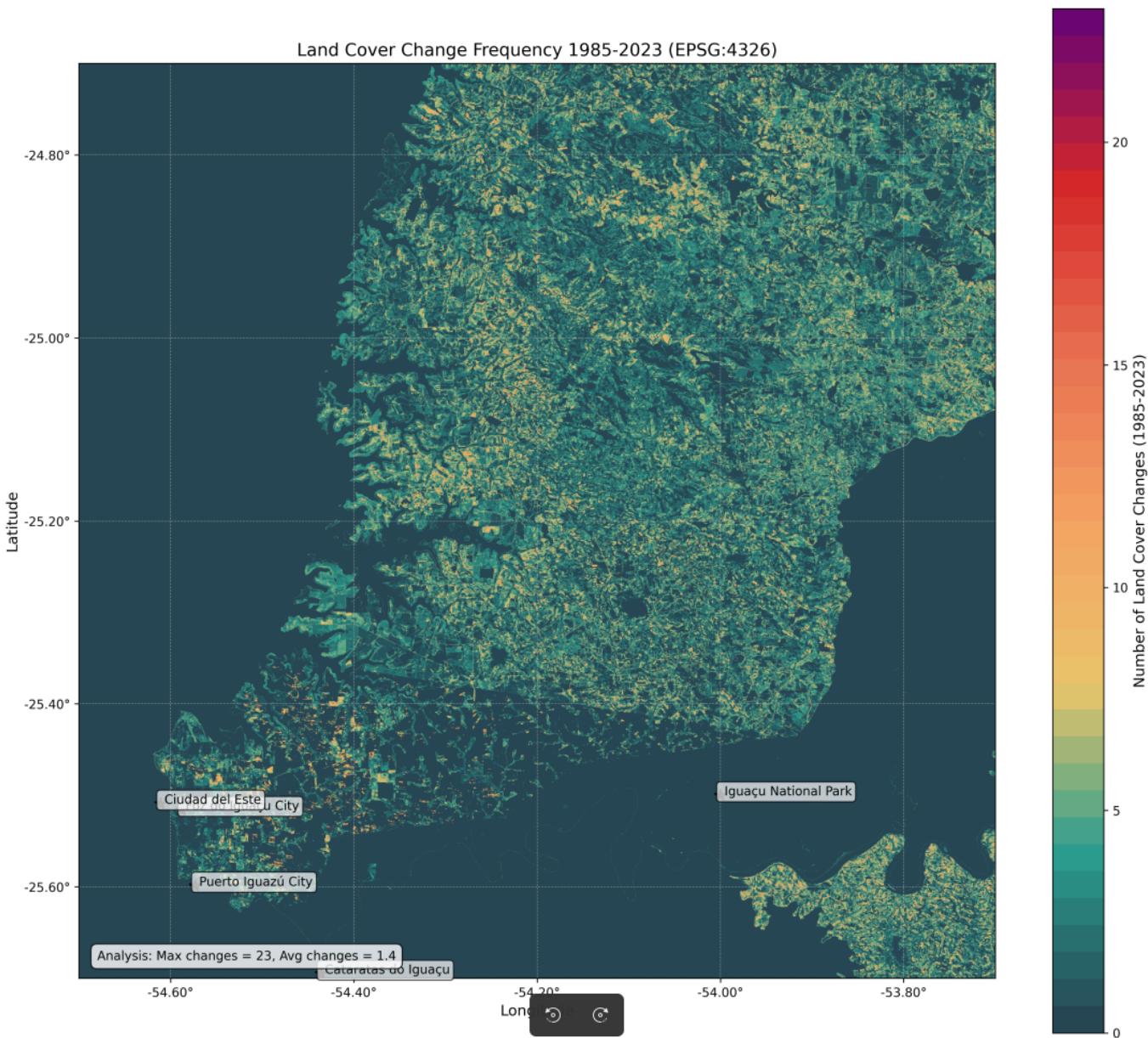


Iguaçu National Park Land Cover Analysis



Overview

This Python script analyzes land cover changes in the Iguaçu National Park region of Brazil from 1985 to 2023 using MapBiomas Collection data. The script extracts, processes, and visualizes land cover data to reveal patterns of landscape change over nearly four decades, with a focus on transitions between different land cover classes.

