living planet symposium | BONN 23–27 May 2022

TAKING THE PULSE OF OUR PLANET FROM SPACE

EUMETSAT ECMWF EUSPA

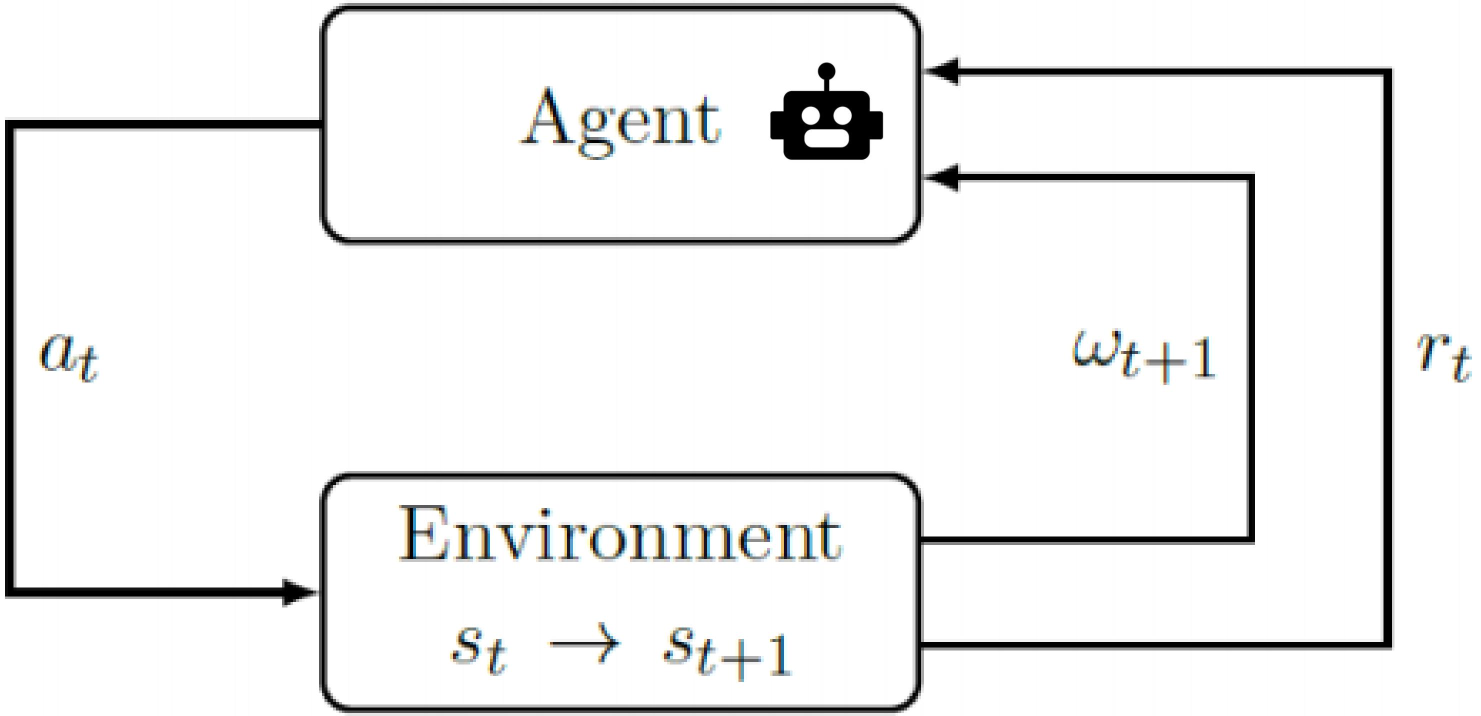
Towards an artificial intelligence framework for ecosystem restoration planning

Julián Equihua, Prof. Dr. Ralf Seppelt, Dr. Michael Beckmann

UFZ HELMHOLTZ Zentrum für Umweltforschung

24.05.2022

Reinforcement learning

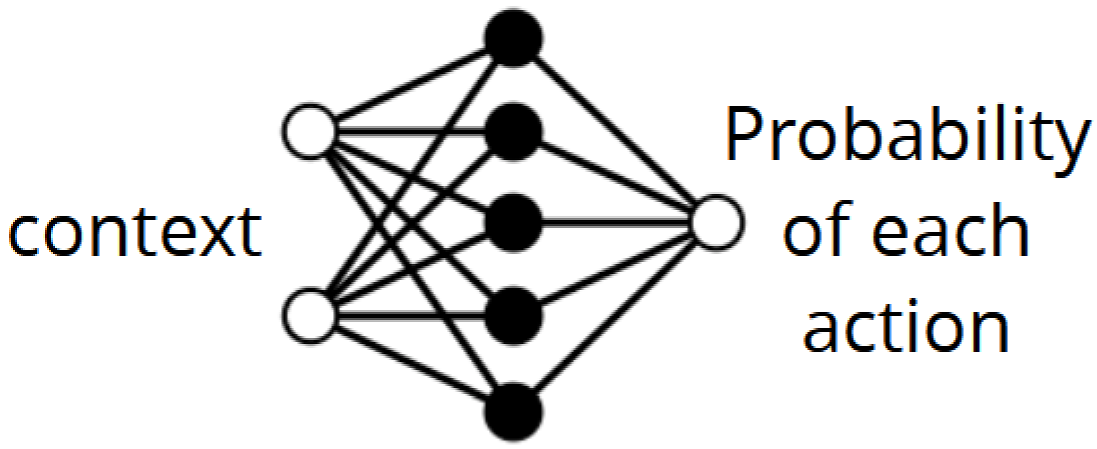


RL vs DRL

RL-table

State	Action
context1	action1
context2	action2
context3	action3
context4	action4

DRL-deep NN



Software considerations

OpenAI Baselines

Dopamine

rllib

Chainer RL

DeepMind TRFL

- Ease of use
- Scalability
- Active development
- Environment agnostic

Software considerations

We need the gym.Env class

Gym

Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.

