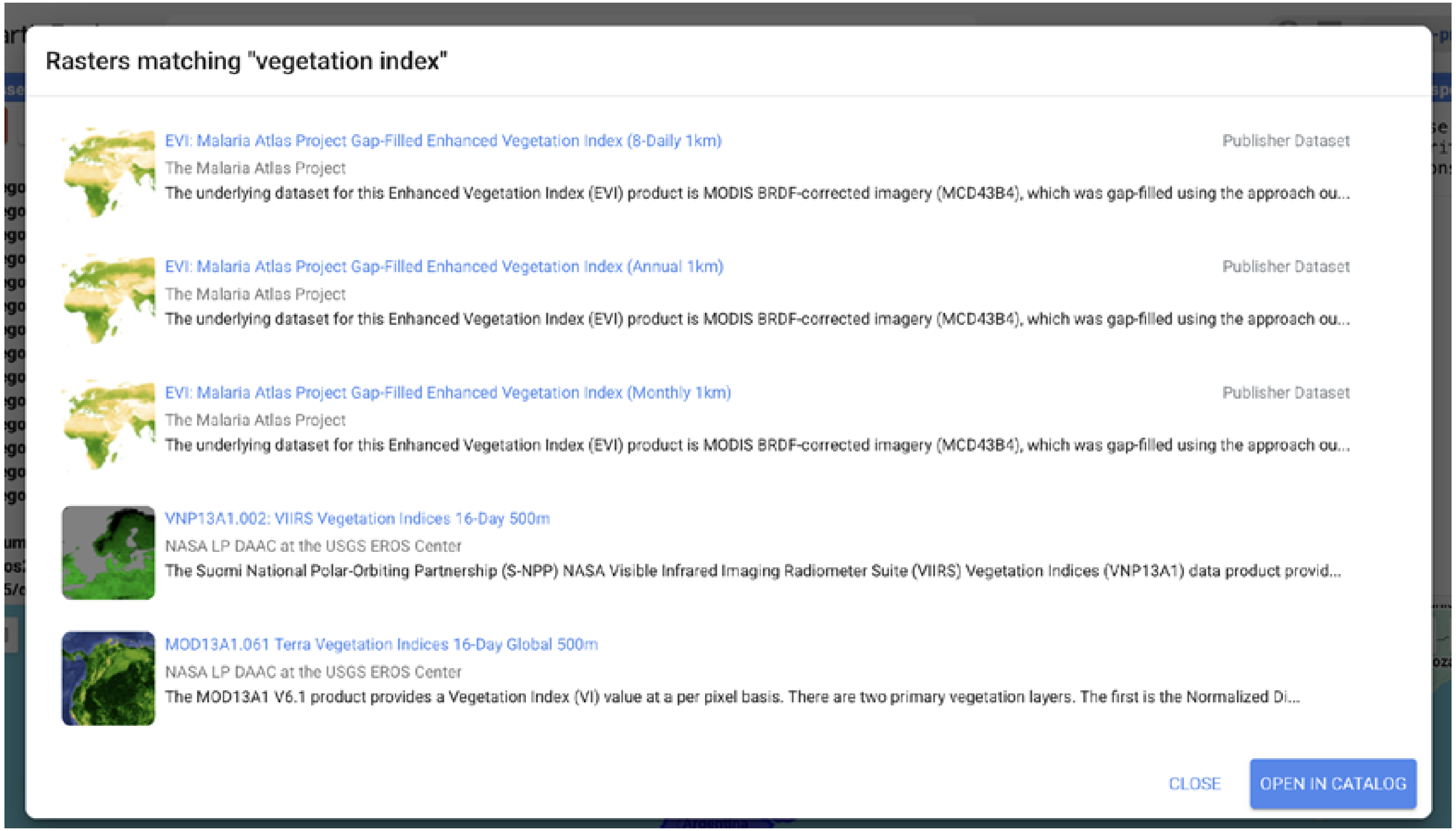
Define Your Area of Interest for Modelling

- The extent of the analysis should be carefully selected and constrained to a geographic region that is actually accessible to the species.
- Avoid massive analysis extents - comparing against distant absence points inflates accuracy metrics and focuses on obvious macro-scale patterns (e.g., forest vs. desert).

```
/////////////////////////////////////////////////////////////////////////////.
// Section 2 - Define Area of Interest (Extent)
/////////////////////////////////////////////////////////////////////////////.

// Define the AOI
var AOI = Data.geometry().bounds().buffer({distance:50000, maxError:1000});

// Add border of study area to the map
var outline = ee.Image().byte().paint({
  featureCollection: AOI, color: 1, width: 3});
right.addLayer(outline, {palette: 'FF0000'}, "Study Area");
left.addLayer(outline, {palette: 'FF0000'}, "Study Area");

// Center map to the area of interest
right.centerObject(AOI, 4); //Number indicates the zoom level
left.centerObject(AOI, 4); //Number indicates the zoom level
```

Selecting Predictor Variables

- A key advantage of implementing SDMs in Google Earth Engine is its cloud-hosted data catalog, allowing users to directly access many geospatial predictor variables via code.



Using Raw Satellite Data as Predictor Variables

Diversity and Distributions, (Diversity Distrib.) (2013) **19**, 855–866

BIODIVERSITY RESEARCH

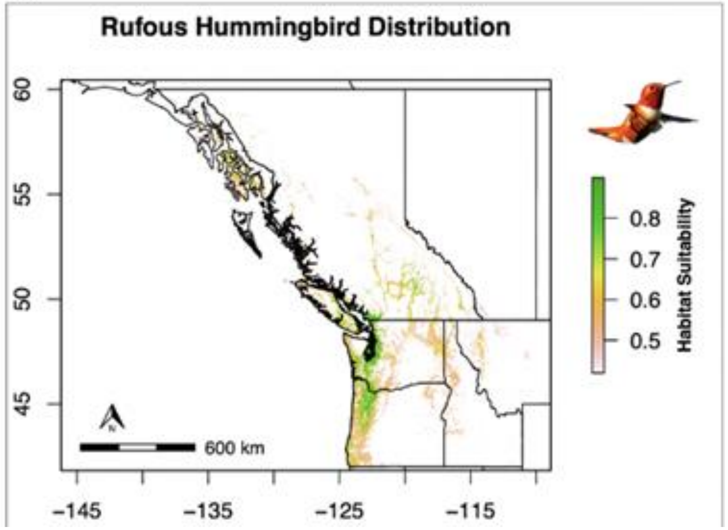
Species distribution modelling for the people: unclassified landsat TM imagery predicts bird occurrence at fine resolutions

S. M. Shirley[1]*, Z. Yang[1], R. A. Hutchinson[2], J. D.Alexander[3], K. McGarigal[4] and M. G. Betts[1]

Breeding habitat loss linked to declines in Rufous Hummingbirds

Kendall M. Jefferys[1], Matthew G. Betts[2], W. Douglas Robinson[3], Jenna R. Curtis[3], Tyler A. Hallman[3,4]
Adam C. Smith[5], Chloë Strevens[1], Jesús Aguirre Gutiérrez[1,6]

(a)



RESTORATION ECOLOGY
SER

RESEARCH ARTICLE

Validating habitat suitability assessments with post-translocation animal movement data

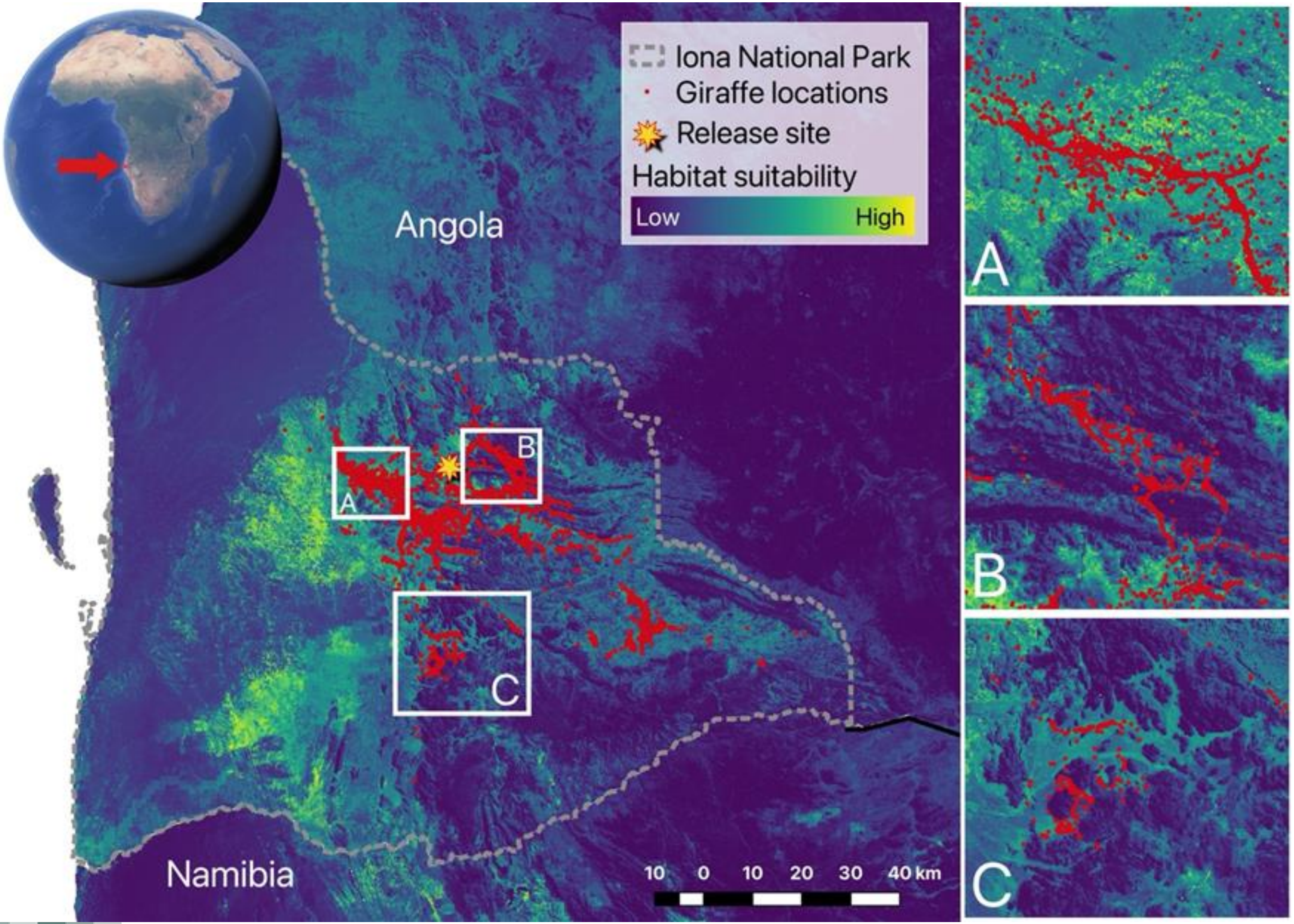
Ramiro D. Crego[1,2,3,4], Michael B. Brown[3,5], Jared A. Stabach[3], Julian Fennessy[5,6], Stephanie Fennessy[5], Pedro Monterroso[7], Daniel van de Vyver[7], Courtney Marneweck[5,8]

Wildlife translocations are essential for restoring species to their native habitats, and species distribution models (SDMs) have proven useful for identifying suitable habitats to guide this process. Validating model predictions with data from individual's post-translocation, however, is rare due to the frequent lack of monitoring data from translocated individuals. The translocation of 14 Angolan giraffe (*Giraffa giraffa angolensis*) in 2023 to Iona National Park (NP), Angola, to restore its devastated fauna provided an opportunity to test the ability of SDMs to inform suitable habitat for translocation programs. Here, we describe the SDM we developed and implemented using Google Earth Engine (GEE) to inform the translocation process, combining Angolan giraffe movement data from 45 individuals from a population in northern Namibia with moderate resolution satellite imagery to predict suitable habitat throughout Iona NP. We then used telemetry data from the 14 translocated individuals to validate the results from the model prediction and to summarize their movements post-translocation. The mean predicted habitat suitability of pixels used by Angolan giraffe was significantly higher than that of random locations across the landscape. Translocated giraffe movements and home ranges were comparable to those of giraffe from other populations. They displaced at least 21.6 km from the translocation release site and remained within the park limits. Our study represents a truly independent validation of a habitat suitability model prediction, demonstrating that SDMs combined with moderate-resolution satellite data can play a critical role in informing wildlife translocations.