Features

- **Geographic Extraction:** Extracts data specifically for the Iguaçu region (approximately 25.7°S-24.7°S, 54.7°W-53.7°W)
- **Multi-temporal Analysis:** Processes land cover data across all available years (1985-2023)
- **Change Detection:** Identifies and quantifies locations of land cover change
- **Transition Analysis:** Tracks conversions between different land cover types

- **Interactive Visualizations:**
 - Land cover maps with geographic coordinates
 - Change frequency heat maps
 - Sankey diagrams showing land cover transitions by decade
- **Landmarks Identification:** Labels key geographic features (Iguaçu Falls, cities, park boundaries)

Prerequisites

- Python 3.7+
- Required packages:

```
rasterio
numpy
zarr
matplotlib
plotly
tqdm
```

- MapBiomass Brazil Collection data as a VRT file

Data Requirements

The script requires a Virtual Raster (VRT) file that references yearly land cover classification GeoTIFFs from the MapBiomass project:

- **Source:** MapBiomass Collection (recommended Collection 8 or later)
- **Resolution:** 30m
- **Projection:** EPSG:4326 (WGS84)
- **Time Range:** 1985-2023
- **Classification:** MapBiomass land cover classes

Implementation Details

Data Extraction

The script extracts data for the Iguaçu region using these steps:

1. Defines target coordinates for the Iguaçu region
2. Converts geographic coordinates (lat/lon) to pixel coordinates
3. Creates a raster window for the specified region
4. Extracts raster data for all available years
5. Calculates pixel-based changes between consecutive years
6. Stores results in a Zarr dataset for efficient access

Land Cover Classification

The analysis uses the MapBiomass classification system, which includes:

- Forest formations (classes 1, 3)

- Savanna formation (class 4)
- Wetlands (class 11)
- Grasslands (class 12)
- Pasture (class 15)
- Agriculture (classes 20, 39, 40, 41, etc.)
- Urban areas (class 24)
- Water bodies (class 33)
- And many more classes

Change Analysis

For each pixel, the script:

1. Counts how many times the land cover class changed from 1985 to 2023
2. Identifies the specific transitions between classes
3. Creates visualization of change frequency
4. Analyzes statistical patterns of change

Sankey Diagram Generation

The script creates Sankey diagrams for four time periods:

- 1985-1995
- 1995-2005
- 2005-2015
- 2015-2023

Each diagram shows:

- Persistence (areas that maintained the same class)
- Transitions (areas that changed from one class to another)
- Proportional flow representation based on pixel counts
- Color-coded flows matching the standard MapBiomas palette

Usage

1. Set the path to your MapBiomas VRT file:

```
VRT_FILE = '/path/to/mapbiomas_coverage_1985_2023.vrt'
```

2. Set the output directory:

```
OUTPUT_DIR = 'iguacu_results'
```

3. Run the script:

```
python iguacu_sankey.py
```

Outputs

The script generates the following outputs in the specified directory:

Data Files

- `data.zarr/` - Zarr dataset containing:
 - Original land cover data for all years
 - Change frequency map
 - Transition matrix
 - Geographic metadata

Visualizations

- `extraction_preview.png` - Quick preview of the extracted area
- `iguacu_2023.png` - Land cover map for 2023 with geographic coordinates
- `iguacu_changes.png` - Heat map showing frequency of land cover changes