The init() method. Which in turn must initialize two required members as Gym spaces:

•self.action_space – the action space of possible actions taken by the agent
•self.observation_space – the observation space for what info the agent receives after taking an action

The reset() method. This resets the state of the environment for a new episode and also returns an initial observation

The step() method. Handles how an agent takes an action during one step in an episode.

The render() method. Allows to visualize the state of the environment.

The seed() method. Allows to set a seed for environments pseudorandom number generator.

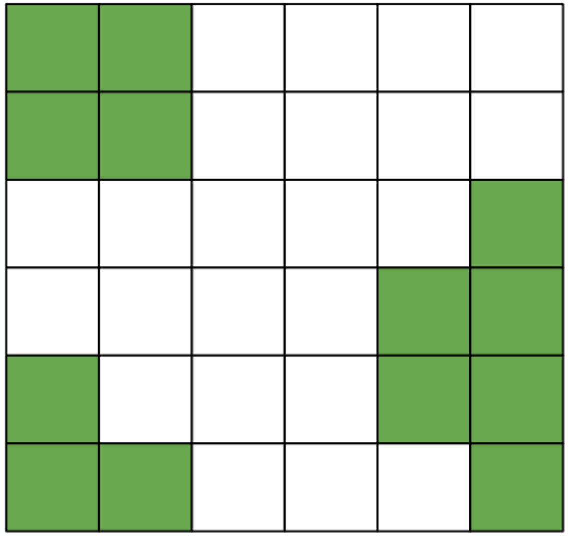
The close() method. Defines how to handle closing an environment.

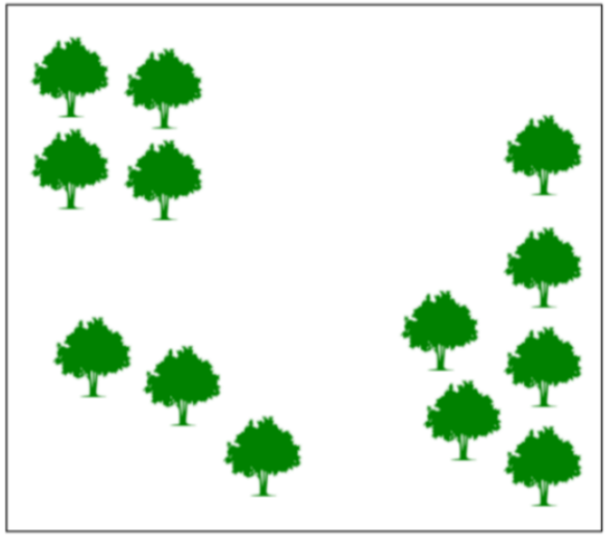
Improving connectivity through habitat restoration

Lets define a biological corridor as a structure which allows the transit of individuals between large or small populations.

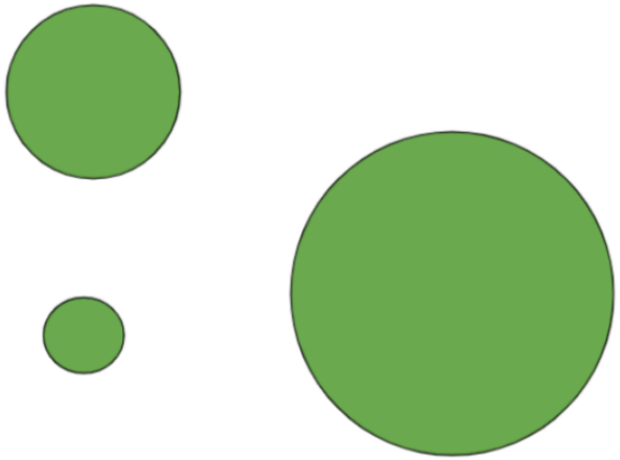
Sites that allow the transit between populations are essential for promoting and maintaining their genetic health.

