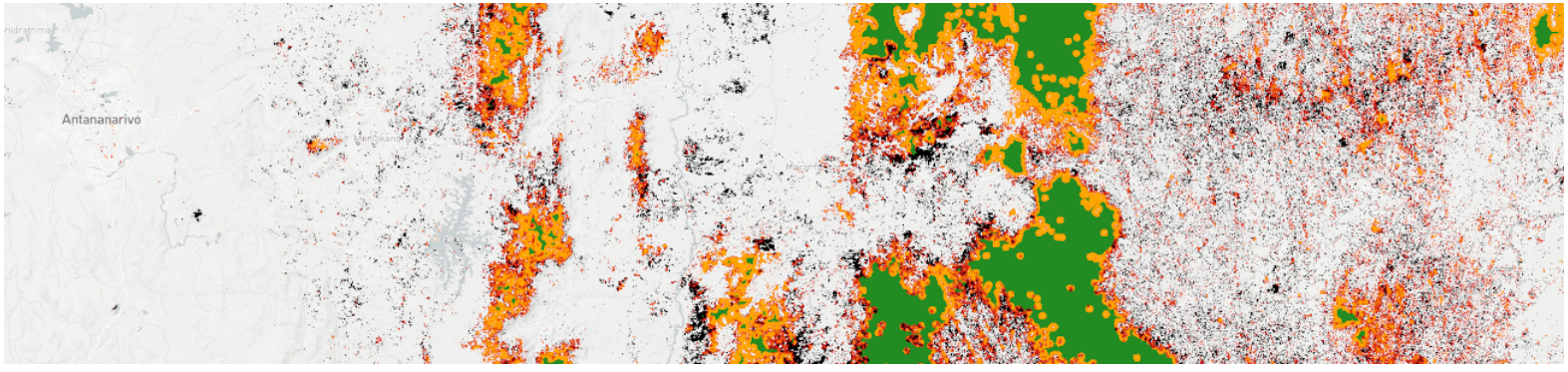
FAO – IMPRESS project – July 2022

`riskmapjnr` Python package for mapping the deforestation risk using JNR's methodology



Ghislain VIEILLEDENT[1] Pierrick RAMBAUD[2] Rémi d'ANNUNZIO[2]

[1] **Cirad** UMR AMAP, [2] **FAO** REDD+ NFM

Plan

1. Introduction
   - Context
   - Objectives
2. Functionalities
   - Python package
   - Functions
3. Case-studies
   - Jurisdictions
   - Kenya
4. Perspectives
   - Comments
   - Alternative approach

Plan

1. Introduction
   - Context
   - Objectives
2. Functionalities
   - Python package
   - Functions
3. Case-studies
   - Jurisdictions
   - Kenya
4. Perspectives
   - Comments
   - Alternative approach

Context

- Paris Agreement on climate change
- REDD+ : Reducing Emissions from Deforestation and forest Degradation
- IMPRESS (Improving Measurement for Payments to Reduce Emissions and Strengthen Sinks) FAO – UK-PACT project
- VCS Jurisdictional and Nested REDD+ (JNR) : certification of jurisdictional REDD+ programs and nested projects

Objectives

Allocate the deforestation spatially

- Given a deforestation intensity (ha/yr) in a jurisdiction, how to allocate deforestation spatially ? ⇒ Map of the deforestation risk.
- JNR risk mapping methodology, by Verra and CBI (Carbon Decision International).
- Simple methodology : use only an historical forest cover change map.

Informatic tool to derive the risk map

- Develop a tool (Python package) to derive this map.
- Following JNR methodology.
- Port that tool to Sepal (FAO side).