Transition Diagrams

- `transitions_1985_1995.html` - Interactive Sankey diagram (HTML)
- `transitions_1985_1995.png` - Static image of Sankey diagram
- `transitions_1995_2005.html/.png` - Diagrams for 1995-2005
- `transitions_2005_2015.html/.png` - Diagrams for 2005-2015
- `transitions_2015_2023.html/.png` - Diagrams for 2015-2023

Geographic Parameters

- **Target Region:** Iguaçu National Park and surroundings
- **Bounding Box:**
 - Latitude: -25.7°S to -24.7°S (South to North)
 - Longitude: -54.7°W to -53.7°W (West to East)
- **Region Size:** Approximately 100 × 100 km

Key Landmarks

The script includes the following landmarks (when within the extracted region):

- **Iguaçu National Park:** Primary conservation area (-25.5°S, -54.0°W)
- **Foz do Iguaçu City:** Brazilian city (-25.516°S, -54.588°W)
- **Cataratas do Iguaçu:** The famous waterfalls (-25.695°S, -54.436°W)
- **Puerto Iguazú City:** Argentine city (-25.599°S, -54.573°W)
- **Ciudad del Este:** Paraguayan city (-25.509°S, -54.611°W)

Notes

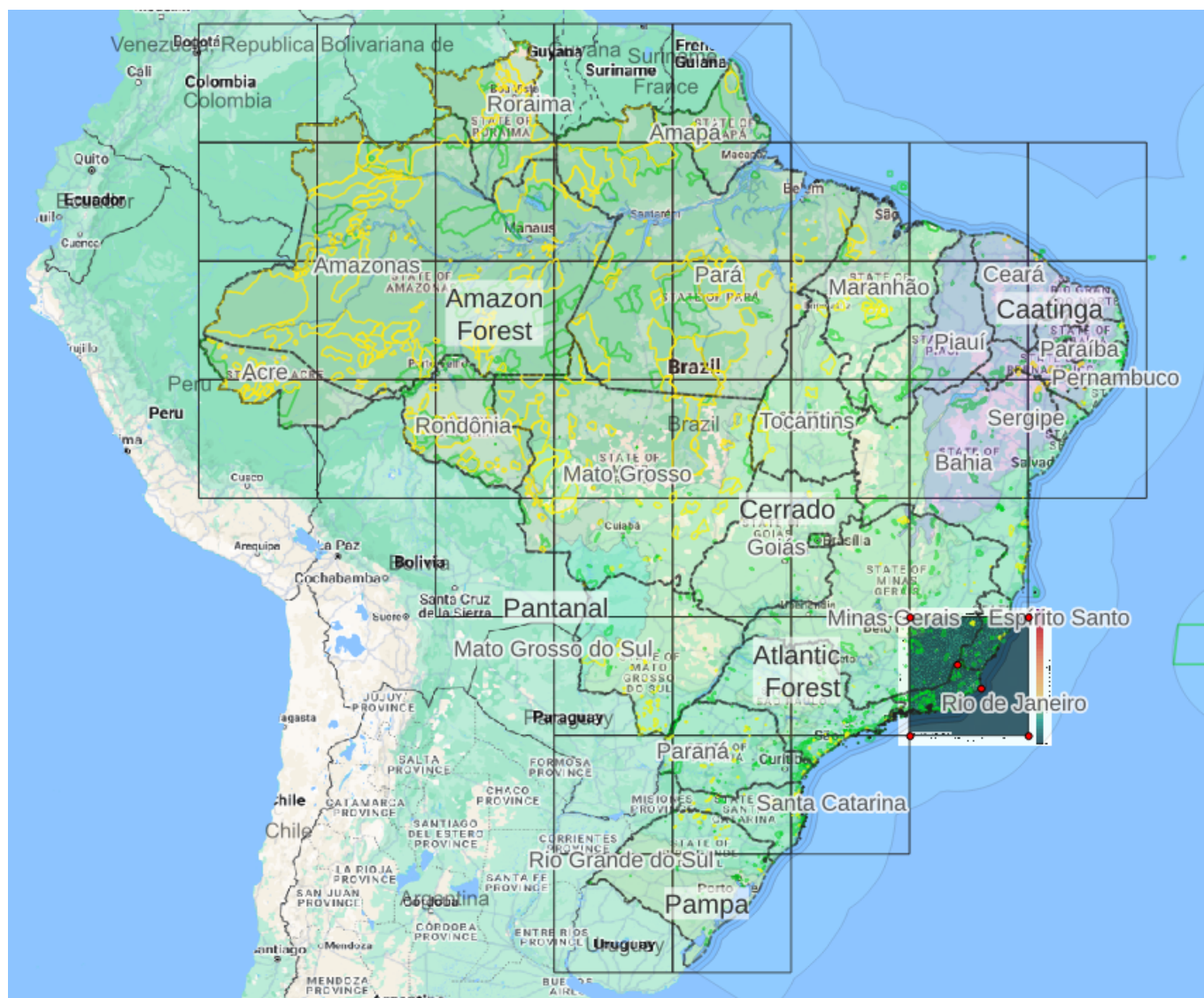
- Processing the full dataset requires approximately 2-3GB of RAM
- The script automatically filters out small, insignificant transitions (less than 0.1% of total)
- Visualization includes annotations with statistics about change patterns
- All visualizations include proper geographic coordinates in EPSG:4326