Landscape

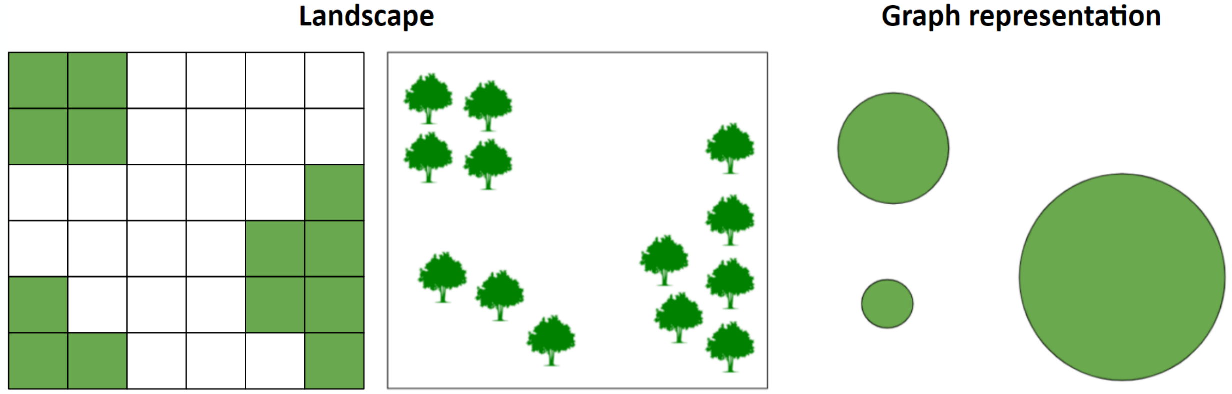




Graph representation



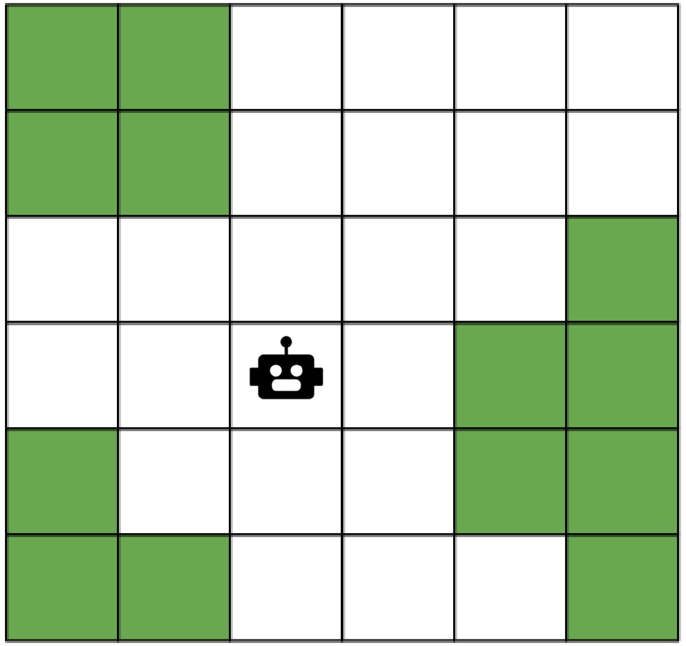
Improving connectivity through habitat restoration

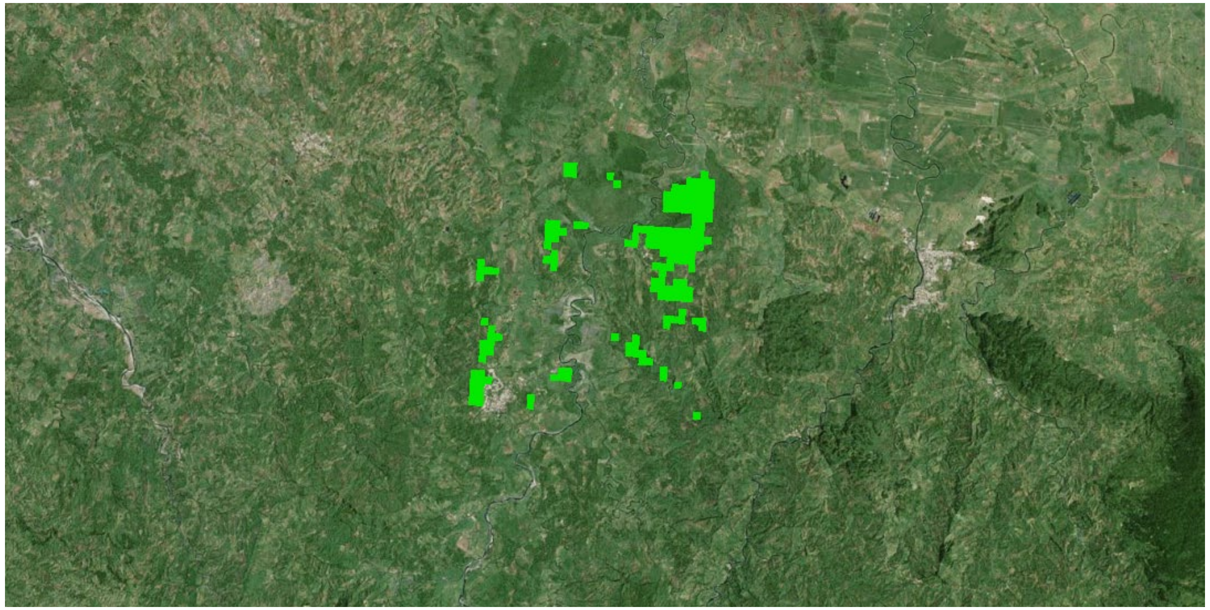


$$PC = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} a_i * a_j * p_{ij}^*}{A_L^2}$$

- a_i and a_j are the areas of the habitat patches i and j.
- A_L is the total landscape area.
- p_{ij}^* is defined as the maximum product probability of all possible paths between patches i and j. The product probability of a path (where a path is made up of a set of steps in which no patch is visited more than once) is the product of all the p_{ij} belonging to each step in that path.