Key words: Angola, *Giraffa giraffa angolensis*, giraffe, reintroduction, species distribution model

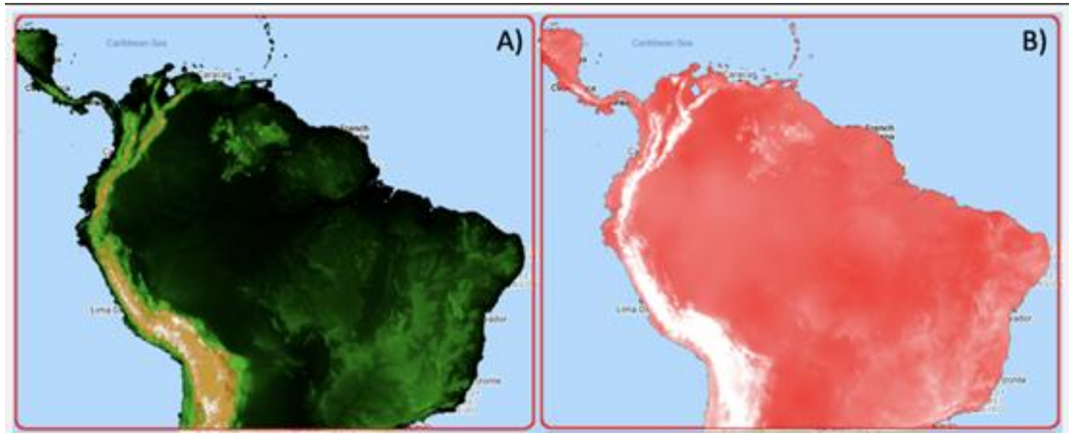
Suitable habitat maps are valuable to identify potential areas for animal translocations.

On 5 July 2023, 14 giraffe were translocated to Iona National Park



Selecting Predictor Variables

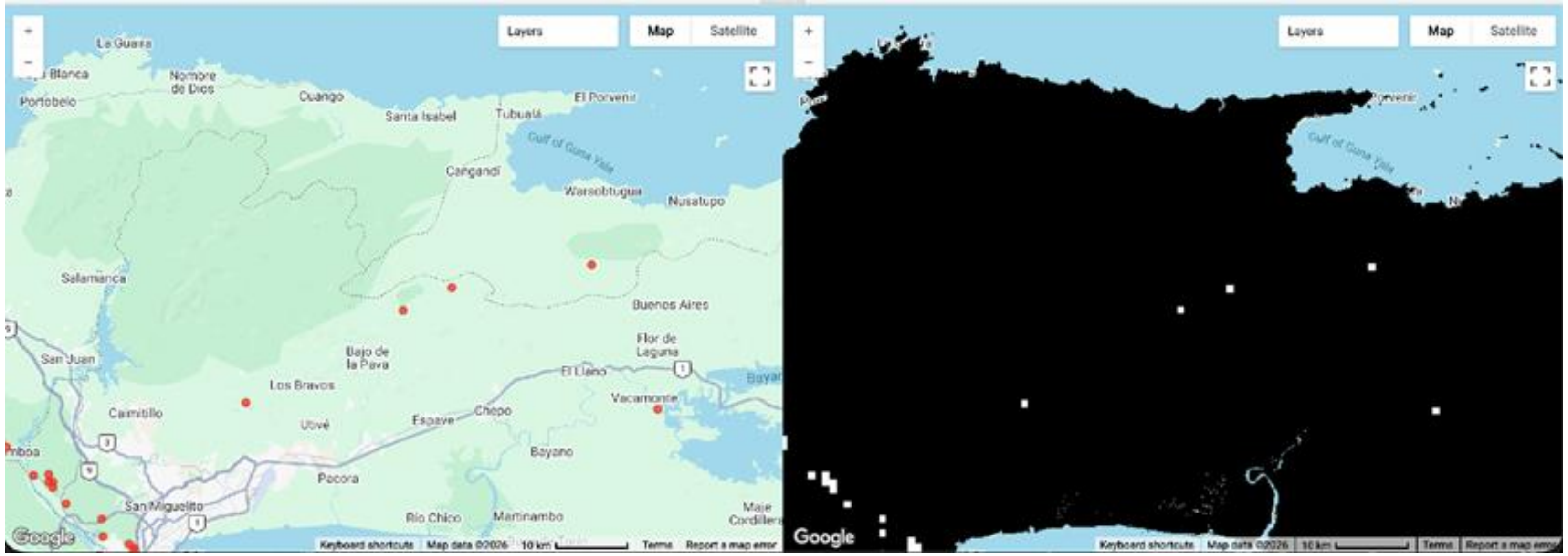
```
/////////////////////////////////////////////////////////////////////////////////////////////////////////
// Section 3 - Selectiong Predictor Variables
/////////////////////////////////////////////////////////////////////////////////////////////////////////

// Load WorldClim BIO Variables (a multiband image) from the data catalog
var BIO = ee.Image("WORLDCLIM/V1/BIO");

// Load elevation data from the data catalog and calculate slope, aspect, and a simple hillshade from the terrain Digital Elevation Model.
var Terrain = ee.Algorithms.Terrain(ee.Image("USGS/SRTMGL1_003"));

// Load NDVI 250 m collection and estimate median annual tree cover value per pixel
var MODIS = ee.ImageCollection('MODIS/061/MOD44B');
var MedianPTC = MODIS.filterDate('2003-01-01', '2020-12-31').select(['Percent_Tree_Cover']).median();

// Combine bands into a single multi-band image
var predictors = BIO.addBands(Terrain).addBands(MedianPTC);
print(predictors)

// Mask out ocean pixels from the predictor variable image
var watermask =  Terrain.select('elevation').gt(0); //Create a water mask
var predictors = predictors.updateMask(watermask).clip(AOI);

// Display layers on the map
right.addLayer(predictors, {bands:['elevation'], min: 0, max: 5000,  palette: ['000000','006600', '009900','33CC00','996600','CC9900','CC9966','FFFFFF',]},
'Elevation (m)', 0);
right.addLayer(predictors, {bands:['bio05'], min: 190, max: 400, palette:'white,red'}, 'Temperature seasonality', 0);
right.addLayer(predictors, {bands:['bio12'], min: 0, max: 4000, palette:'white,blue'}, 'Annual Mean Precipitation (mm)', 0);
right.addLayer(predictors, {bands:['Percent_Tree_Cover'], min: 1, max: 100, palette:'white,yellow,green'}, 'Percent_Tree_Cover', 0);

// Select subset of bands to keep for habitat suitability modeling
var bands = ['bio04','bio05','bio06','bio12','elevation','Percent_Tree_Cover'];
var predictors = predictors.select(bands);
```



Defining the Area to Create Pseudo-absences

- We first need to create a mask to avoid creating pseudo-absences in the same pixels where we have presence points.

```
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Section 4 - Defining area for pseudo-absences and spatial blocks for model fitting and cross validation
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// Make an image out of the presence locations. The pixels where we have presence locations will be removed from the area
// to generate pseudo-absences.
// This will prevent having presence and pseudo-absences in the same pixel.
var mask = Data
  .reduceToImage({
    properties: ['random'],
    reducer: ee.Reducer.first()
}).reproject('EPSG:4326', null, ee.Number(GrainSize));
```



Defining the Area to Create Pseudo-absences

Three main methods:

1. Generate random pseudo-absences across the entire study area (not recommended for pseudo-absence as it risks sampling suitable habitat; use only for creating background points).

Defining the Area to Create Pseudo-absences

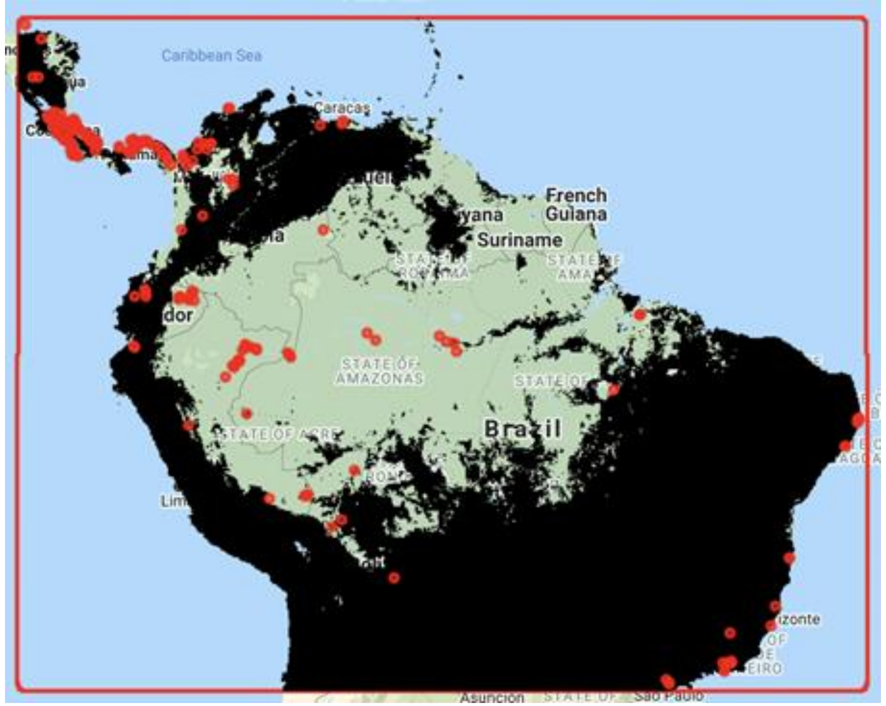
Three main methods:

2. Generate pseudo-absences within a specified distance range from presence locations. This limits pseudo-absences to areas potentially accessible to the species (Araújo et al., 2019) and can be used to mirror potential spatial or environmental sampling bias present in the occurrence data (Phillips et al., 2009).