Created by Leandro Meneguelli Biondo for the UBCO PhD project, 2025.

MapBiomass Brazil Grid Analysis

Overview

This Python script analyzes land cover change patterns across Brazil using MapBiomass data from 1985-2023. The analysis divides Brazil into 5-degree grid cells and processes each cell to identify land cover changes, transitions between classes, and persistence patterns.



Features

- **Systematic Grid Analysis:** Divides Brazil into 43 grid cells of 5×5 degrees for manageable processing
- **Change Detection:** Quantifies frequency of land cover changes for each pixel from 1985-2023

- **Transition Analysis:** Tracks conversions between different land cover classes (e.g., Forest → Pasture)
- **Optimized Processing:** Uses chunked data processing via Zarr for handling large raster datasets
- **Rich Visualizations:**
 - Land cover maps with geographic coordinates
 - Change frequency heat maps
 - Interactive Sankey diagrams showing land cover flows

Prerequisites

- Python 3.7+
- Required packages:

```
rasterio
numpy
zarr
matplotlib
plotly
tqdm
shapely
```

- MapBiomass Brazil Collection 8 data as a VRT file

Data Source

This script processes the MapBiomass Collection 8 dataset, which provides annual land cover classifications for Brazil from 1985 to 2023 at 30m resolution. The data contains 39 land cover classes including forest formations, savannas, grasslands, pasture, agriculture, and urban areas.

- **VRT File:** The script expects a Virtual Raster (VRT) file that references all yearly land cover GeoTIFFs
- **Coordinate System:** EPSG:4326 (WGS84)
- **Data Structure:** Each band in the VRT represents one year (1985-2023)