Jurisdictional & Nested REDD+

Plan

1 Introduction
- Context
- Objectives

2 Functionalities
- Python package
- Functions

3 Case-studies
- Jurisdictions
- Kenya

4 Perspectives
- Comments
- Alternative approach

Python package and website

- Python package : `riskmapjnr`
- Website : https://ecology.ghislainv.fr/riskmapjnr
- GitHub repository with open source code : https://github.com/ghislainv/riskmapjnr
- Tutorials : see *Get Started* and *Articles* sections on the website

Figure – `riskmapjnr` logo

Code efficiency

Fast computations

Python scientific libraries used :

- `gdal` for fast processing of georeferenced data.
- `NumPy`, `SciPy`, and `Pandas` for fast matrix and vector operations.

Handling large rasters

- Large rasters are divided into blocks of data for in-memory processing.
- Analysis on large geographical extents (e.g. country scale) and high spatial resolutions (eg. 30 m).

Repeated tasks can be parallelized

- Several ($25 \times 3 = 75$) maps need to be produced and compared.
- Function to produce maps on separate computer cores in parallel.

Main functions

The `riskmapjnr` package includes functions to :

1. Estimate the distance to forest edge beyond which the deforestation risk is negligible : `dist_edge_threshold()`.
2. Compute local deforestation rates using a moving window whose size can vary : `local_defor_rate()`.
3. Transform local deforestation rates into categories of deforestation risks using several slicing algorithms : `set_defor_cat_zero()` and `defor_cat()`
4. Validate maps of deforestation risk and select the map with the higher accuracy : `defrate_per_cat()` and `validation()`.

Distance to forest edge threshold

- `rmj.dist_edge_threshold()` : Compute the distance to the forest edge after which the risk of deforestation becomes negligible.
- Here, $>$99% of deforestation occurs within a distance $\leq$180m.
- Forest pixels with a distance $>$180m will be in Category 0 (zero risk of deforestation).

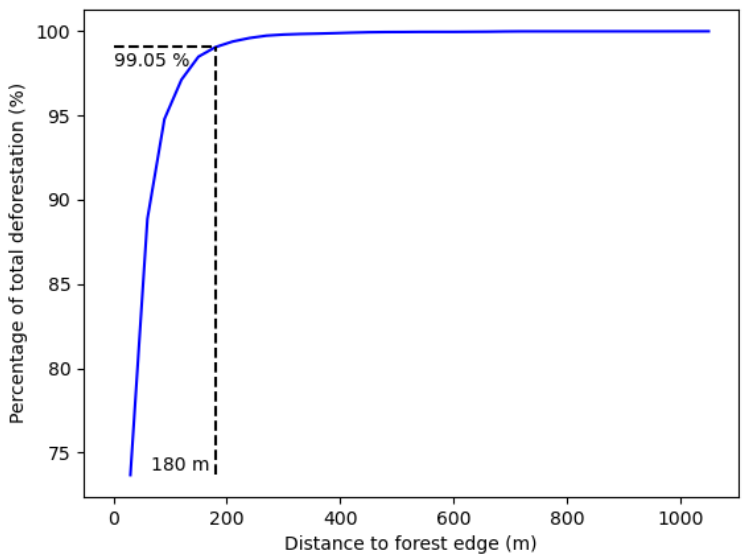


Figure – Cumulative deforestation as a function of the distance to forest edge.

Local deforestation rate

- `rmj.local_defor_rate()` : Compute a local risk of deforestation at the pixel level using a moving window made of several pixels.
- Different window sizes can be chosen.
- The JNR methodology recommends the use of 25 different window sizes.