Improving connectivity through habitat restoration

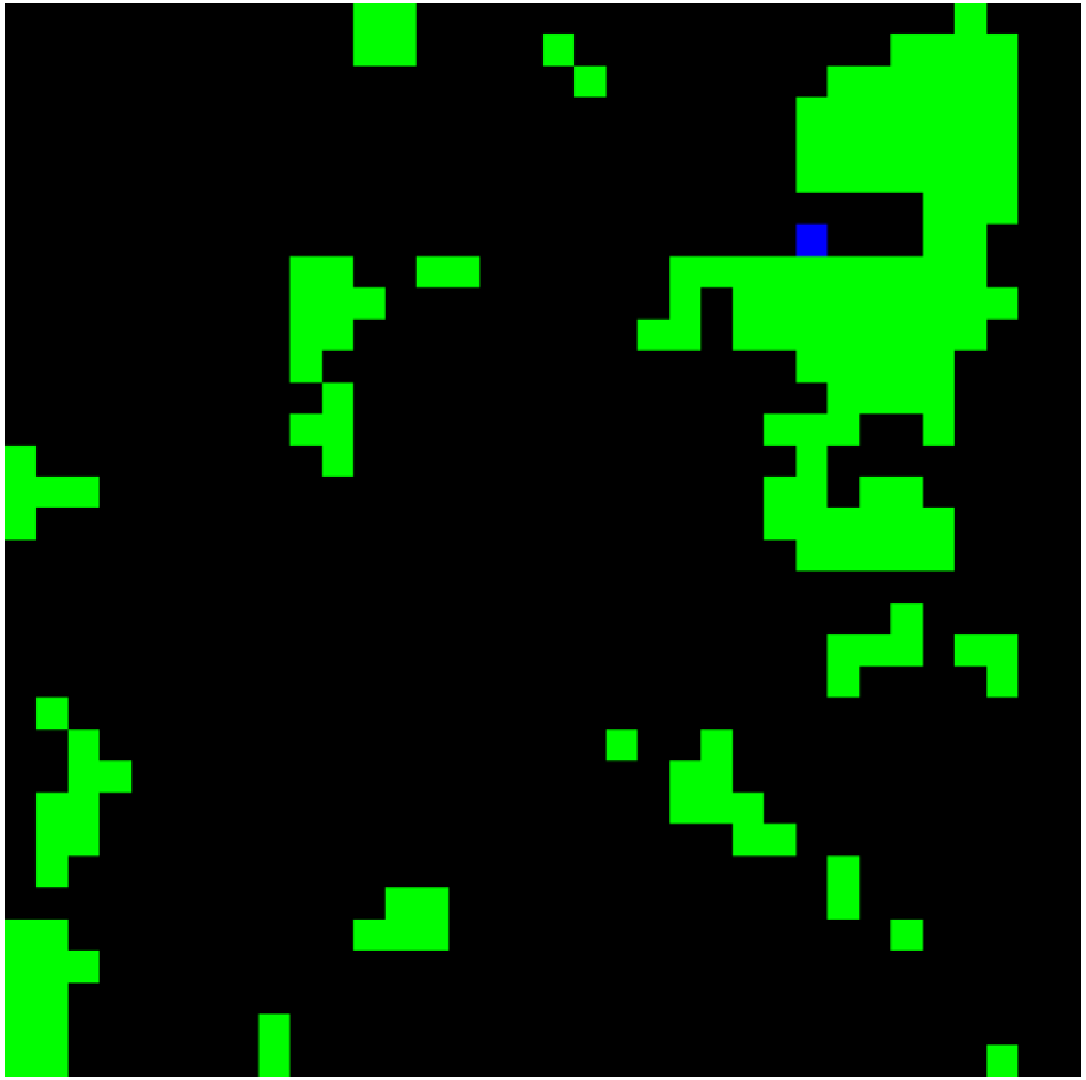




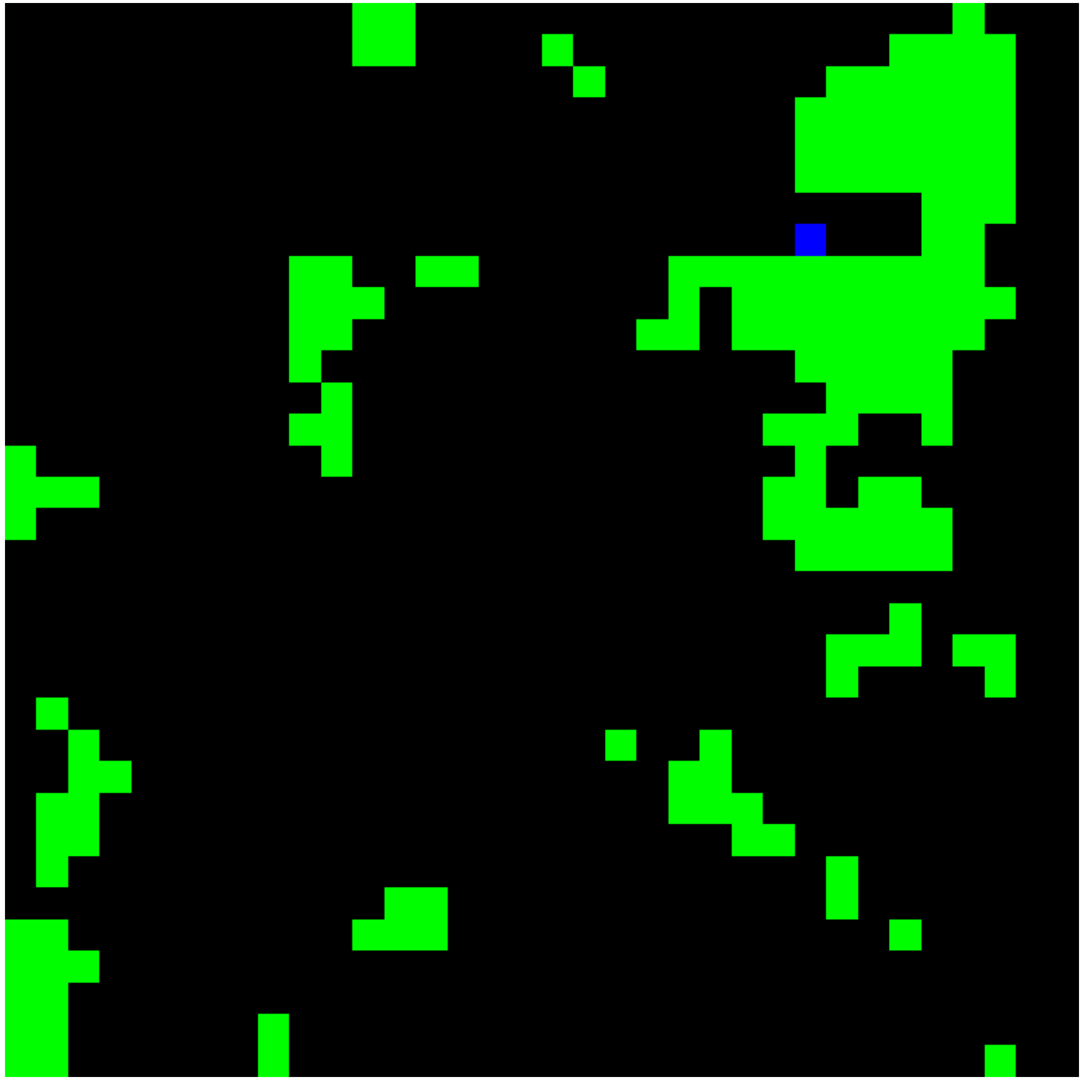
The init() method. will create a gridded landscape
•self.action_space – the agent may move left, right, up, down, add a habitat pixel
•self.observation_space – the agent sees the whole landscape matrix