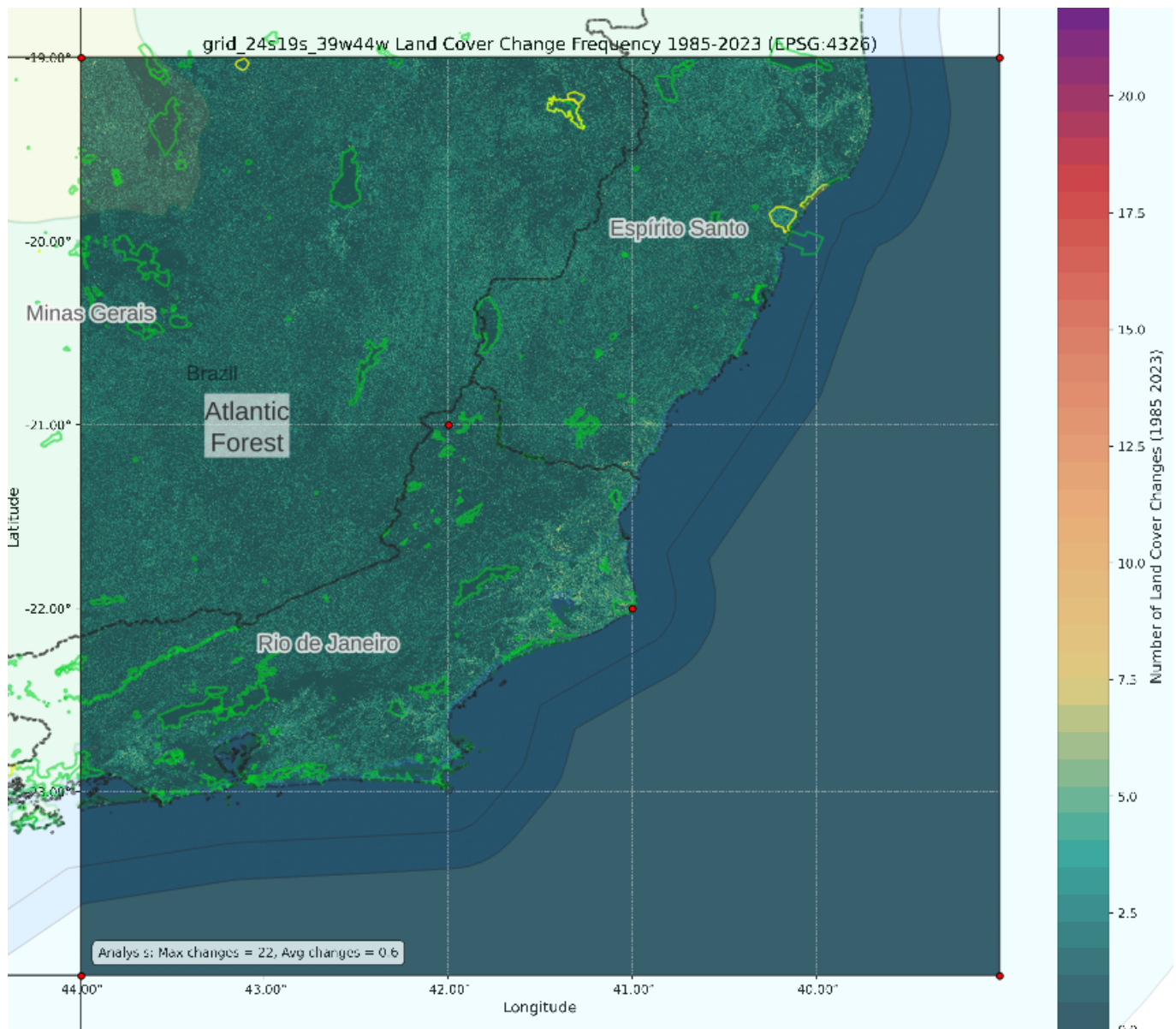
Grid System

The script divides Brazil into 43 cells using a 5-degree grid system:

```
((-74, 6), (-69, 6), (-69, 1), (-74, 1), (-74, 6)), # Northwest Amazon
((-69, 6), (-64, 6), (-64, 1), (-69, 1), (-69, 6)), # North Amazon
...
((-54, -29), (-49, -29), (-49, -34), (-54, -34), (-54, -29)) # South
Brazil
```

Each cell is processed independently, allowing for parallel processing and manageable memory usage.