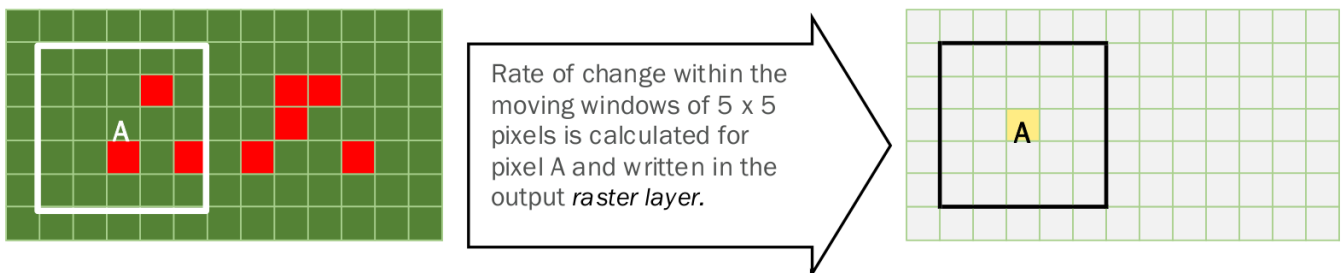


Figure – Moving window.

Categorize the deforestation risk

- `rmj.defor_cat()` : Convert local deforestation rate into categories of deforestation risk.
- The JNR methodology suggests to use 31 categories of risk from "0" to "30" (including the "0" category).
- The JNR methodology recommends the use of three slicing algorithms : "equal area", "equal interval", and "natural breaks".
  - "equal area" : each class covers approximately the same area
  - "equal interval" : bins of the same range size
  - "natural breaks" : data are normalized before applying the "equal interval" algorithm.

0	0	0	0	0	0	1	1	2	2	2	1	2
0	0	0	0	0	1	1	2	3	4	3	3	2
0	0	0	1	2	2	2	4	5	8	8	2	4
0	0	1	1	3	1	3	4	10	9	7	3	5
0	0	1	1	2	2	4	5	3	8	6	6	7
0	1	1	1	2	3	3	6	7	8	10	9	8
0	0	1	1	2	1	2	3	6	5	4	3	10

Figure – Categories of deforestation risk.

Validate the map

- `rmj.validation()` : Validate the map of deforestation risk on a validation period.
- Square grid of at least 1000 spatial cells covering the jurisdiction.
- Predicted deforestation using deforestation rates for risk categories.
- Comparison of predictions and observations for each spatial cells
- Accuracy index : weighted Root Mean Squared Error (wRMSE)

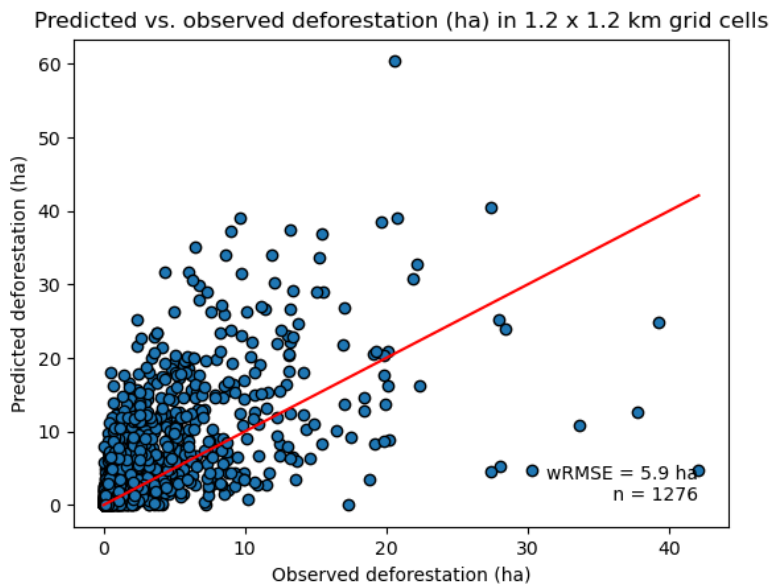


Figure – Predictions vs. observations.

Derive maps in parallel

- `rmj.makemap()` : Derive maps with different window sizes and slicing algorithms and choose the best map.
- Maps are produced on separate computer cores in parallel.

Plan

1 Introduction
- Context
- Objectives

2 Functionalities
- Python package
- Functions

3 Case-studies
- Jurisdictions
- Kenya

4 Perspectives
- Comments
- Alternative approach

Jurisdictions

- Guadeloupe (*Get Started* tutorial)
- Madagascar tropical moist forests
- Kenya (IMPRESS project)
- more to come...