Defining the Area to Create Pseudo-absences

Three main methods:

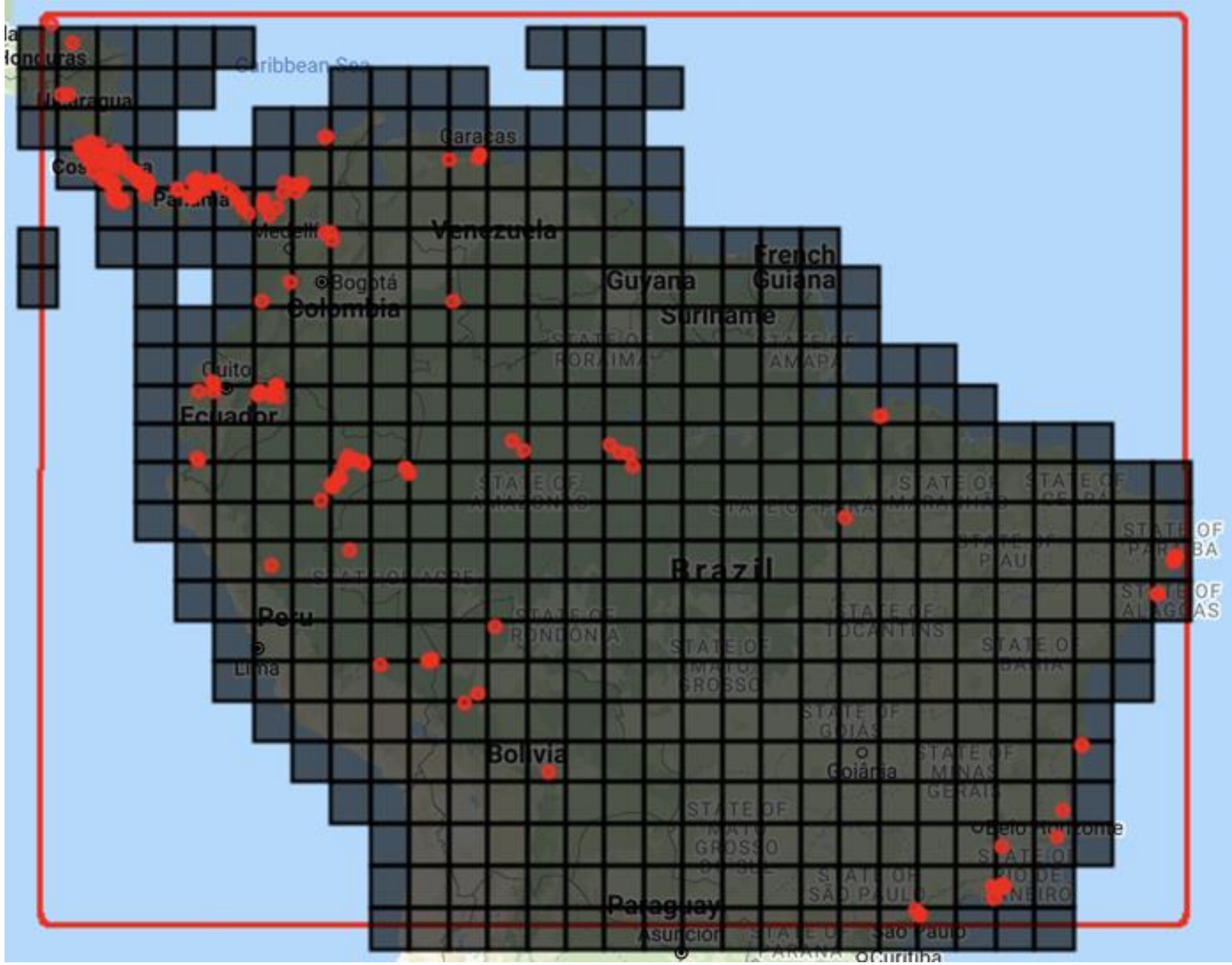
3. Constrain pseudo-absences by masking out suitable areas using an environmental profiling technique (Chefaoui & Lobo, 2008; Senay, Worner, & Ikeda, 2013). This ensures pseudo-absences are drawn from lower-suitability environments and minimizes false absences.



Spatial Block Cross-validation

Divide the study area into geographic blocks and randomly assign blocks into separate subsets for model training and validation (Roberts et al., 2017; Valavi, Elith, Lahoz-Monfort, & Guillera-Arroita, 2019).

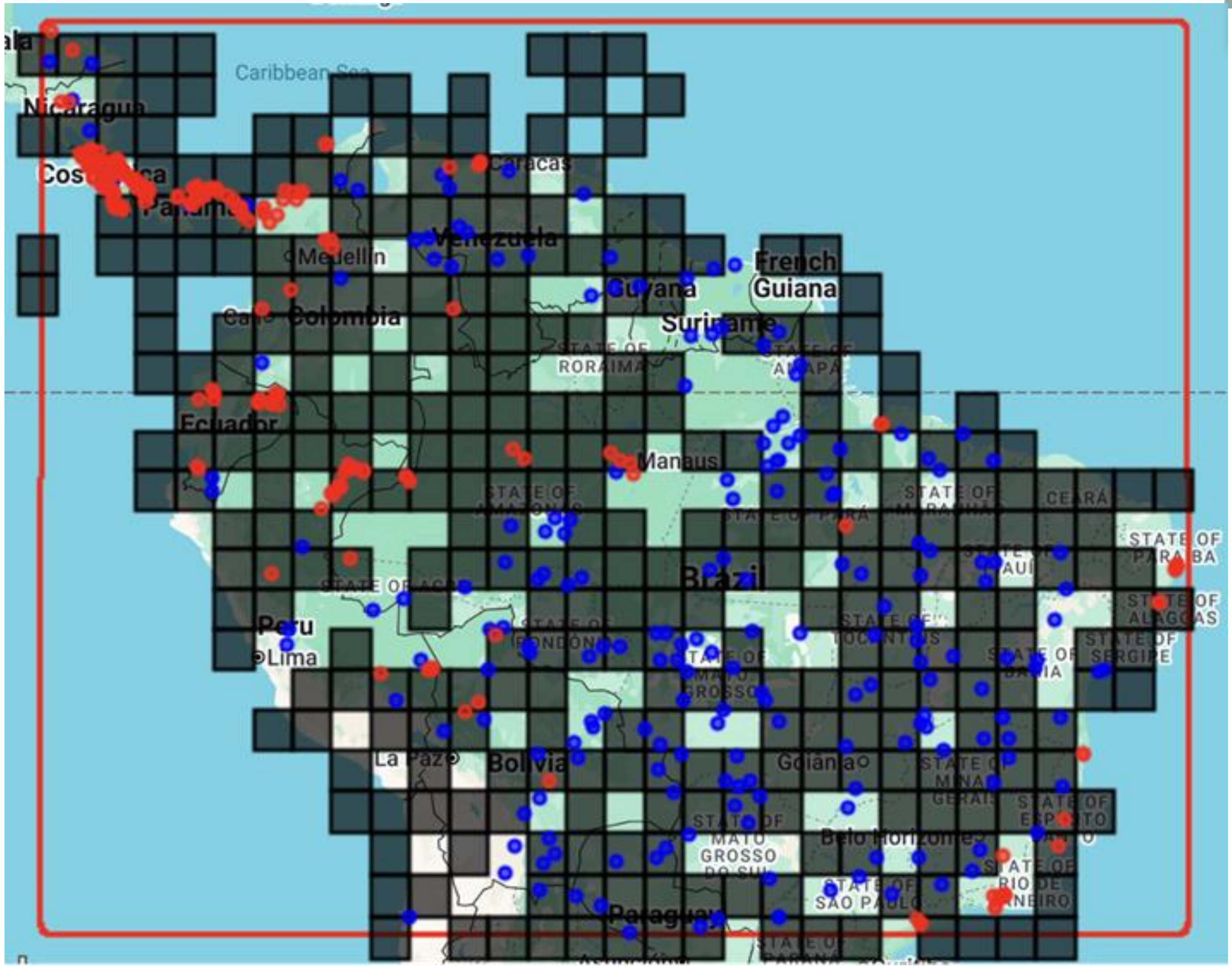
In our workflow, we run multiple model iterations with random block splits to create training and validation data sets.

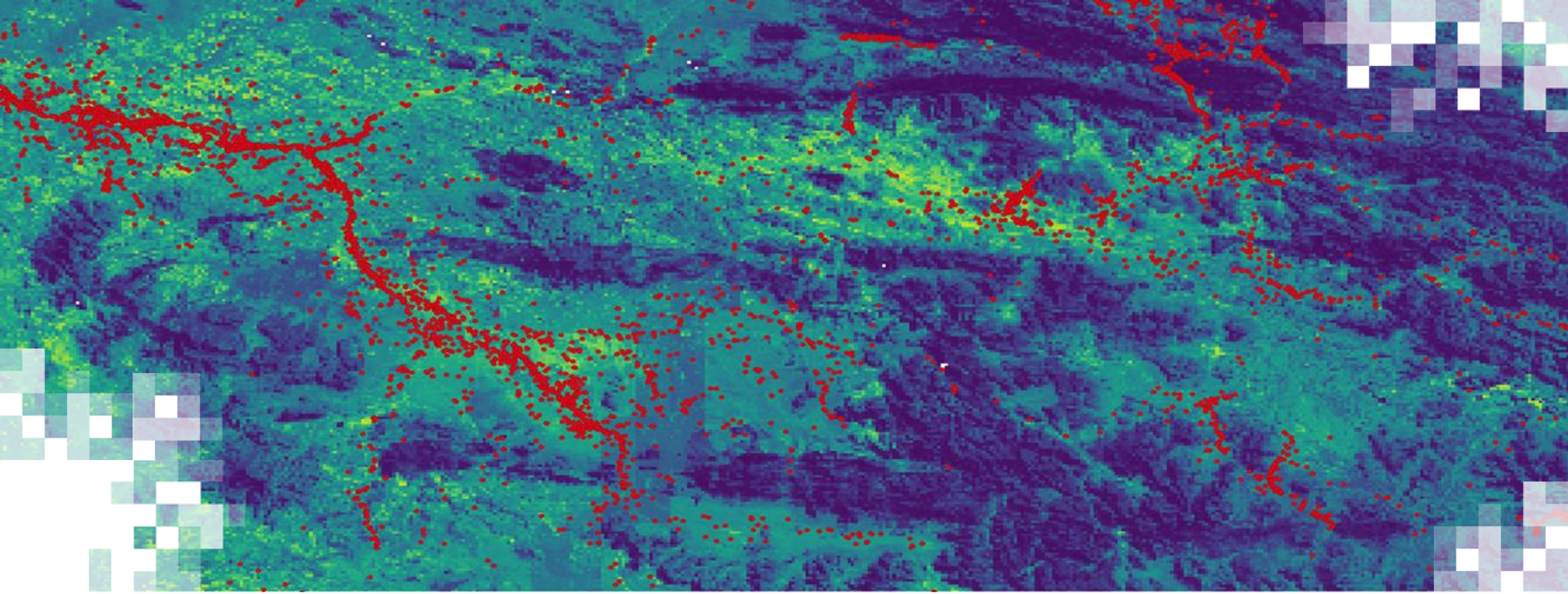


Spatial Block Cross-validation

We will select location data from 70% of the blocks for model training, including presence and pseudo-absences.

All points outside these blocks are going to be used for model testing.