Analysis Process



For each grid cell, the script:

1. **Extracts Data:** Clips the VRT file to the grid cell coordinates
2. **Calculates Changes:** Compares each consecutive year to identify pixels that changed
3. **Builds Transition Matrix:** Tracks all transitions between land cover classes
4. **Stores Results:** Creates a Zarr dataset with original data, changes, and transitions
5. **Generates Visualizations:** Creates maps and diagrams to visualize the results

Output Files

For each grid cell, the script generates:

- **Zarr Dataset** (`data.zarr`): Contains:
 - Original yearly land cover data
 - Change frequency map
 - Transition matrix
 - Metadata (coordinate system, bounds, etc.)
- **Visualizations:**

- `extraction_preview.png`: Quick preview of the grid cell
- `{grid_name}_2023.png`: Map of 2023 land cover
- `{grid_name}_changes.png`: Map of change frequency
- `transitions_{start}_{end}.html`: Interactive Sankey diagrams for each decade
- `transitions_{start}_{end}.png`: Static image of Sankey diagrams

Key Functions

`extract_grid_data(vrt_path, polygon_coords, output_dir)`

Extracts and processes data for a single grid cell. Converts geographic coordinates to pixel coordinates, reads data from the VRT file, and calculates changes and transitions.

`create_decadal_sankey_diagrams(root, output_dir)`

Creates Sankey diagrams showing land cover transitions for four periods: 1985-1995, 1995-2005, 2005-2015, and 2015-2023. Filters out insignificant transitions (less than 0.1% of total).

`visualize_results(zarr_path, output_dir)`

Generates visualizations from the processed data, including land cover maps and change frequency maps with proper geographic coordinates.

`get_grid_name(polygon_coords)`

Generates a standardized name for each grid cell based on its coordinates (e.g., `grid_6n1n_74w69w`).

Usage

1. Set the path to your MapBiomass VRT file in the script:

```
VRT_FILE = '/path/to/mapbiomas_coverage_1985_2023.vrt'
```