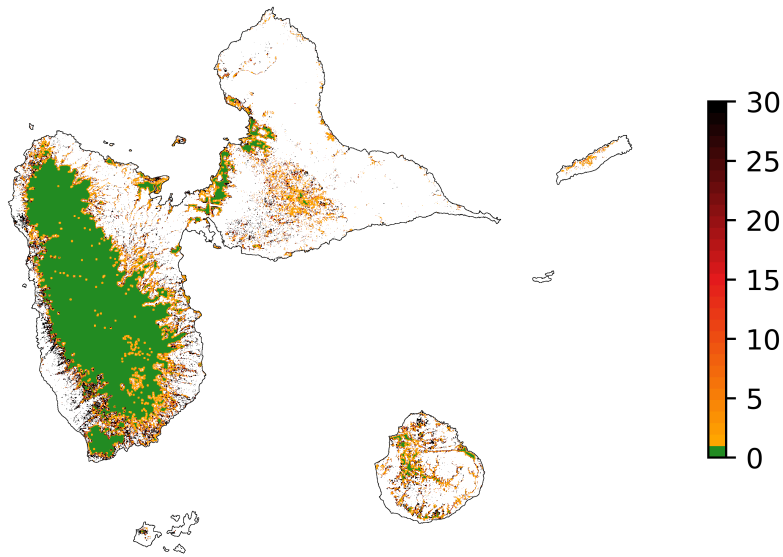


Figure – Map of the deforestation risk for Guadeloupe.

Kenya

- Forest cover change map : 2010–2014–2018.
- Distance to forest edge threshold : 780 m.
- Computation time : ∼20 min for 8 window sizes and 2 slicing algorithms on a personal computer using 6 cores.

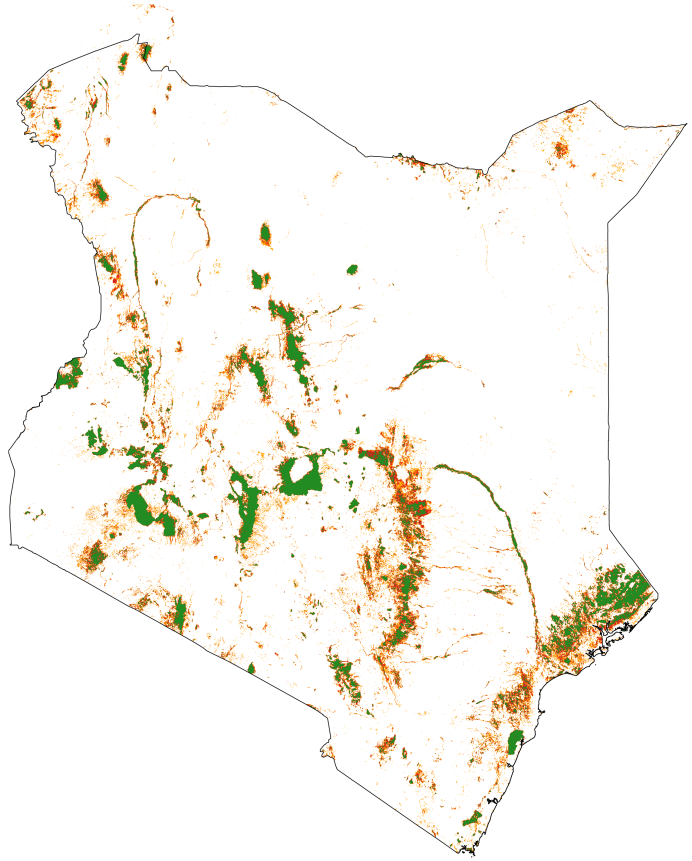


Figure – Forest cover change (2010–2014–2018) for Kenya.