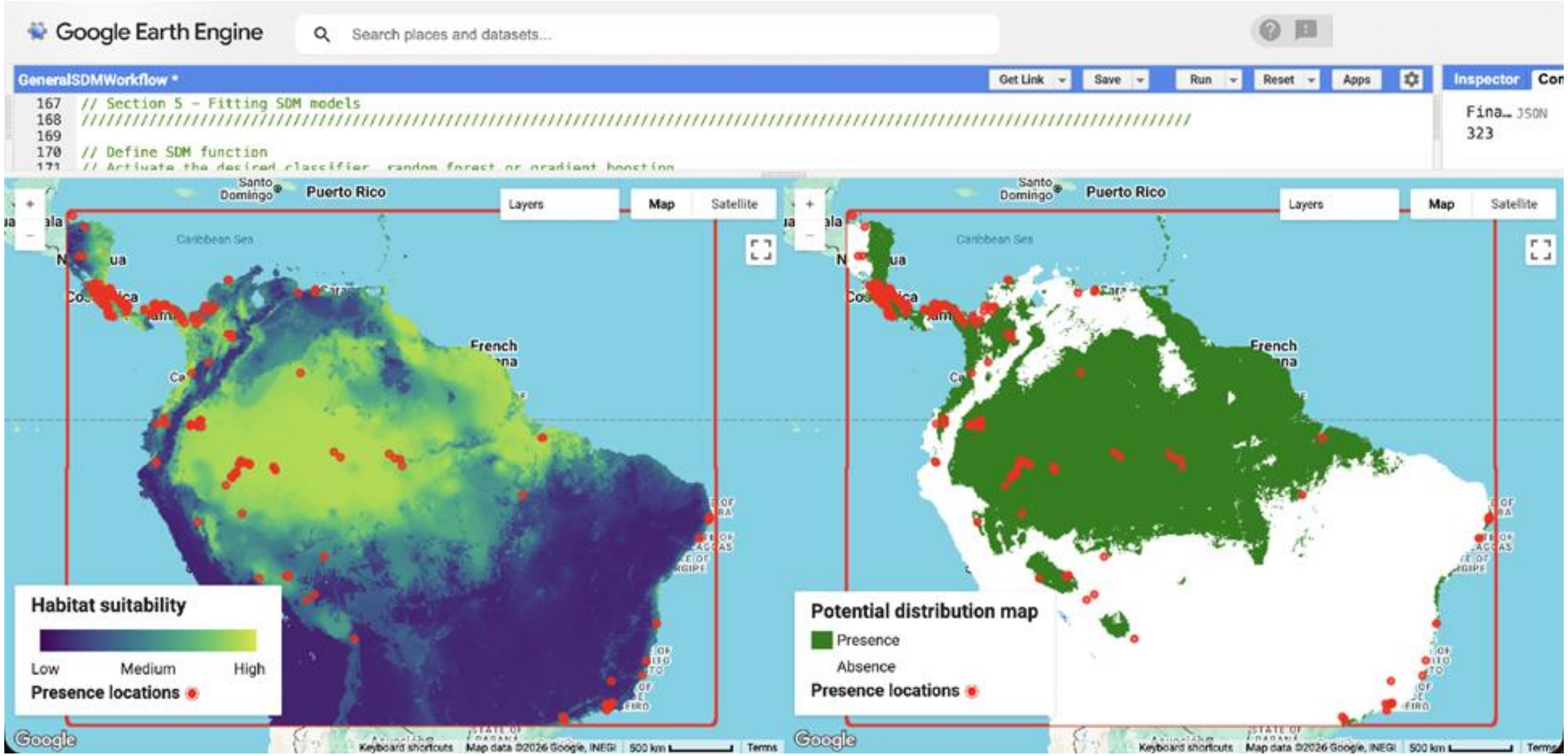




Model Fitting and Model Validation

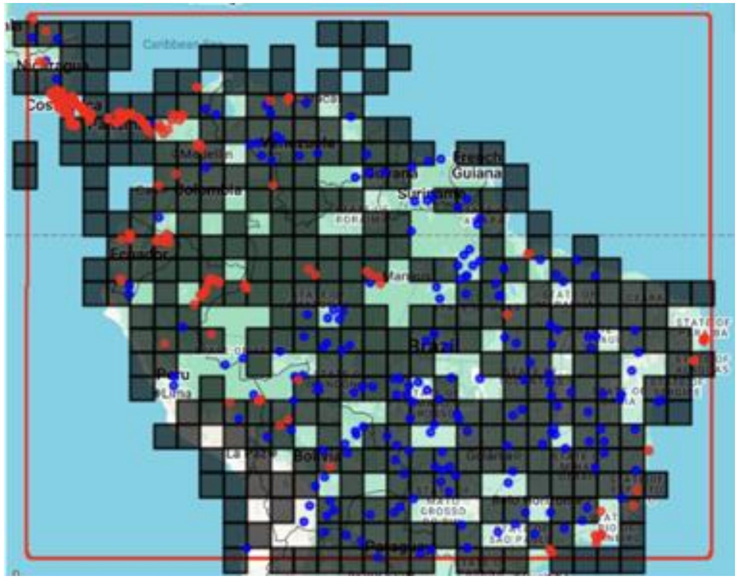
Fitting an SDM Model

- Next, we will define the SDM function.
- Let's go to the code editor.



Validating an SDM

- Ask basic questions:
  - Is this model meaningful?
  - Do the predictions make ecological sense?
- Derive evaluation metrics:
  - Deviance explained, residual patterns, goodness-of-fit test
- **Cross-validation:**
  - **Evaluation of model predictions with independent data**
  - **Partitioning data: training data vs. testing data**



Validating an SDM

- **Quantify model accuracy (Interpolation)**: Test predictive performance of the model within the **same** region and timeframe as the training data (Araújo et al. 2019).

- **Quantify model generality (Extrapolation)**: Test predictive performance of the model in a **different** region and timeframe from the training data (Araújo et al. 2019).

Methods for Model Validation

Threshold dependent metrics

- Requires presence/absence data (or pseudo-absences)
- Calculated from an error matrix (a.k.a. confusion matrix)
- Examples: Percent overall accuracy, Kappa, True Skill Statistic (TSS)
- Key challenge: selecting the optimal threshold for converting suitability to a binary 0/1 map

Threshold independent metrics

- Area under the Receiver Operating Characteristic Curve (AUC-ROC)
- Area under the Precision Recall Curve (AUC-PR)

The Error Matrix

		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Sensitivity (recall): the proportion of actual presence records the model correctly identifies out of all actual presences, emphasizing the prevention of false negatives

Specificity: the proportion of actual absence locations a model correctly identifies out of all actual absences, emphasizing the prevention of false positives

Precision: the proportion of predicted presence locations that are truly presences out of all total predicted presences, emphasizing the reliability of your suitability map

The Error Matrix: Example

Testing presence data

	x	
	x	x

Predicted probability

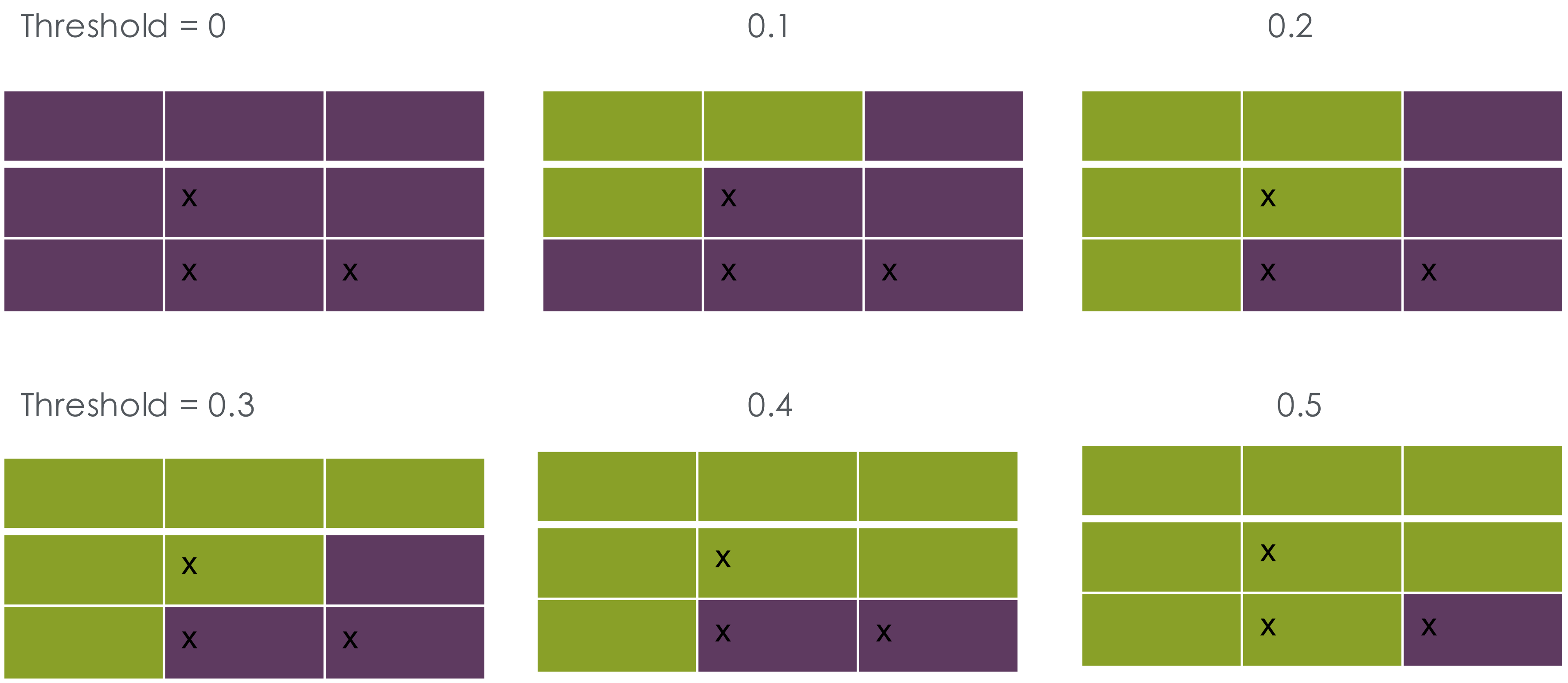
0	**0.1**	**0.3**
0.1	0.2	0.4
0.2	0.5	0.6

Sensitivity

$$\frac{TP}{TP + FN}$$

Specificity

$$\frac{TN}{TN + FP}$$



Threshold Independent Validation: AUC-ROC

- Area under the ROC (AUC) provides a single measure of model performance
- Independent of any particular choice of threshold

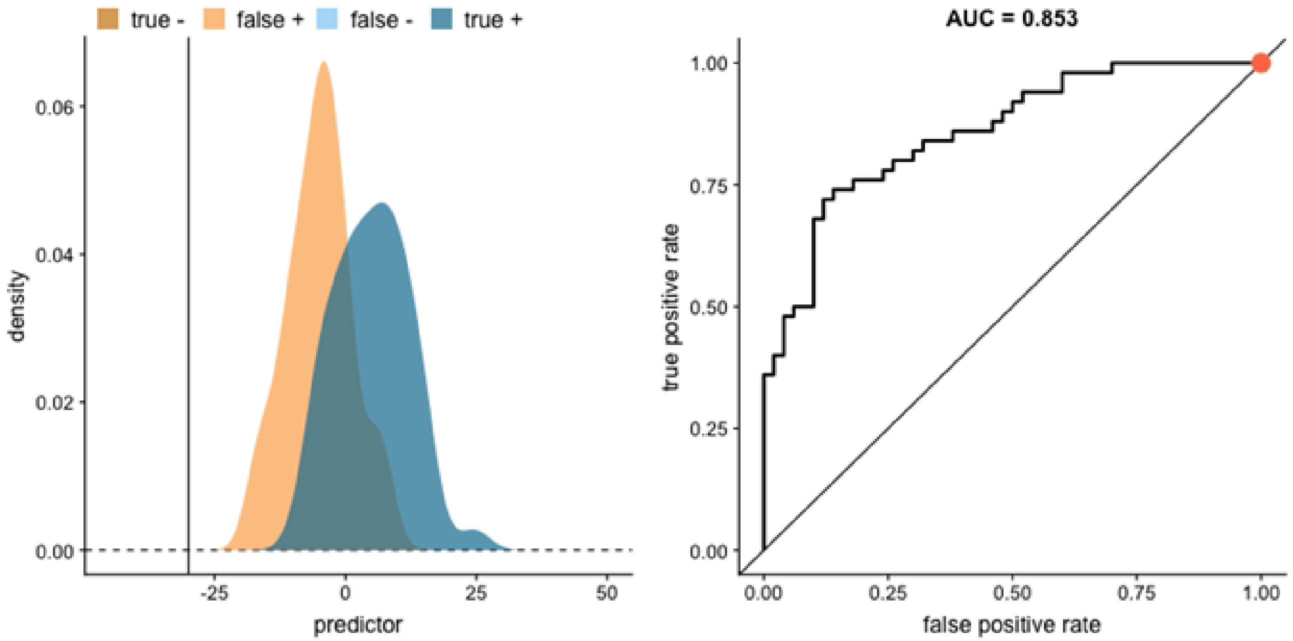


Image Source

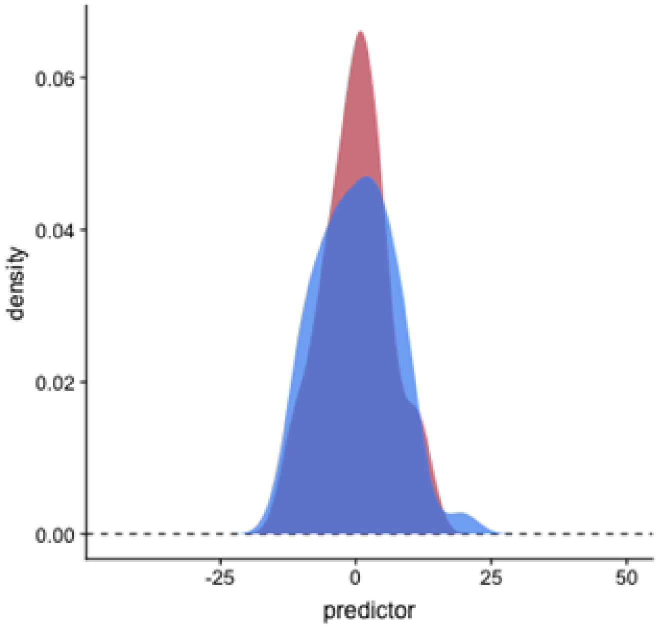
Threshold Independent Validation: AUC-ROC

