#Last python code

import os

import rasterio

import csv

plants\_path = r'D:\sdm\new papers\idea\chile\_plant\_pollinator\Plants\future\binary\mask\crop'

pollinators\_path = r'D:\sdm\new papers\idea\chile\_plant\_pollinator\pollinators\future\binary\mask\crop'

interaction\_matrices = {}

plant\_files = [os.path.join(plants\_path, f) for f in os.listdir(plants\_path) if f.endswith('.tif')]

pollinator\_files = [os.path.join(pollinators\_path, f) for f in os.listdir(pollinators\_path) if f.endswith('.tif')]

plants = [rasterio.open(file) for file in plant\_files]

pollinators = [rasterio.open(file) for file in pollinator\_files]

plant\_layer\_names = [os.path.splitext(os.path.basename(file))[0] for file in plant\_files]

pollinator\_layer\_names = [os.path.splitext(os.path.basename(file))[0] for file in pollinator\_files]

output\_folder = r'D:\sdm\new papers\idea\chile\_plant\_pollinator\future\_csv\_files'

if not os.path.exists(output\_folder):

os.makedirs(output\_folder)

for idx, plant in enumerate(plants):

bounds = plant.bounds

xmin, ymin, xmax, ymax = bounds.left, bounds.bottom, bounds.right, bounds.top

x\_res, y\_res = plant.res

nrow, ncol = plant.shape

x\_increment = (xmax - xmin) / ncol

y\_increment = (ymax - ymin) / nrow

for i in range(ncol \* nrow):

row\_idx = i // ncol

col\_idx = i % ncol

cell\_center\_x = xmin + (col\_idx + 0.5) \* x\_increment

cell\_center\_y = ymax - (row\_idx + 0.5) \* y\_increment

cell\_name = f"Lat{cell\_center\_y}\_Lon{cell\_center\_x}"

print(f"Processing cell {cell\_name}")

valid\_cell = True # Initialize a flag to determine if the cell meets criteria

# Check if any layer has a value other than 1 or 0

for band in plants + pollinators:

cell\_value = band.read(1)[row\_idx, col\_idx]

if cell\_value not in [0, 1]:

valid\_cell = False

break # Exit the loop if an invalid value is found

if not valid\_cell:

print("Skipping cell due to non-binary values")

continue

interaction\_matrix = [[0 for \_ in range(len(pollinators))] for \_ in range(len(plants))]

for j, plnt in enumerate(plants):

for k, pol in enumerate(pollinators):

plnt\_val = plnt.read(1)[row\_idx, col\_idx]

pol\_val = pol.read(1)[row\_idx, col\_idx]

# Capture interaction when both plant and pollinator have a value of 1

if plnt\_val == 1 and pol\_val == 1:

interaction\_matrix[j][k] = 1

# Writing CSV immediately after processing the cell

csv\_filename = os.path.join(output\_folder, f"{cell\_name}.csv")

with open(csv\_filename, mode='w', newline='') as file:

writer = csv.writer(file)

writer.writerow([''] + pollinator\_layer\_names)

for i, row\_name in enumerate(plant\_layer\_names):

writer.writerow([row\_name] + [str(val) for val in interaction\_matrix[i]])

print(f"Interaction matrix created and saved as CSV for cell: {cell\_name}")