Kenya

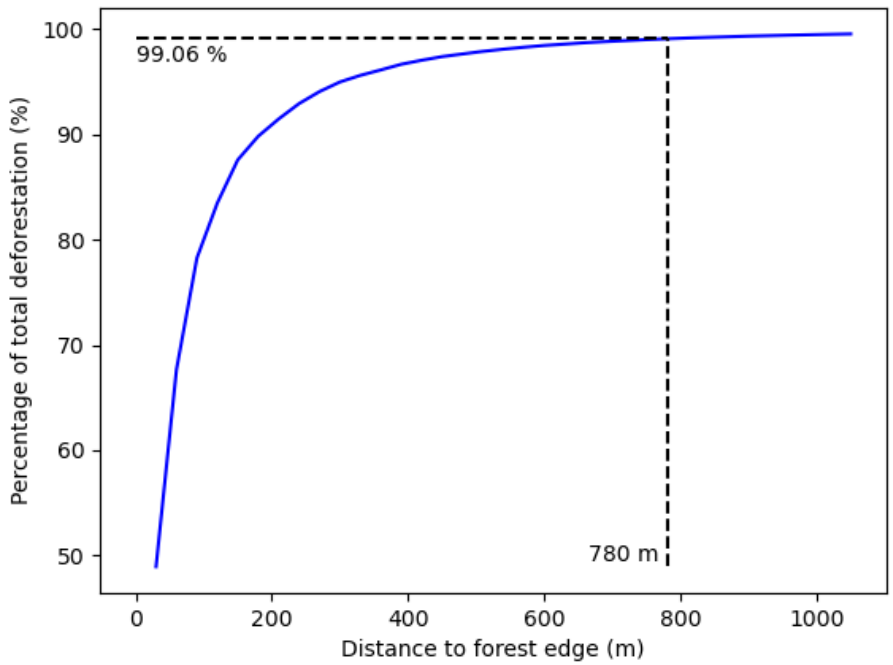


Figure – Cumulative deforestation as a function of the distance to forest edge for Kenya.

Kenya

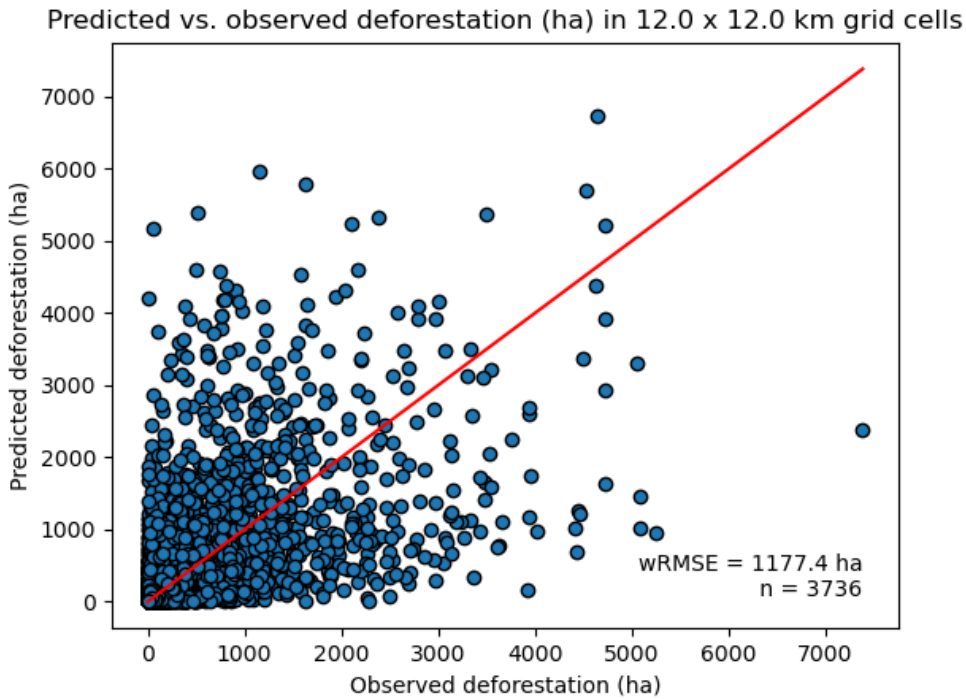


Figure – Predictions vs. observations for Kenya.