The reset() method. each episode ends when the agent adds 20 habitat pixels.

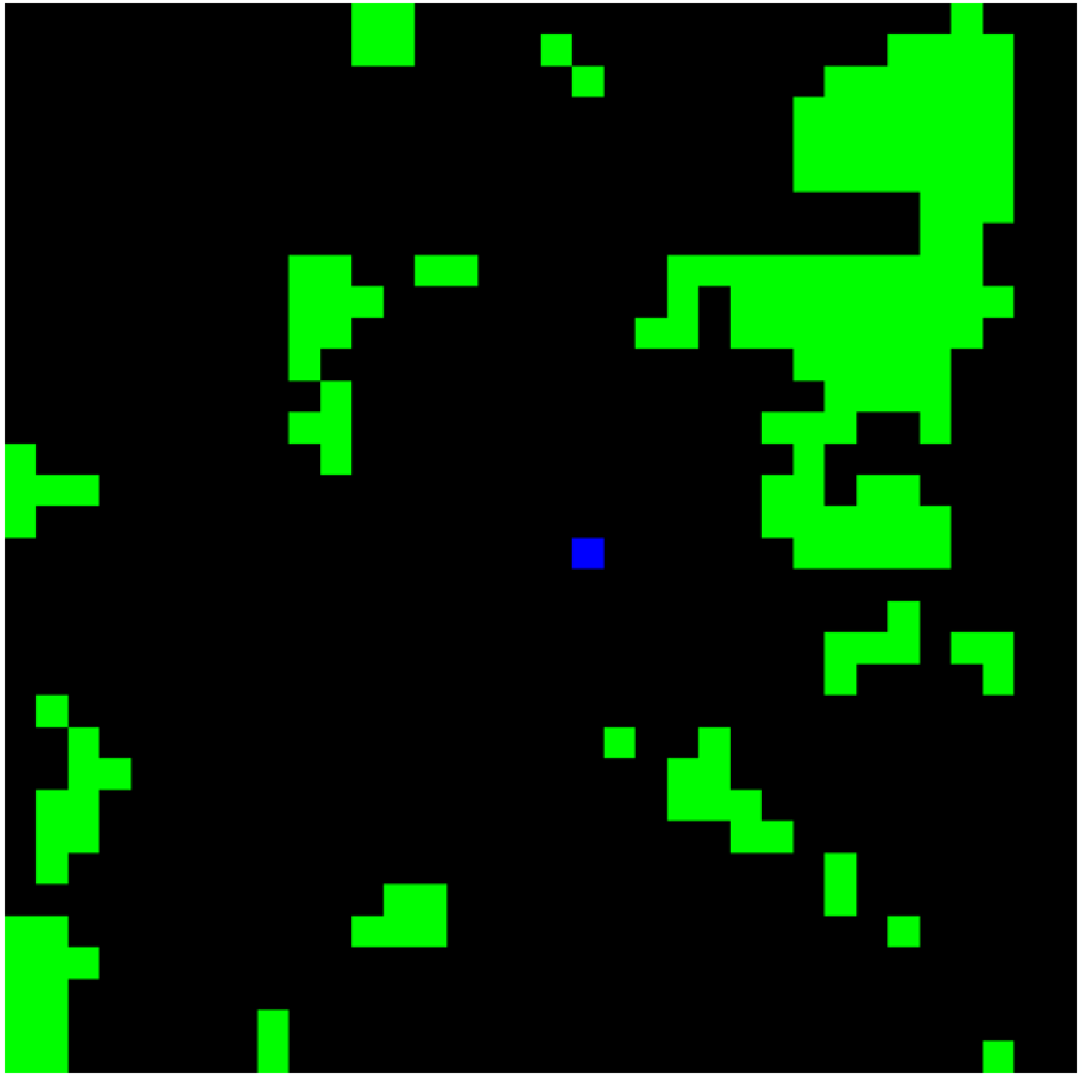
Improving connectivity through habitat restoration