- Area under the ROC (AUC) provides a single measure of model performance
- Independent of any particular choice of threshold
- The AUC ranges from 0 to 1

AUC = 1 indicates perfect discrimination between presence and absence/pseudo-absence locations

AUC = 0.5 discrimination is no better than random performance

AUC <0.5 performance is worse than random



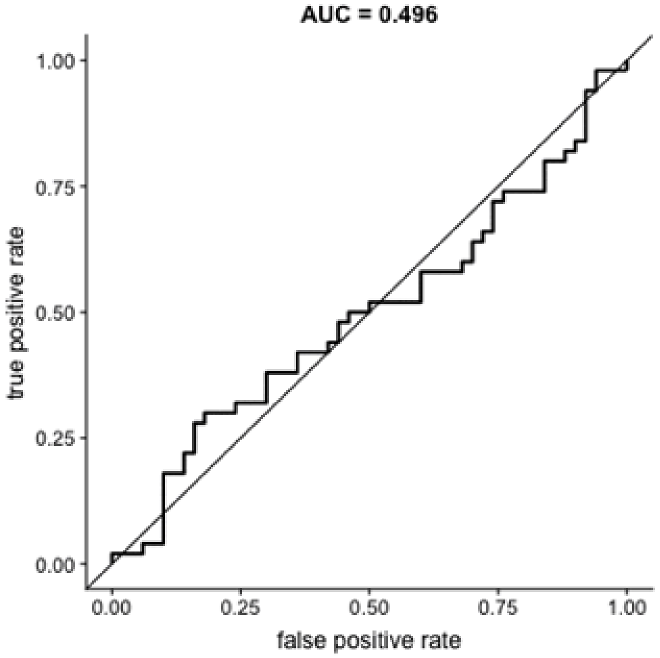
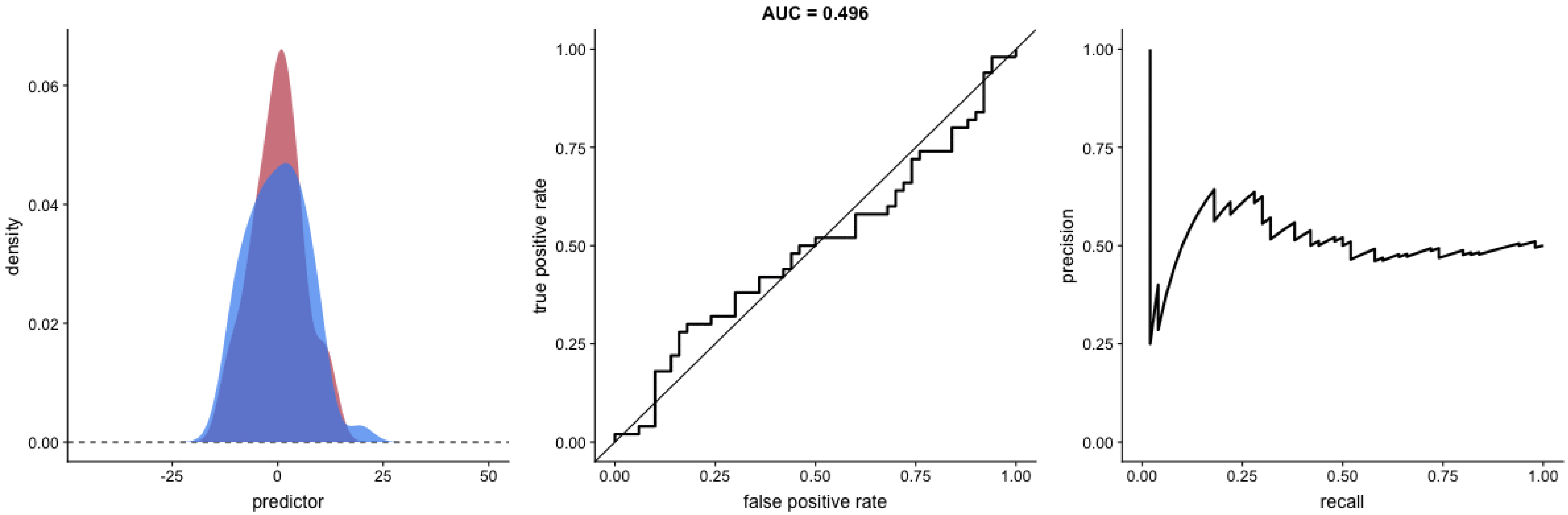


Image Source

Threshold Independent Validation: AUC-PR

- AUC-PR evaluates the model's ability to find true presences (Recall or Sensitivity) against the reliability of its suitability map (Precision) across all possible thresholds.



[Image Source](#)

The Problem of Class Imbalances

- AUC-ROC overestimates accuracy when the number of absences is much higher than the number of presences (e.g., rare species)
- Here we see 10,000 absences vs 50 presences

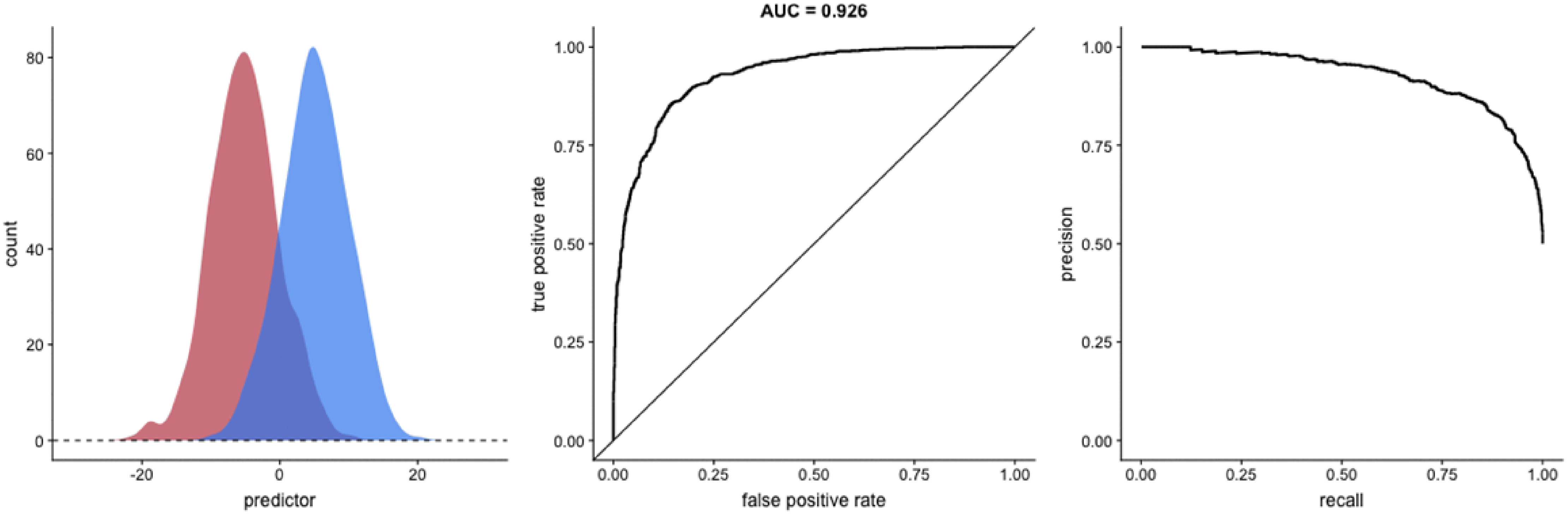


Image Source

Validating our Model

```
// Define partition for training and testing data
var split = 0.70;  // The proportion of the blocks used to select training data

// Define number of repetitions
var numiter = 5;

// While the runif function can be used to generate random seeds, we map the SDM
// function over random created numbers for reproducibility of results
var results = ee.List([35,68,43,54,17]).map(SDM);

// Extract results from list
var results = results.flatten();
//print(results)
```

```
var AUCROCs = ee.List.sequence(0,ee.Number(numiter).subtract(1),1).map(AUCROCaccuracy);
print('AUC-ROC:', AUCROCs);
print('Mean AUC-ROC', AUCROCs.reduce(ee.Reducer.mean()));
```

```
var AUCPRs = ee.List.sequence(0,ee.Number(numiter).subtract(1),1).map(AUCPRaccuracy);
print('AUC-PR:', AUCPRs);
print('Mean AUC-PR', AUCPRs.reduce(ee.Reducer.mean()));
```

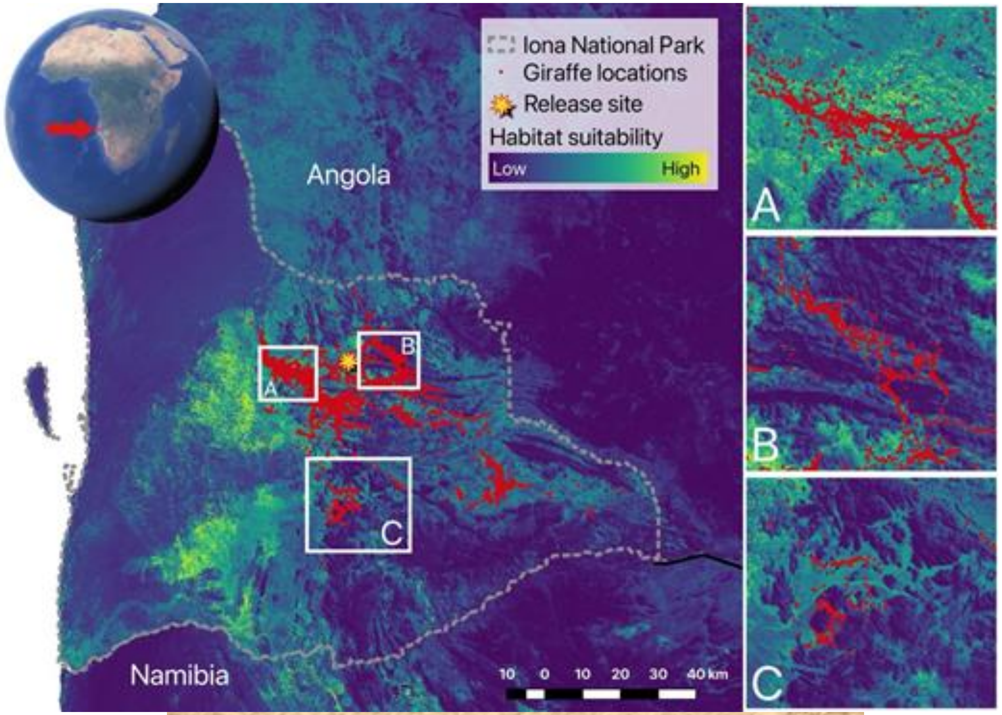
```
// Extract sensitivity, specificity and mean threshold values
var Metrics = ee.List.sequence(0,ee.Number(numiter).subtract(1),1).map(getMetrics);
print('Sensitivity:', ee.FeatureCollection(Metrics).aggregate_array("TPR"));
print('Specificity:', ee.FeatureCollection(Metrics).aggregate_array("TNR"));

var MeanThresh = ee.Number(ee.FeatureCollection(Metrics).aggregate_array("cutoff").reduc
print('Mean threshold:', MeanThresh);
```