Kenya

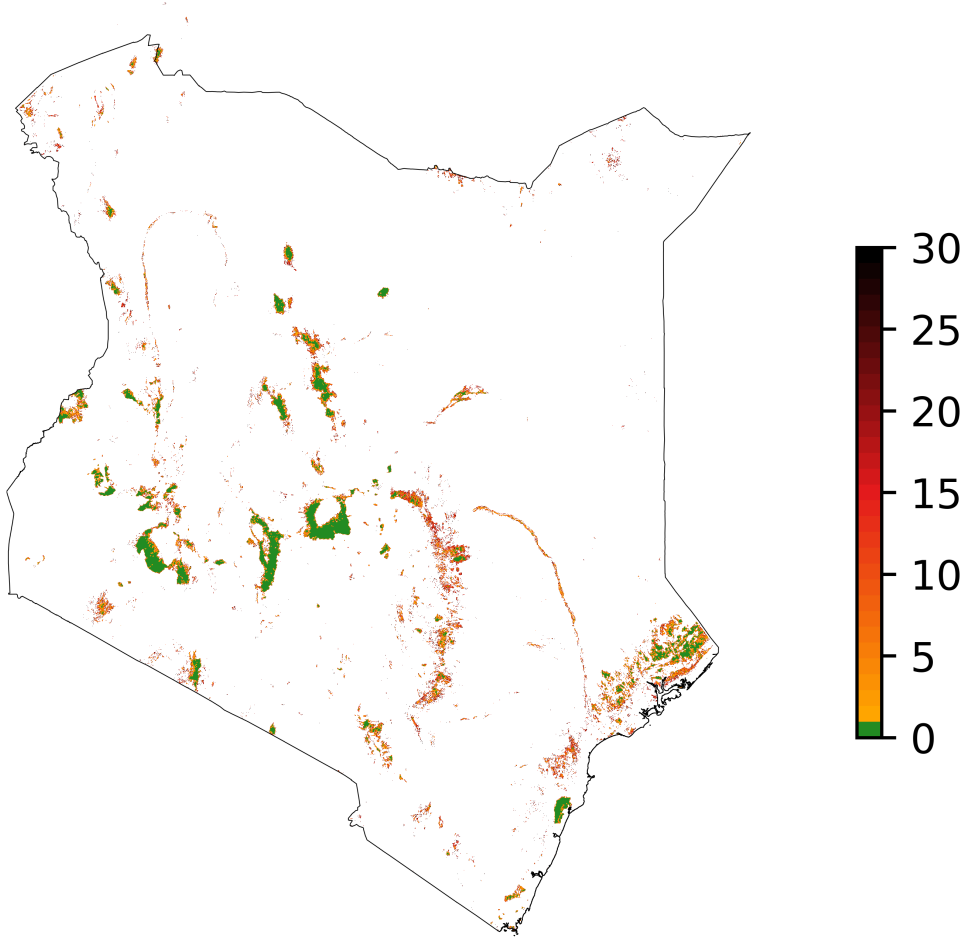


Figure – Map of the deforestation risk for Kenya.

Plan

1. Introduction
   - Context
   - Objectives
2. Functionalities
   - Python package
   - Functions
3. Case-studies
   - Jurisdictions
   - Kenya
4. Perspectives
   - Comments
   - Alternative approach

Additional tests

- ! ! First results
- Code might include some errors
- Functions still need to be thoroughly tested
- Results must be consolidated

Issues

- The best window size is always the smallest.
- No differences between slicing algorithms (ei or ea).
- ei : “equal interval”
  ea : “equal area”.
- The “natural breaks” algorithm is not yet implemented.