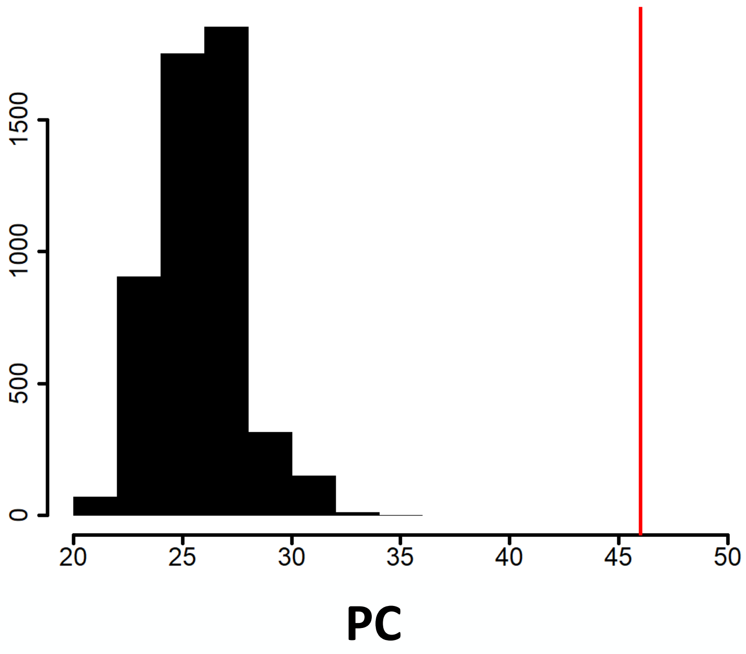
Improving connectivity through habitat restoration

