Brazil Amazon and Atlantic Forest Land Cover Analysis Documentation

Overview

The brazil_color_table_amazon.py script provides a robust framework for extracting, analyzing, and visualizing land cover change in the Amazon region of Brazil using MapBiomas classification data from 1985-2023. The script employs parallel processing, memory optimization, and sophisticated visualization techniques to handle large geospatial datasets efficiently. For the Atlantic forest we just need to change coordinates, as this script can be run for any region in Brazil.

![Amazon Land Cover Example] (image-5.png)

![Atlantic forest Land Cover Example] (image-4.png)

Features

- **Robust Data Extraction**: Implements retry logic and error handling for reliable extraction from large VRT datasets
- **Parallel Processing**: Uses Python's `ProcessPoolExecutor` for efficient multi-core processing
- **Memory Management**: Dynamically monitors and manages memory usage to prevent overflow
- **Chunked Processing**: Processes data in configurable tiles to handle arbitrarily large regions
- **Comprehensive Visualization**:
  - Land cover maps with proper geographic referencing
  - Change frequency heat maps
  - Interactive Sankey diagrams showing land cover transitions
  - Decadal transition analysis
  - Color-coded transition matrices
- **GIS Integration**: Creates PNGW world files for geospatial software compatibility

Prerequisites

- Python 3.7+
- Required packages:

```
rasterio
numpy
zarr
matplotlib
plotly
tqdm
psutil
kaleido
```

- MapBiomas Collection data as a VRT file

Configuration Parameters

Parameter	Default	Description
TILE_SIZE	256	Size of processing tiles in pixels
MAX_WORKERS	10	Maximum number of parallel workers
MEMORY_BUFFER_GB	5	Memory buffer to maintain in GB
MAX_READ_RETRIES	3	Maximum retries for failed reads
RETRY_DELAY	0.05	Delay between retries in seconds
VRT_BLOCK_SIZE	512	Block size matching VRT structure

Land Cover Classification

The script uses the MapBiomas classification system with 50+ land cover classes, including:

- Forest formations (classes 1, 3)
- Savanna formation (class 4)
- Mangrove (class 5)
- Floodable forest (class 6)
- Wetlands (class 11)
- Grassland (class 12)
- Pasture (class 15)
- Agriculture (classes 20, 39, 40, 41, etc.)
- Urban areas (class 24)
- Water bodies (class 33)

Each class has a predefined color and label for visualization purposes.

```
# Color mapping and labels
COLOR_MAP = {
    0: "#ffffff",
    1: "#1f8d49", 3: "#1f8d49", 4: "#7dc975", 5: "#04381d", 6: "#007785",
    9: "#7a5900", 10: "#d6bc74", 11: "#519799", 12: "#d6bc74",
13:"#ffffff", 14: "#ffefc3",
    15:"#edde8e", 18: "#e974ed", 19:"#c27ba0", 20: "#db7093",
    21: "#ffefc3", 22:"#d4271e", 23: "#ffa07a", 24: "#d4271e", 25:
"#db4d4f", 26:"#2532e4", 29: "#ffaa5f",
    30: "#9c0027", 31: "#091077", 32: "#fc8114", 33: "#259fe4", 35:
"#9065d0", 36:"#d082de",
    39: "#f5b3c8", 40: "#c71585", 41: "#f54ca9", 46: "#d68fe2", 47:
"#9932cc",
    48: "#e6ccff", 49: "#02d659", 50: "#ad5100", 62: "#ff69b4", 27:
"#ffffff"
}

LABELS = {
    0: "No data",
```

```
    1: "Forest", 3: "Forest Formation", 4: "Savanna Formation", 5:
"Mangrove", 6: "Floodable Forest",
    9: "Forest Plantation",  11: "Wetland", 10: "Herbaceous", 12:
"Grassland",13:"other", 14:"Farming",
    15: "Pasture", 18:"Agri", 19:"Temporary Crop", 20: "Sugar Cane",
    21: "Mosaic of Uses", 22:"Non vegetated", 23: "Beach and Sand", 24:
"Urban Area",
    25: "Other non Vegetated Areas", 26:"Water", 29: "Rocky Outcrop", 30:
"Mining", 31: "Aquaculture",
    32: "Hypersaline Tidal Flat", 33: "River Lake and Ocean", 35: "Palm
Oil", 36:"Perennial Crop", 39: "Soybean",
    40: "Rice", 41: "Other Temporary Crops", 46: "Coffee", 47: "Citrus",
48: "Other Perennial Crops",
    49: "Wooded Sandbank Vegetation", 50: "Herbaceous Sandbank Vegetation",
62: "Cotton", 27: "Not Observed"
}
```