2. Set the output directory:

```
OUTPUT_BASE_DIR = 'grid_results'
```

3. Run the script:

```
python brazil_grid5.py
```

4. Results will be saved in the output directory, with a subdirectory for each grid cell.

Notes and Limitations

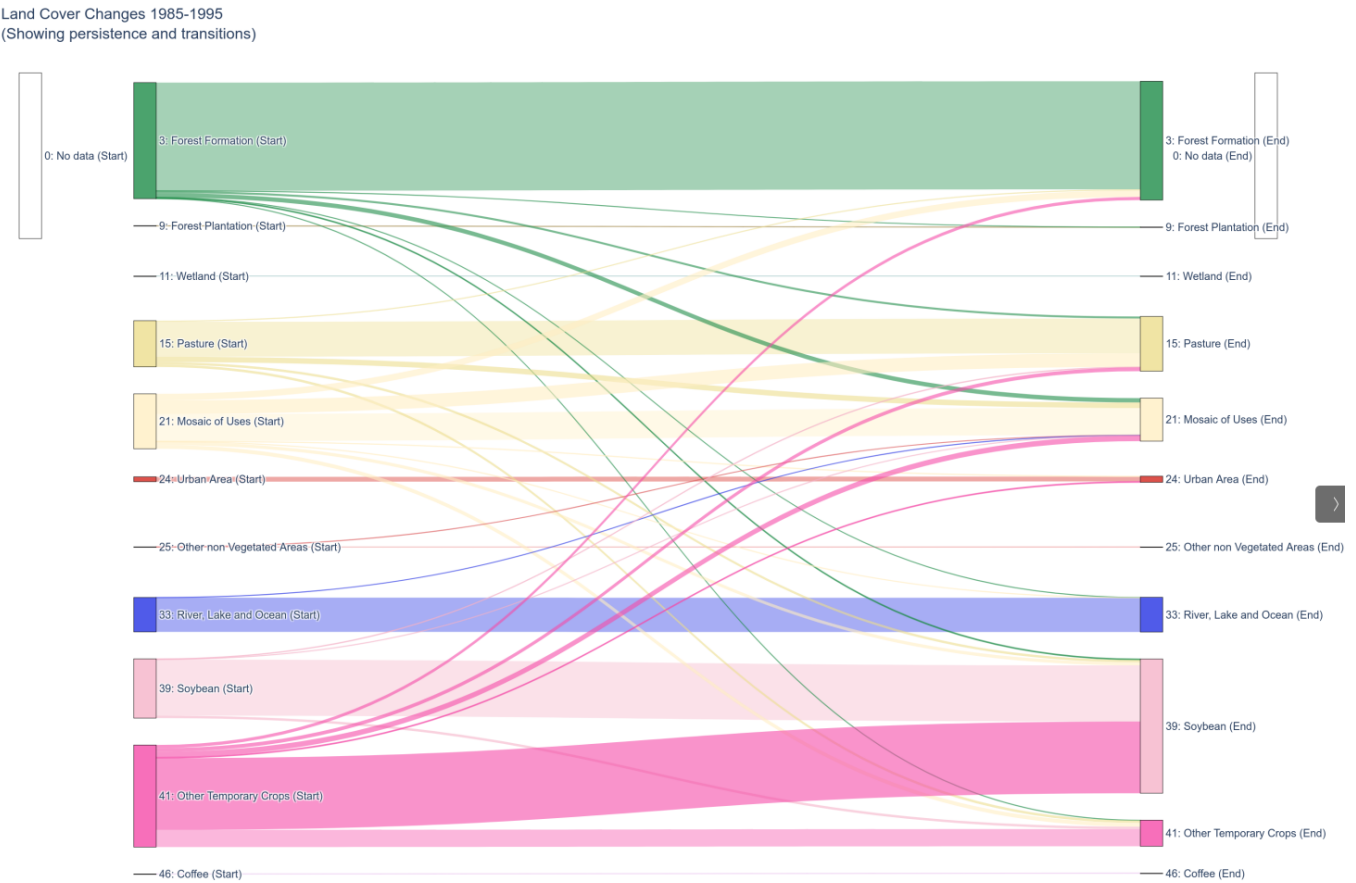
- **Memory Usage:** Processing each 5-degree cell requires approximately 2-4GB of RAM
- **Disk Space:** Each grid cell generates 500MB-1GB of data
- **Processing Time:** Each grid cell takes 10-30 minutes to process on a modern system
- **Edge Effects:** The grid system may introduce edge effects at cell boundaries
- **Data Quality:** The analysis inherits any classification errors from the MapBiomas dataset

Created by Leandro Meneguelli Biondo for the UBCO PhD project, 2025.

Brazil Land Cover Analysis Combiner

Overview

This Python script combines and analyzes land cover data across Brazil from the MapBiomas dataset (1985-2023). It merges multiple Zarr datasets from grid-based processing into a comprehensive national dataset while implementing memory-efficient techniques for handling very large geospatial data.



Features

- **Memory-Efficient Data Combination:** Merges multiple Zarr grid datasets into a single comprehensive dataset
- **Chunked Processing:** Uses Zarr's chunked array capabilities to handle datasets larger than available RAM
- **Downsampling:** Implements visualization downsampling for displaying continent-scale data

- **Statistical Sampling:** Uses representative sampling for creating Sankey diagrams of nationwide land cover transitions
- **Comprehensive Metadata:** Preserves geographic information and transformations from source datasets

Prerequisites

- Python 3.7+
- Required packages:

```
zarr
numpy
matplotlib
plotly
tqdm
rasterio
```

- Pre-processed MapBiomass grid datasets in Zarr format

Data Sources

This script expects pre-processed MapBiomass data in Zarr format, typically generated by the `brazil_grid5.py` script that divides Brazil into 5-degree grid cells. Each grid dataset contains:

- Original land cover data (yearly, 1985-2023)
- Change frequency maps
- Transition matrices
- Geographic metadata (bounds, transforms, etc.)