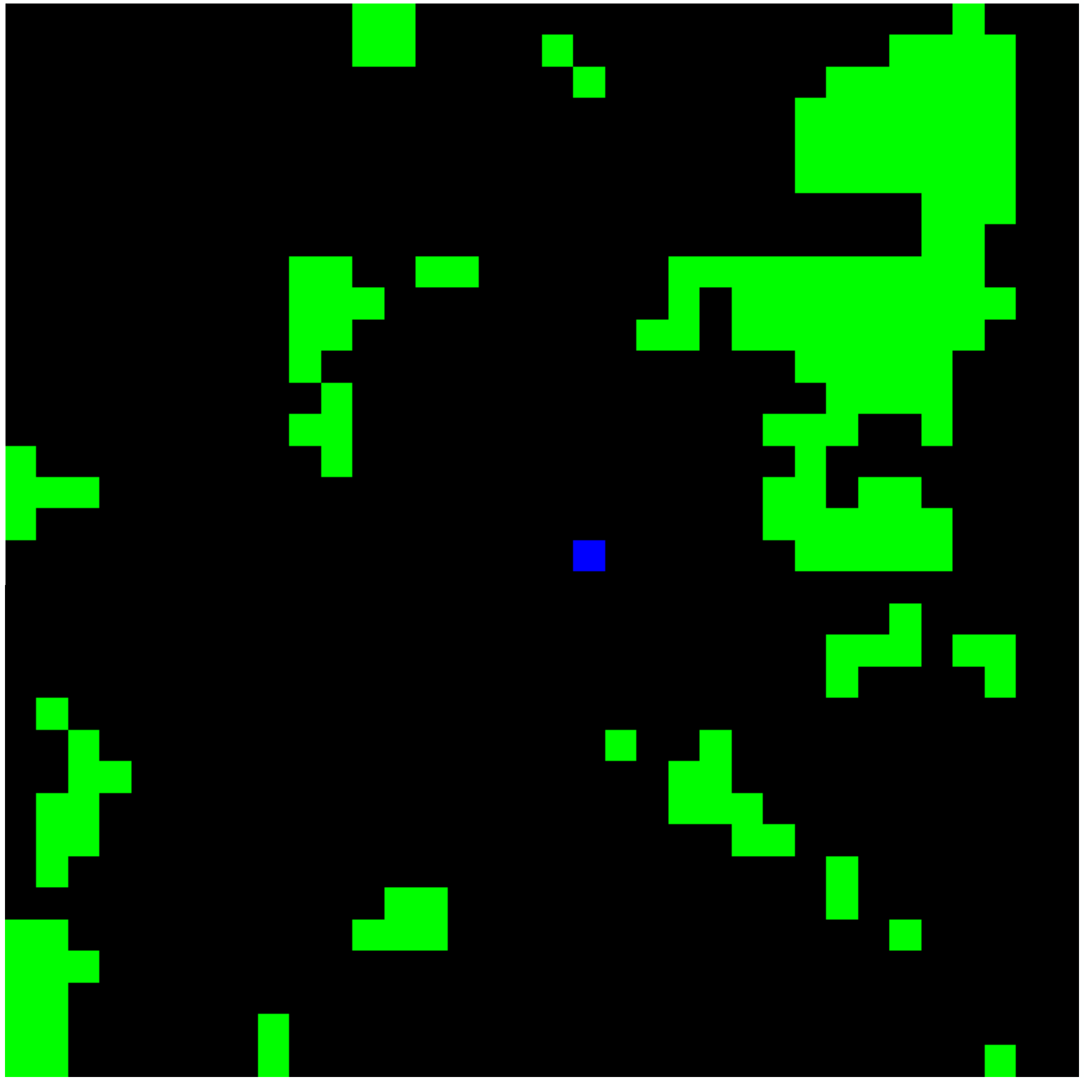


Improving connectivity through habitat restoration

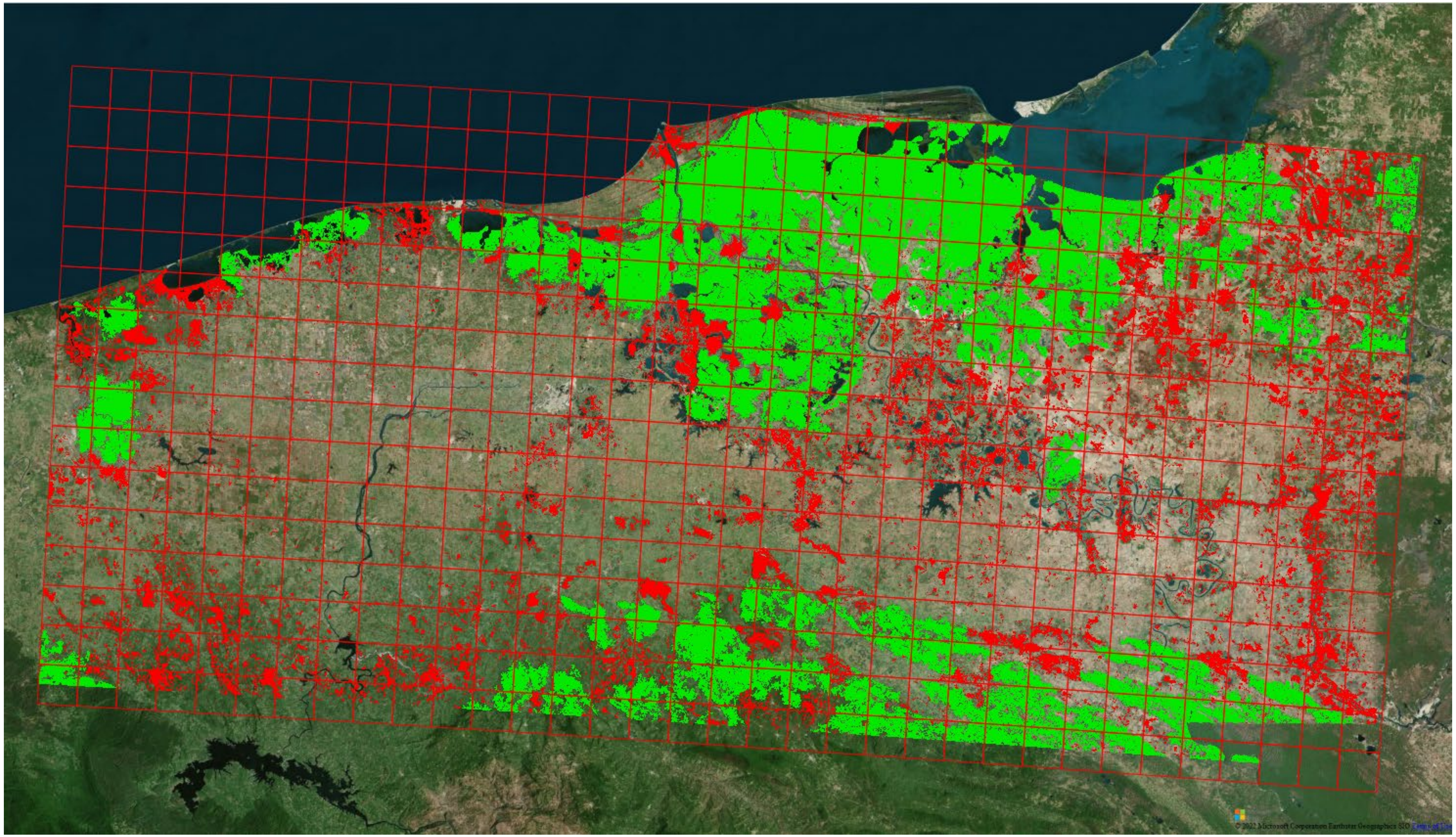


Improving connectivity through habitat restoration

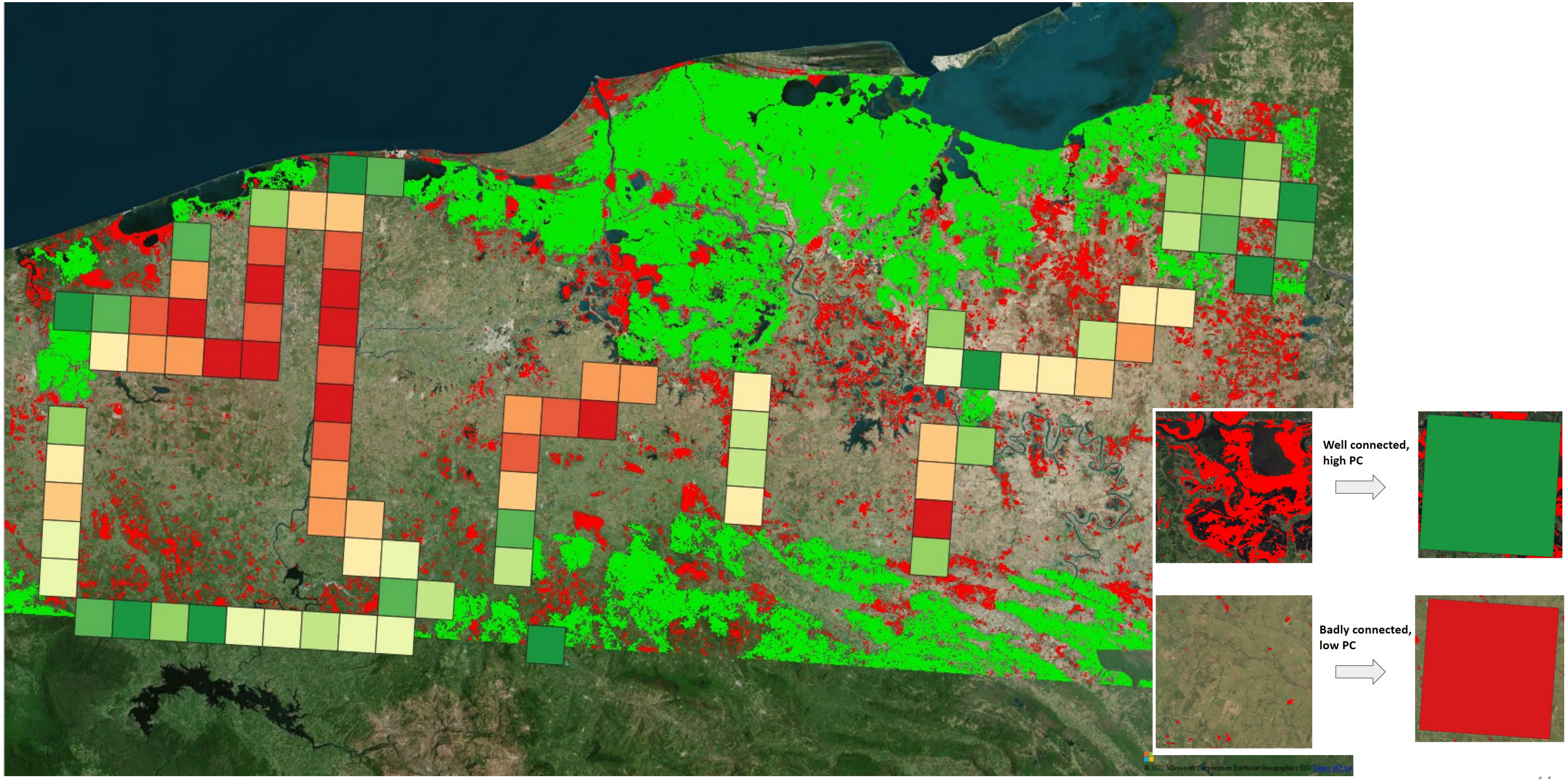




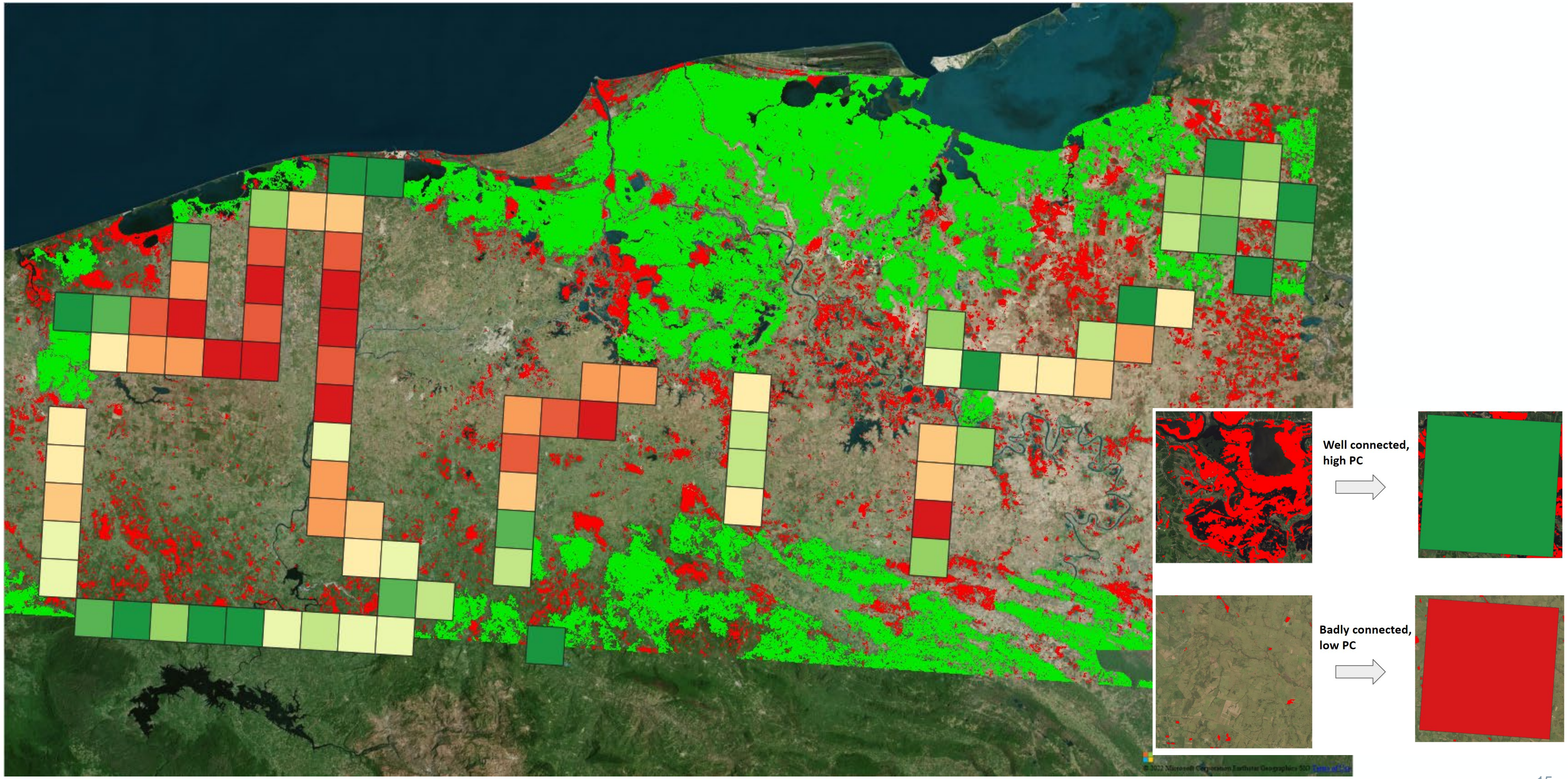
Improving connectivity through habitat restoration



Improving connectivity through habitat restoration



Improving connectivity through habitat restoration



Pros

- It offers a clear way to tackle complex decision making tasks
- You do not necessarily need to know the explicit equations governing your learning task
- If used creatively it can scale in ways that have not been seen before

Cons

- It's hard for the models to know exactly which actions in a sequence lead to a reward (sparse rewards, credit assignment problem)
- tends to be quite sample inefficient. For some complex problems initial random exploration fails completely
- tends to overfit reward functions so designing rewards is extremely difficult