Core Functions

Data Processing

`robust_read(src, band, window, retries=MAX_READ_RETRIES)`

Reads raster data with retry logic for fault tolerance.

```
# Example usage
data = robust_read(src, 1, window)
```

`process_tile(task)`

Processes a single tile to calculate land cover changes and transitions.

```
# Task structure
task = (y, x, path, window, temp_dir)
changes, transitions, (y, x) = process_tile(task)
```

`extract_grid_data_optimized(vrt_path, polygon_coords, output_dir)`

Extracts and processes data for a specific geographic region.

```
# Example call for a specific region
zarr_path, grid_output_dir = extract_grid_data_optimized(
    '/path/to/mapbiomas.vrt',
    [((-55, -2), (-55, -7), (-51, -7), (-51, -2), (-55, -2))],
```

```
    'output_directory'
)
```

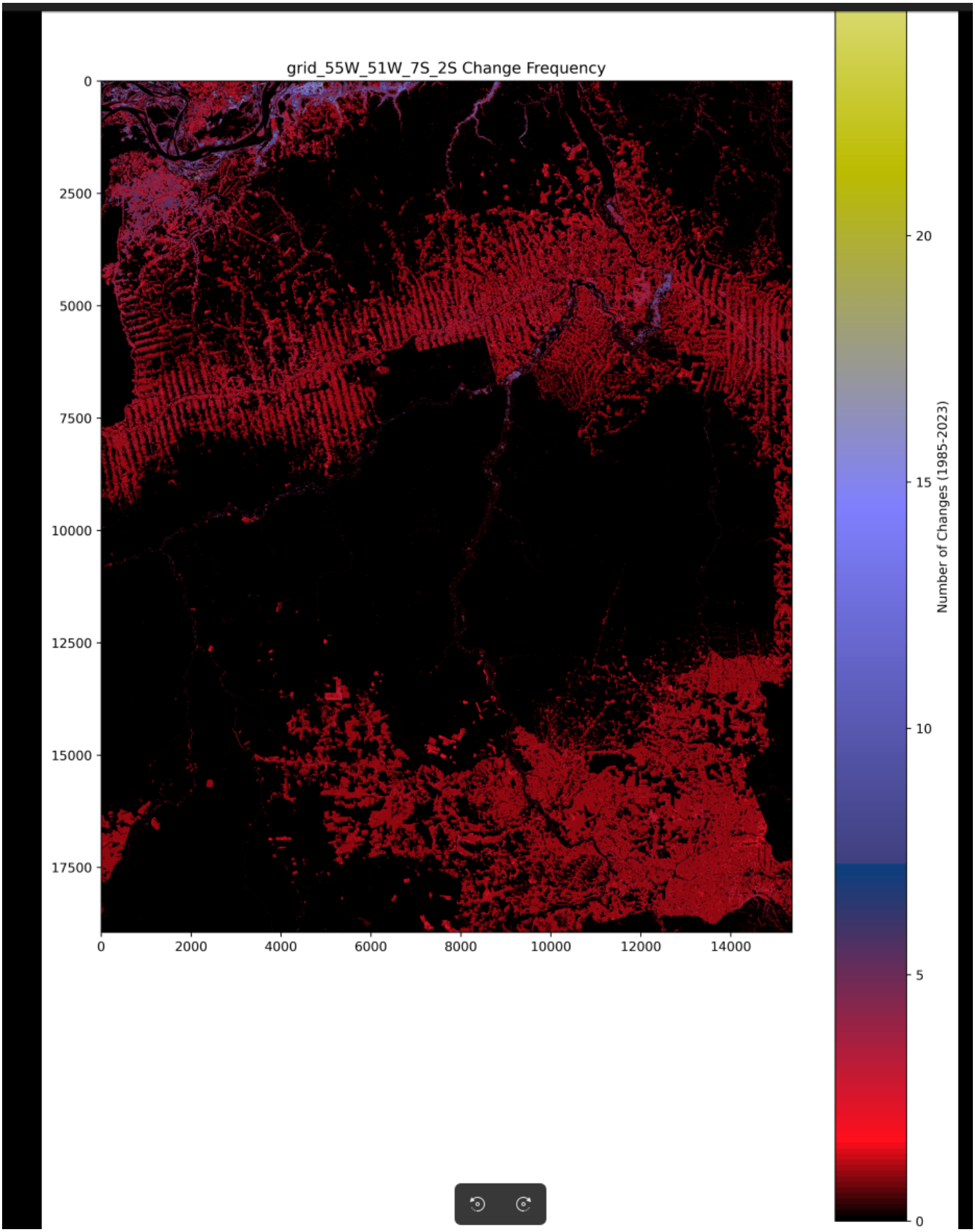
Visualization

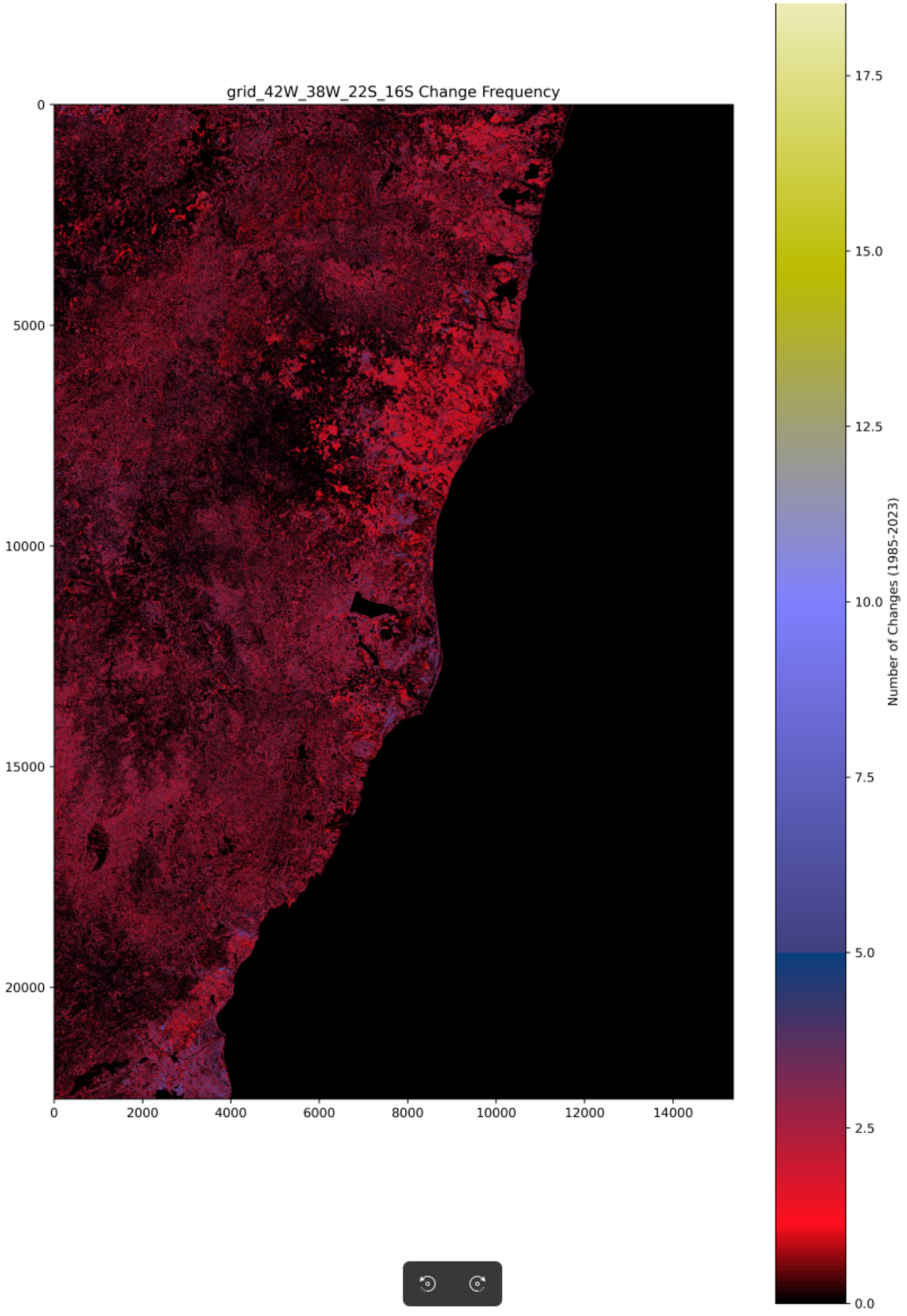
`create_landcover_map(root, output_dir)`