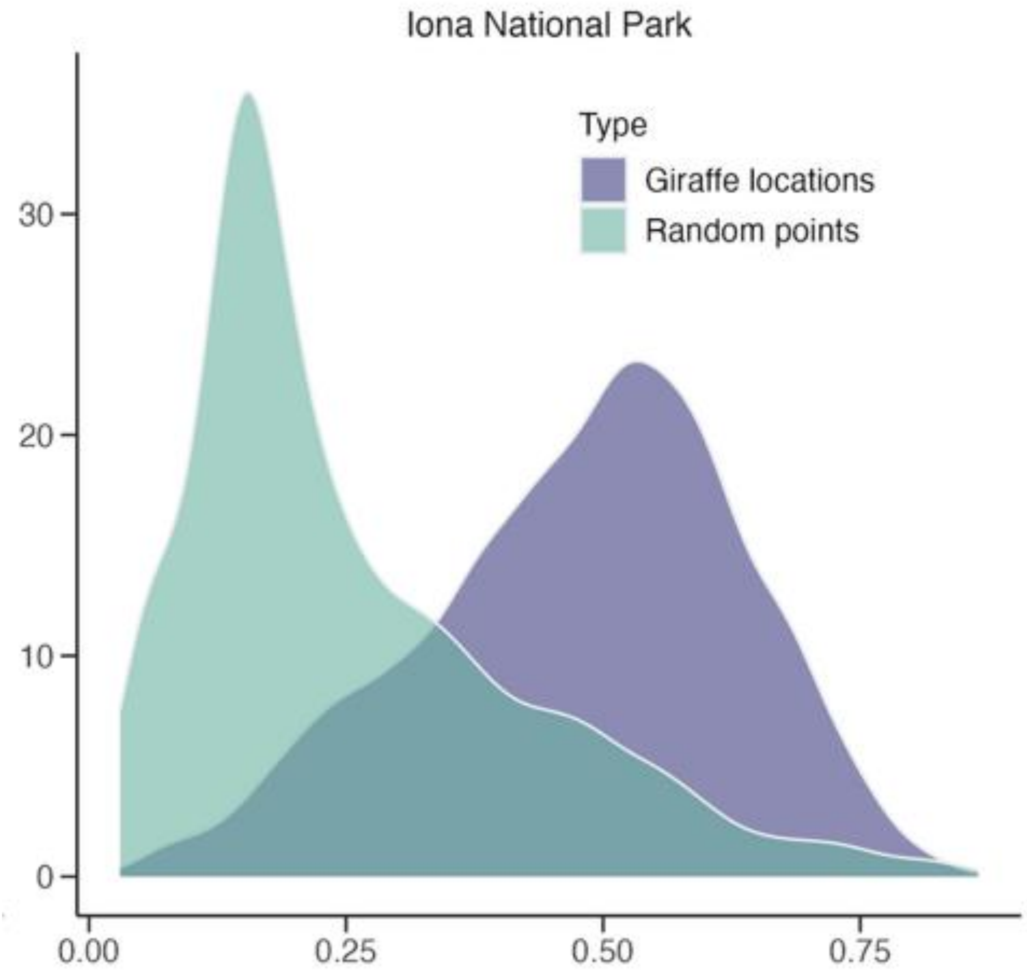
```
Number of presence and pseudo…                      JSON
▾[[49,49],[75,73],[190,119],[…                      JSON
  ▸0: [49,49]
  ▸1: [75,73]
  ▸2: [190,119]
  ▸3: [182,139]
  ▸4: [69,69]

AUC-ROC:                                            JSON
▾List (5 elements)                                  JSON
  0: 0.9403508771929826
  1: 0.9640777495284178
  2: 0.929691368212576
  3: 0.8446691176470589
  4: 0.9077457345613772

Mean AUC-ROC                                        JSON
0.9173069694284826

AUC-PR:                                             JSON
▾List (5 elements)                                  JSON
  0: 0.7904449098307224
  1: 0.8010769754187687
  2: 0.8324427108146504
  3: 0.8483621840917326
  4: 0.8416996331824128
```

```
Mean AUC-PR                                         JSON
0.8228052826676573

Sensitivity:                                        JSON
▾List (5 elements)                                  JSON
  0: 0.8043478260869565
  1: 0.7464788732394366
  2: 0.9320987654320988
  3: 0.9444444444444444
  4: 0.875

Specificity:                                        JSON
▾List (5 elements)                                  JSON
  0: 0.89
  1: 0.9343065693430657
  2: 0.7671232876712328
  3: 0.7058823529411765
  4: 0.8986486486486487

Mean threshold:                                     JSON
0.4583333333333333
```

Importance of Independent Validation Data

It is often critical to quantify the generality of a model (i.e., capacity of the model to predict in a different region and/or timeframe from the training data).





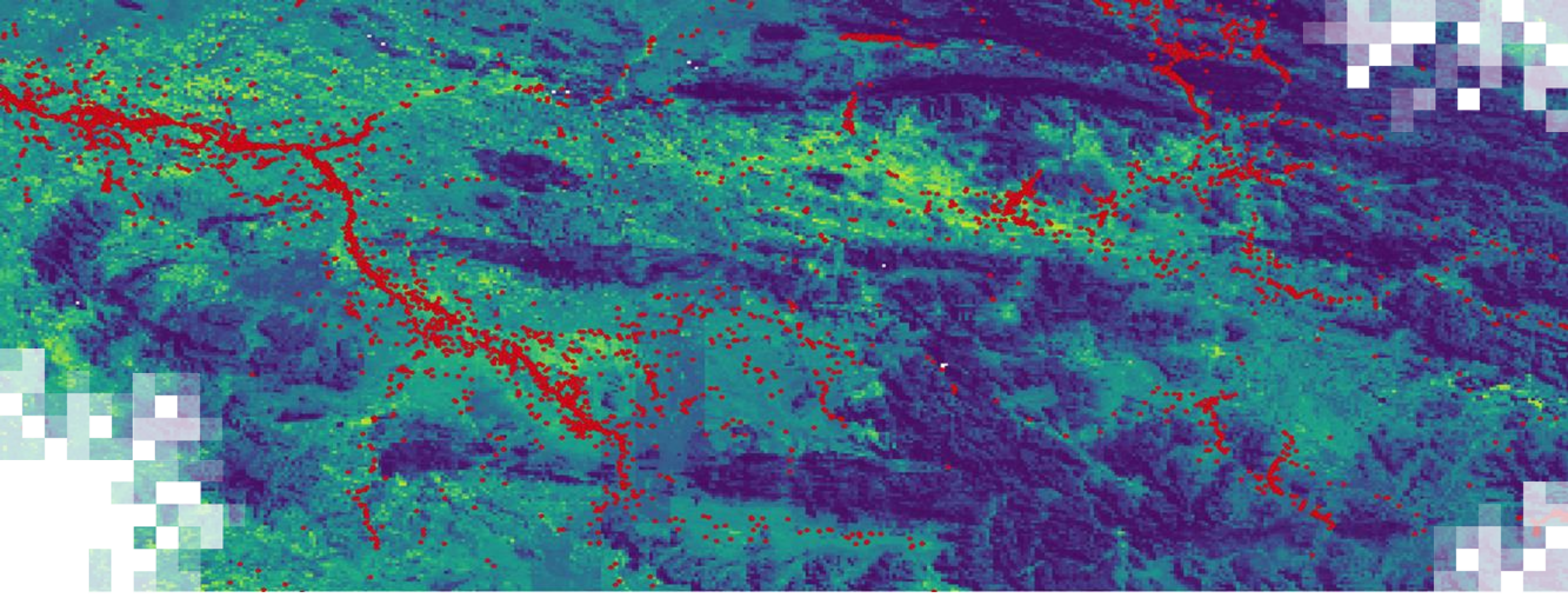
Importance of Independent Validation Data

- **Prevent overfitting:** Ensure the model is not 'memorizing' the training data but failing to predict at new locations.
- **Ensure transferability:** Proves the reliability of your model to predict habitat suitability across different geographic areas and time periods.
- **Assess true accuracy:** Validating models with the same data used for training (even when using cross-validation) can give the false impression of high performance.

Exporting Model Outputs: “Batch Mode”

```
//////////////////////////////////////////////////////////////////////
// Section 9 - Export outputs
//////////////////////////////////////////////////////////////////////

// Export final model predictions to drive

// Averaged habitat suitability
Export.image.toDrive({
  image: ModelAverage, //Object to export
  description: 'HSI', //Name of the file
  scale: GrainSize, //Spatial resolution of the exported raster
  maxPixels: 1e10,
  region: AOI //Area of interest
});

// Export final binary model based on a mayority vote
Export.image.toDrive({
  image: DistributionMap, //Object to export
  description: 'PotentialDistribution', //Name of the file
  scale: GrainSize, //Spatial resolution of the exported raster
  maxPixels: 1e10,
  region: AOI //Area of interest
});

// Export final binary model based on the threshold that maximises
// the sum of specificity and sensitivity
Export.image.toDrive({
  image: DistributionMap2.unmask(-9999),
  description: 'PotentialDistributionThreshold',
  scale: GrainSize,
  maxPixels: 1e10,
  region: AOI
});
```

```
// Export Accuracy Assessment Metrics

Export.table.toDrive({
  collection: ee.FeatureCollection(AUCROCs
                        .map(function(element){
                        return ee.Feature(null,{AUCROC:element})})),
  description: 'AUCROC',
  fileFormat: 'CSV',
});

Export.table.toDrive({
  collection: ee.FeatureCollection(AUCPRs
                        .map(function(element){
                        return ee.Feature(null,{AUCPR:element})})),
  description: 'AUCPR',
  fileFormat: 'CSV',
});

Export.table.toDrive({
  collection: ee.FeatureCollection(Metrics),
  description: 'Metrics',
  fileFormat: 'CSV',
});
```



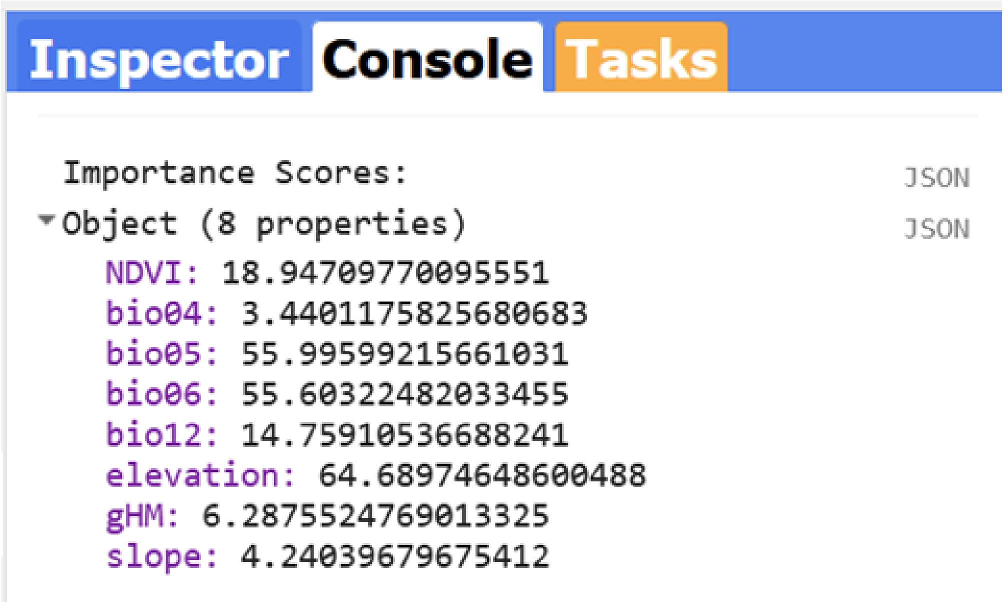
Modifying the Workflow to Get Variable Importance

Extracting Variable Importance

- Variable importance quantifies how effectively each environmental layer was able to sort presence points away from the absence points.
- These scores are a relative ranking meant to be compared against each other **within that specific model run** to see which predictors had the most influence.