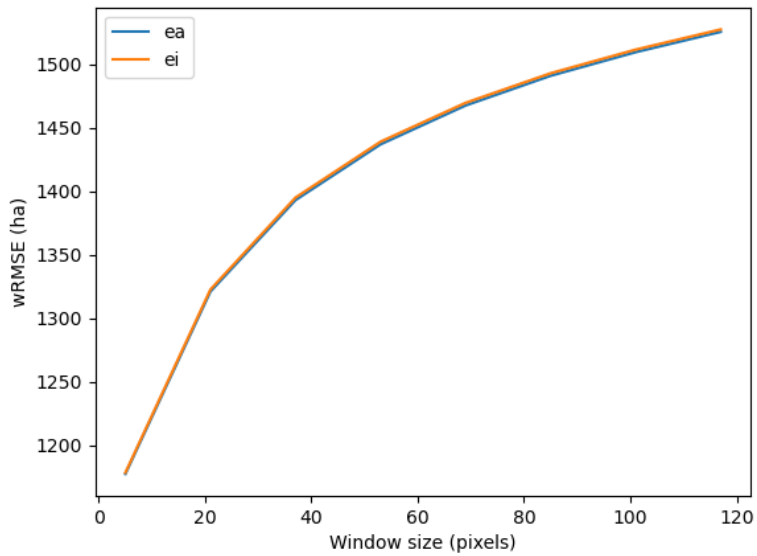


Figure – Prediction error as a function of window size.

Issues

- Weak relationship between predictions and observations (high wRMSE).

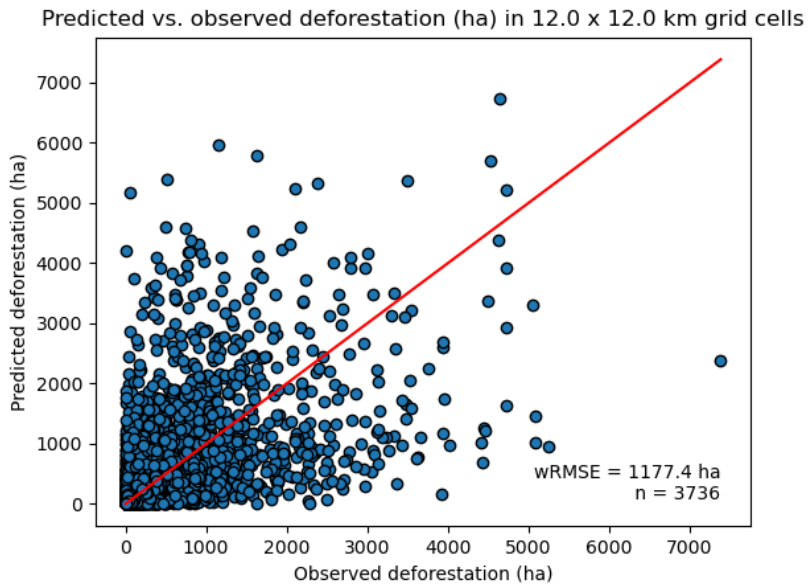


Figure – Predictions vs. observations for Kenya.

Discussions with partners

- Cirad, FAO, IMPRESS, Verra and CBI.
- To improve the methodology itself.
- To test the `riskmapjnr` package and have feedbacks.
- To increase computational speed on Sepal (use of GPU).

Alternative approach

- Comparison with the `forestatrisk` approach
- Statistical model estimating the deforestation risk θ
- θ = function(environmental variables + location)
- Variables : distance to forest edge, roads, towns, protected areas

https://ecology.ghislainv.fr/forestatrisk/

. . . Thank you for attention . . .

https://ecology.ghislainv.fr/presentations

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

cirad

AMAPlab

FAO

UK PACT