Creates a land cover map with proper georeferencing.

`create_changes_map(root, output_dir)`

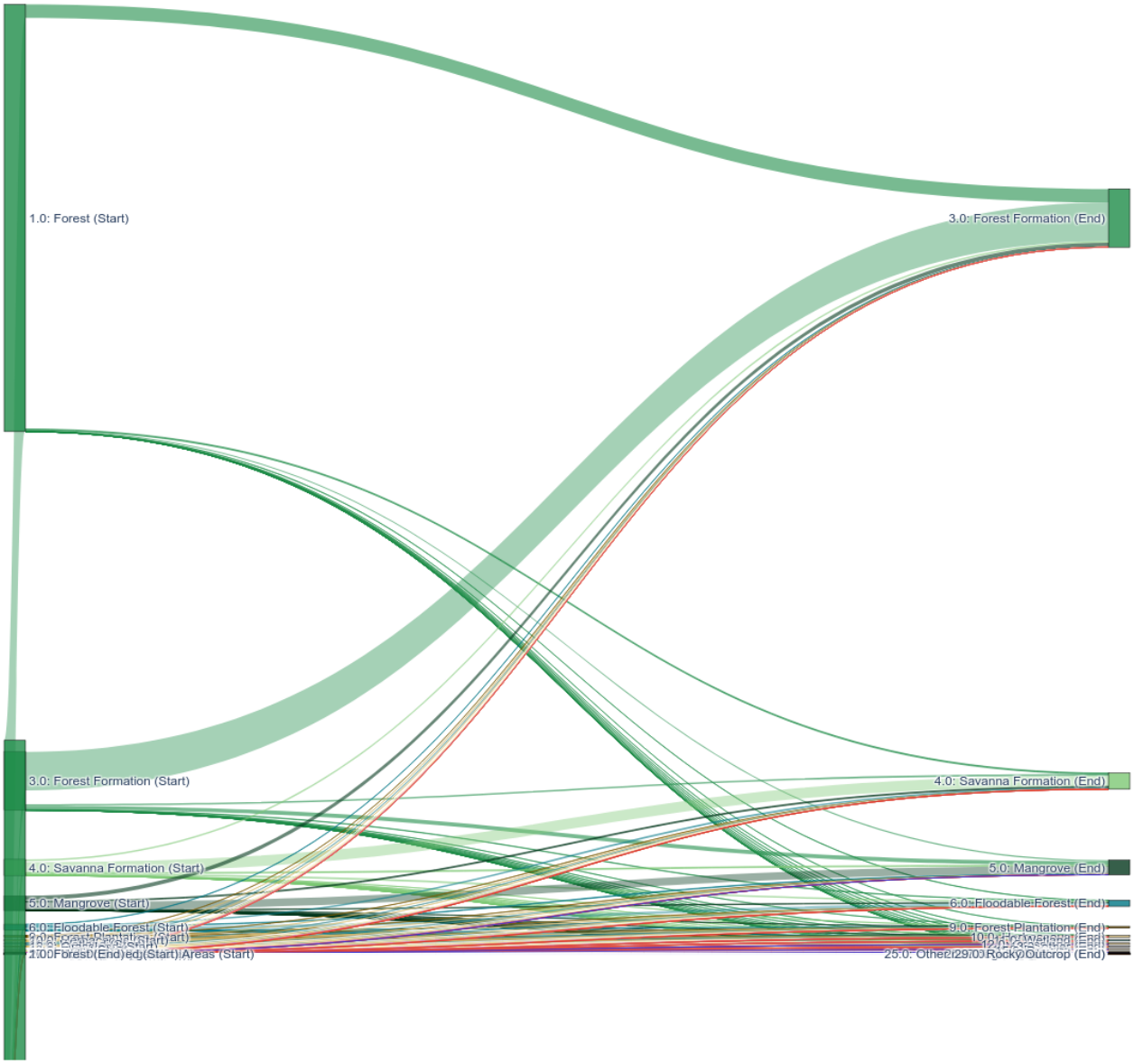
Creates a change frequency map showing where land cover has changed most frequently.