Trained Classifier

```
// Variable importance
var variable_importance = ClassifierPr.explain().get('importance');
print("Importance Scores:", variable_importance);
```



Extracting Variable Importance within a K-fold Loop

[Rest of SDM Function Above]

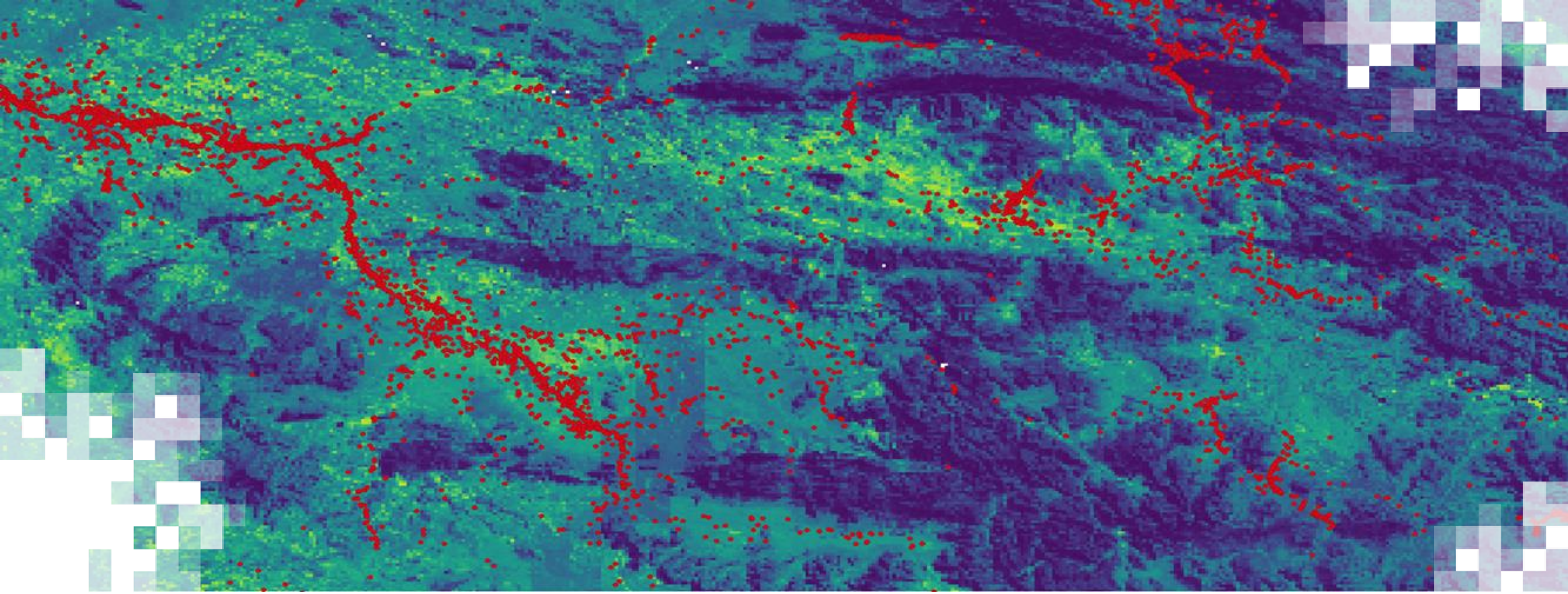
```
  // Variable importance
  var variable_importance = ee.Feature(null, ee.Dictionary(ClassifierPr.explain()).get('importance'));

  return ee.List([ClassifiedImgPr, ClassifiedImgBin, trainingPartition, testingPartition, variable_importance]);
}

// While the runif function can be used to generate random seeds, we map the SDM function over fixed random numbers
var results = ee.List([618,143,421,554,213]).map(SDM);

// Extract results from list
var results = results.flatten();

//////////////////////
// Variable importance

var VarImportance = ee.List.sequence(4,ee.Number(numiter).multiply(5).subtract(1),5).map(function(x){
                  return results.get(x)});
var Var_Imp = ee.FeatureCollection(VarImportance);
print(Var_Imp);

var Variable_Imp = Var_Imp.reduceColumns({
  reducer: ee.Reducer.mean().repeat(ee.List(bands).length()),
  selectors: bands
});

print('Variable importance', Variable_Imp);
//['bio04','bio05','bio06','bio12','elevation','slope','NDVI','gHM'];
```

Variable Importance Interpretation

- **Methodological Decisions Affect the Ranking**: For example, if you expand your study area to a continental scale, regional climate variables will look massively important. If you clip that exact same model down to a single national park or watershed, climate becomes relatively uniform across the points, and localized variables (like slope, aspect, or canopy structure) will be at the top of the importance ranking.

"Across this specific study area, and given our specific presence-absence dataset, precipitation was the most effective tool for distinguishing presence locations from absence locations."

"Precipitation is a more vital niche requirement for this species than soil type."



Additional Considerations

Model Evaluation Strategy

The Illusion of Predictive Power

Ploton et al. demonstrated that a standard random cross-validation showed an R^2>0.50, while spatial cross-validation revealed virtually null predictive power.

https://doi.org/10.1038/s41467-020-18321-y OPEN

Spatial validation reveals poor predictive performance of large-scale ecological mapping models

Pierre Ploton[1], Frédéric Mortier[2,3], Maxime Réjou-Méchain[1], Nicolas Barbier[1], Nicolas Picard[4], Vivien Rossi[5], Carsten Dormann[6], Guillaume Cornu[2,3], Gaëlle Viennois[1], Nicolas Bayol[7], Alexei Lyapustin[8], Sylvie Gourlet-Fleury[2,3] & Raphaël Pélissier[1]

The Illusion of Failure

Wadoux et al., 2021 argued that spatial CV punishes the model for poor extrapolation rather than measuring how good the map is. It also results in information loss by fitting the model without fully representative data.

ELSEVIER

Ecological Modelling

Volume 457, 1 October 2021, 109692

Short communication

Spatial cross-validation is not the right way to evaluate map accuracy

Alexandre M.J.-C. Wadoux[a], Gerard B.M. Heuvelink[b], Sytze de Bruin[c], Dick J. Brus[d]

Ensembling SDMs

- Can smooth over-prediction and under-prediction of individual models (e.g., Helmeted Hornbill example).
- However, it inherently averages out peak performance (best model averaged with worse models) and rarely beat the absolute "best" single model for a specific landscape.

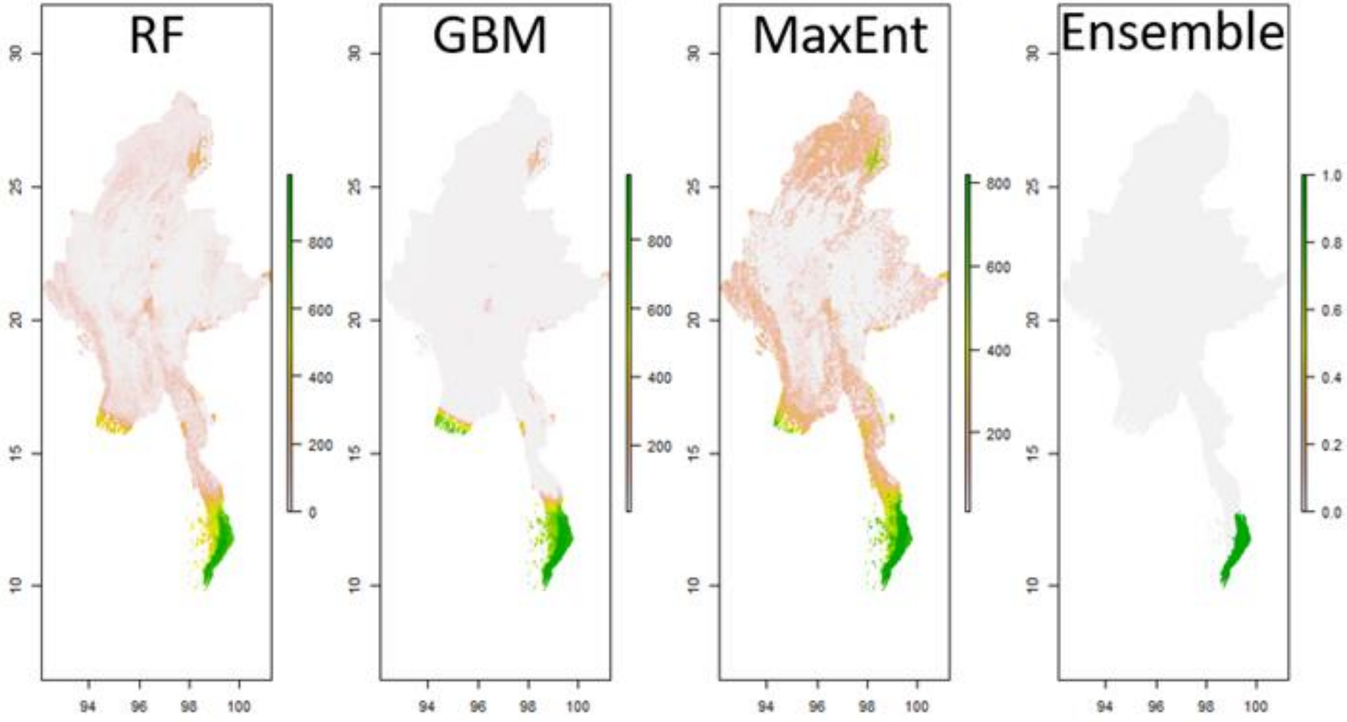


Photo credit: Craig Ansibin

Pitfalls & Best Practices: Part 1

Garbage In, Garbage Out: Avoid relying on unvetted, heavily clustered crowd-sourced presence data, or blindly throwing environmental layers into the model.

Ecological Predictor Curation: Select environmental layers based on known biological limiting factors rather than data convenience, and drop highly collinear variables.

Intentional Background Selection:

- If you have true absence data, use it!
- Absence ≠ Pseudo-absence ≠ Background
- Define your pseudo-absences strategically (e.g., using areas potentially reachable)

Objective-Driven Validation: Match your validation strategy to your goal; e.g., use random sampling to verify local map accuracy, but use spatial cross-validation if you are testing the model's regional transferability (e.g., the giraffe example).

Pitfalls & Best Practices: Part 2

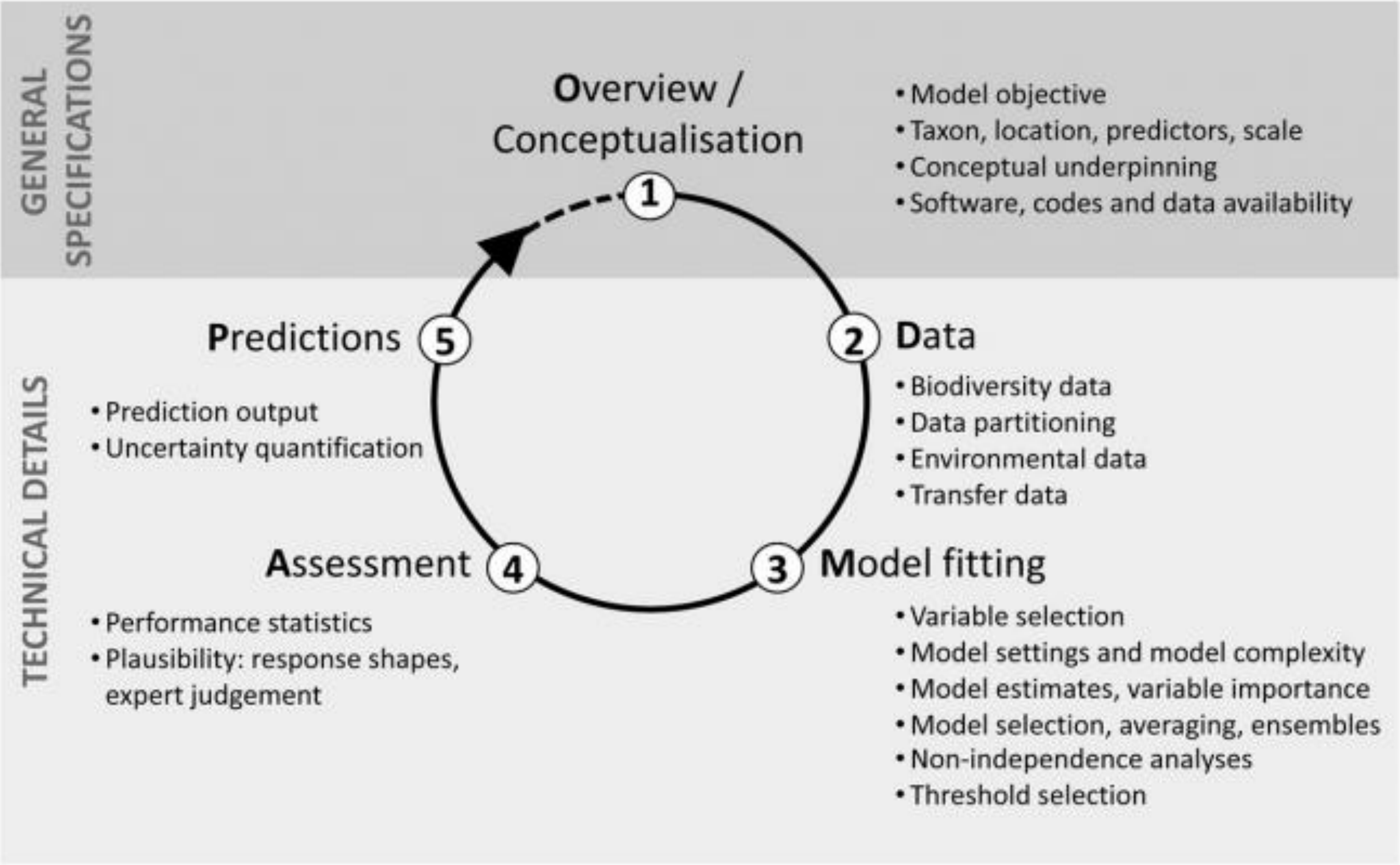
Ensemble-Based Prediction: Avoid relying on a single "winning" cross-validation fold for your final outputs and average the continuous probability maps across all your folds to smooth out localized mathematical noise.