Core Functions

```
combine_zarr_datasets(input_base_dir, output_dir, max_chunk_size=512)
```

Combines multiple grid-based Zarr datasets into a single comprehensive dataset:

1. **Collects Metadata:** Scans all grid datasets to determine combined dimensions and bounds
2. **Creates Output Structure:** Initializes a new Zarr store with appropriate chunking
3. **Combines Data:** Merges data from each grid, preserving geographic relationships
4. **Memory-Efficient Processing:** Uses chunked reads/writes to manage memory usage
5. **Preserves Metadata:** Combines and stores critical geographic information

```
create_full_sankey_diagrams(zarr_path, output_dir,
sample_fraction=0.01)
```

Creates Sankey diagrams showing land cover transitions between decades:

1. **Random Sampling:** Selects a representative sample (default 1%) of pixels to analyze
2. **Transition Analysis:** Identifies persistent areas and transitions between land cover types
3. **Statistical Scaling:** Scales sample counts to represent the full dataset

4. **Filtering:** Removes statistically insignificant transitions (<0.1% of total)
5. **Interactive Visualization:** Creates interactive Sankey diagrams using Plotly

```
visualize_full_results(zarr_path, output_dir, downsample_factor=10)
```

Generates visualizations for the combined dataset:

1. **Downsampling:** Reduces resolution by a specified factor (default 10x) for visualization
2. **Geographic Rendering:** Maintains proper geographic coordinates and bounds
3. **Legend Creation:** Includes comprehensive legend with all land cover classes
4. **High-Quality Export:** Produces publication-quality images with appropriate metadata

Usage

1. Set input and output directories in the script:

```
INPUT_BASE_DIR = '/path/to/mapbiomas_proc_zarr' # Directory
containing grid Zarr datasets
OUTPUT_DIR = '/path/to/combined_results'        # Output directory
for combined results
```

2. Run the script:

```
python brazil_analyse_combine.py
```

3. The script will:

- Combine all grid datasets into a single Zarr store
- Create downsampled visualizations of the entire dataset
- Generate Sankey diagrams for nationwide land cover transitions

Output Files

- **Combined Zarr Dataset:** `combined_data.zarr` containing:
 - Full land cover data for all years
 - Nationwide change frequency map
 - Comprehensive transition matrix
 - Combined metadata
- **Visualizations:**
 - `downsampled_10_full_dataset_2023.png`: Land cover map of Brazil for 2023 (10x downsampled)
 - `downsampled_10_full_dataset_changes.png`: Change frequency map (10x downsampled)
- **Sankey Diagrams:**

- `sampled_transitions_1985_1995.html/.png`: Interactive/static diagrams showing transitions
- `sampled_transitions_1995_2005.html/.png`: for each decadal period
- `sampled_transitions_2005_2015.html/.png`:
- `sampled_transitions_2015_2023.html/.png`:

Memory Considerations

This script employs several techniques to manage large datasets efficiently:

- **Chunked Storage**: Data is stored and processed in manageable chunks (default 512×512 pixels)
- **Statistical Sampling**: Sankey diagrams use a 1% sample of pixels (adjustable via `sample_fraction`)
- **Visualization Downsampling**: Images are generated at reduced resolution (adjustable via `downsample_factor`)

For a full dataset covering Brazil (approximately 8,500,000 km²), the memory requirements are:

- Base memory usage: ~500MB
- Peak usage during combination: ~2-4GB (adjustable via chunk size)
- Disk space for combined dataset: ~20-30GB

Notes and Limitations

- **Processing Time**: Full combination process may take 1-2 hours depending on input size
- **Statistical Accuracy**: Sampling introduces a small margin of error in Sankey diagrams (~1-2%)
- **Visualization Detail**: Downsampling reduces the detail visible in nationwide visualizations
- **Geographic Alignment**: Minor pixel alignment issues may occur when combining grid datasets
- **Memory Usage**: Adjust sampling and downsampling parameters for your available memory

Created by Leandro Meneguelli Biondo for the UBCO PhD project, 2025.