Model Extrapolation Risks: Many algorithms cannot extrapolate outside their training range. You can run a MESS (Multivariate Environmental Similarity Surfaces) analysis when projecting onto new geographic areas or future climates to flag where predictions may be unreliable.

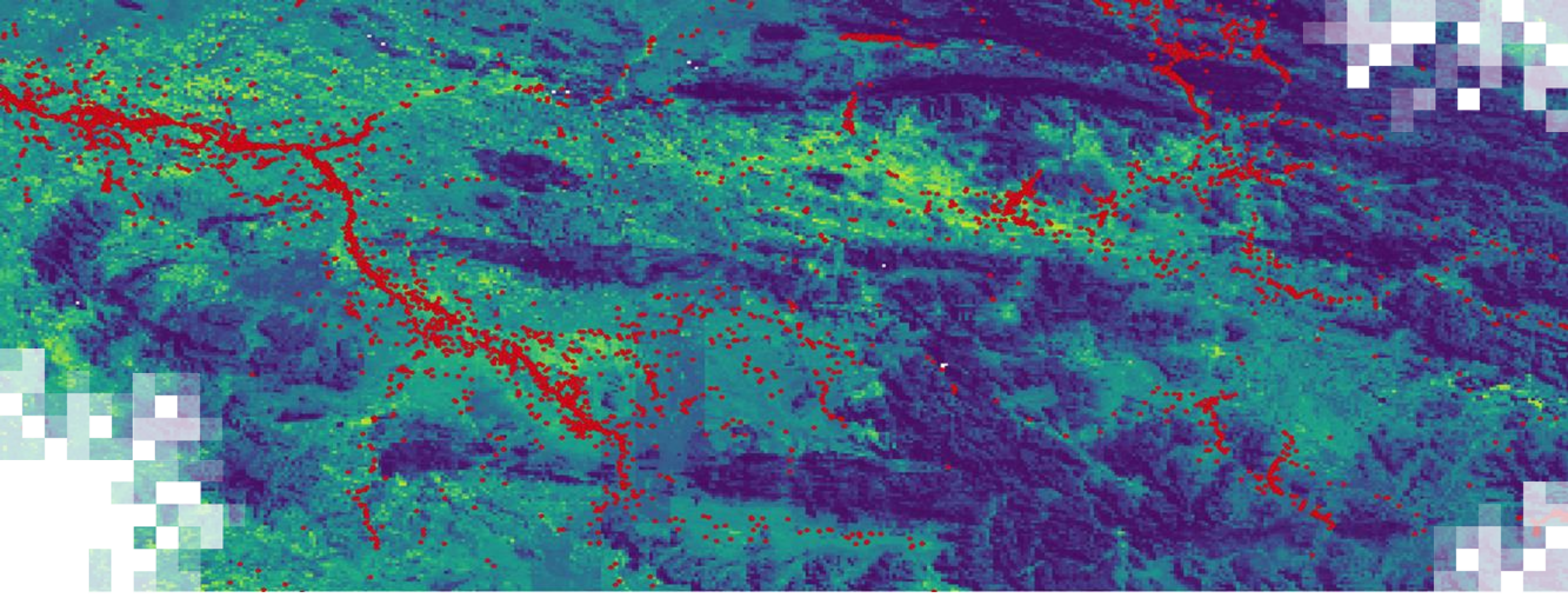
Continuous Probability Over Binary Thresholds: Prioritize analyzing and reporting the raw, continuous habitat suitability gradients rather than over-relying on arbitrary binary presence/absence thresholds that discard meaningful ecological detail.

Suitability ≠ Presence

The ODMAP


GENERAL SPECIFICATIONS
TECHNICAL DETAILS
Overview / Conceptualisation
1
• Model objective
• Taxon, location, predictors, scale
• Conceptual underpinning
• Software, codes and data availability
2 Data
• Biodiversity data
• Data partitioning
• Environmental data
• Transfer data
3 Model fitting
• Variable selection
• Model settings and model complexity
• Model estimates, variable importance
• Model selection, averaging, ensembles
• Non-independence analyses
• Threshold selection
Assessment 4
• Performance statistics
• Plausibility: response shapes, expert judgement
Predictions 5
• Prediction output
• Uncertainty quantification


ECOGRAPHY
Review and synthesis
A standard protocol for reporting species distribution models
Damaris Zurell, Janet Franklin, Christian König, Phil J. Bouchet, Carsten F. Dormann, Jane Elith, Guillermo Fandos, Xiao Feng, Gurutzeta Guillera-Arroita, Antoine Guisan, José J. Lahoz-Monfort, Pedro J. Leitão, Daniel S. Park, A. Townsend Peterson, Giovanni Rapacciuolo, Dirk R. Schmatz, Boris Schröder, Josep M. Serra-Diaz, Wilfried Thuiller, Katherine L. Yates, Niklaus E. Zimmermann and Cory Merow
EDITOR'S CHOICE



Summary

Resources

- Original manuscript

METHOD

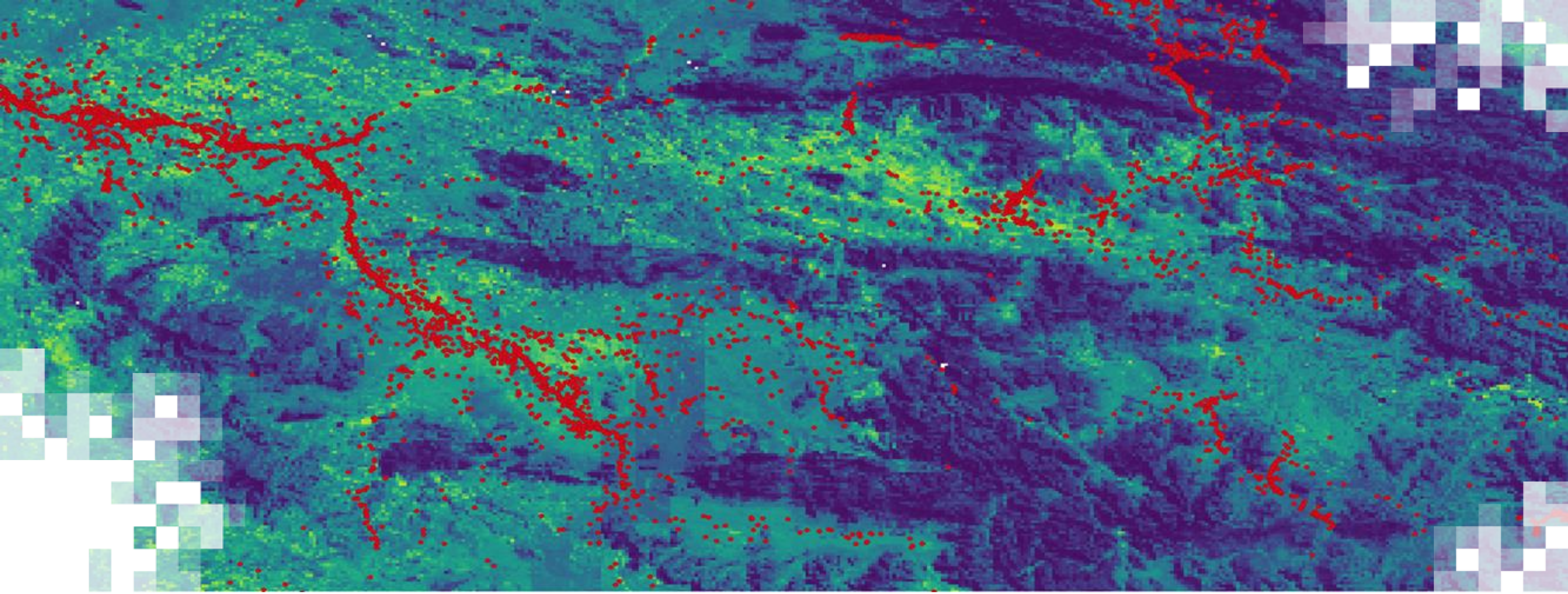
WILEY

Implementation of species distribution models in Google Earth Engine

Ramiro D. Crego[1,2] | Jared A. Stabach[2] | Grant Connette[1,2]

Abstract

- The Google Earth Engine code and data used in this workshop and manuscripts are freely available at this Google Earth Engine repository.



Species Distribution Modeling with Google Earth Engine

Part 2: Building, Evaluating, and Interpreting an SDM on GEE

Summary

Part 2 Training Summary

- Demonstrated end-to-end steps to build machine learning models within Google Earth Engine.
- Reviewed initial steps for uploading species data and essential preprocessing steps.
- Covered spatial thinning techniques for presence data and methods to define modeling Areas of Interest (AOIs).
- Evaluated criteria for choosing environmental predictors and accessed spatial data directly from the GEE catalog.
- Analyzed the concepts of true presence/absence data and techniques for generating pseudo-absence points.
- Contrasted methods for quantifying model accuracy (interpolation) versus model generality (extrapolation).
- Explored how to identify and rank the environmental layers most influential in predicting species distribution.
- Emphasized aligning model evaluation strategies directly with overarching project objectives.
- Pitfalls & Best Practices

Homework and Certificates

- **Homework:**
  - One homework assignment
  - Opens on 07/14/2026
  - Access from the training webpage
  - Answers must be submitted via Google Forms
  - **Due by 07/28/2026**

- **Certificate of Completion:**
  - Attend two live webinars (attendance is recorded automatically)
  - Complete the homework assignment by the deadline
  - You will receive a certificate via email approximately two months after completion of the course.

Contact Information

Hosts:

- Justin Fain
  - justin.j.fain@nasa.gov
- Sativa Cruz
  - sativa.cruz@nasa.gov

- ARSET Website
- ARSET YouTube

For questions, comments, or to share how you have applied our trainings to your work or studies, email nasa.arset@gmail.com.

Join our mailing list to stay up-to-date on our latest trainings. Visit our Contact page to subscribe.

Share Your Thoughts!

Within a day or two, we will send you an invitation to complete a short online survey.* **Your feedback helps us improve the ARSET program.**

- Participation is optional, and all responses are confidential.
- **Survey data are crucial to helping us understand how to better meet your needs. We want to help you use Earth observation data more effectively.**
- We read every survey comment – so please share your thoughts!

*The survey will come from noreply@alchemer.com. This is not spam.

Thank You!