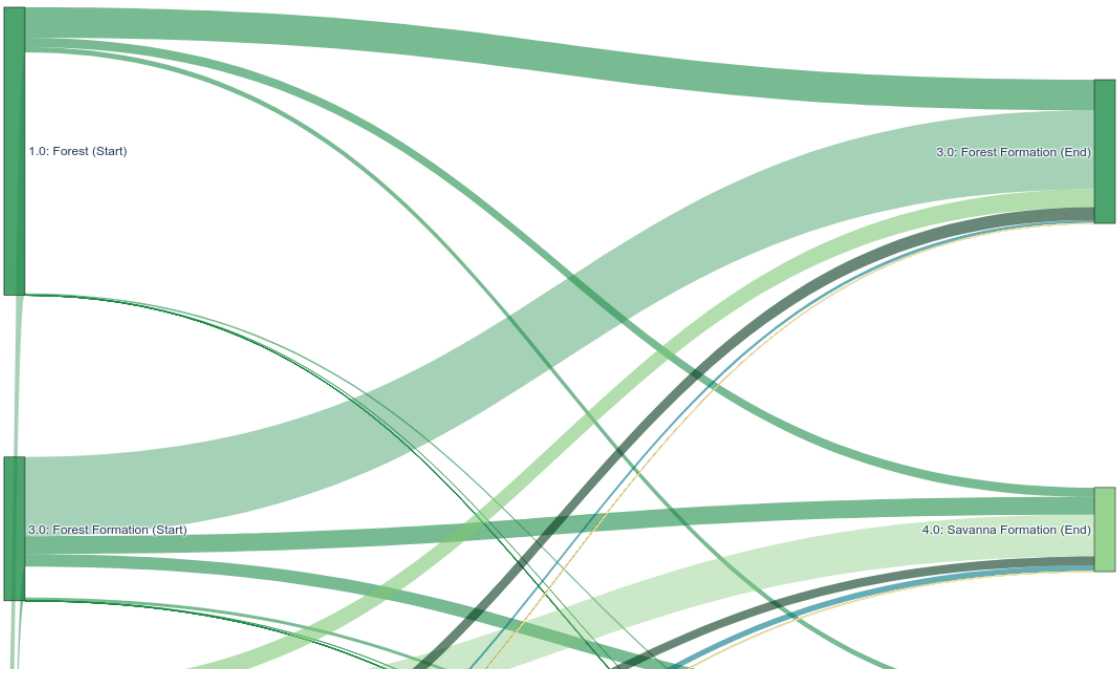

grid_42W_38W_22S_16S Change Frequency
0
5000
10000
15000
20000
0
2000
4000
6000
8000
10000
12000
14000
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Number of Changes (1985-2023)

```
create_sankey_diagrams(root, output_dir)
```

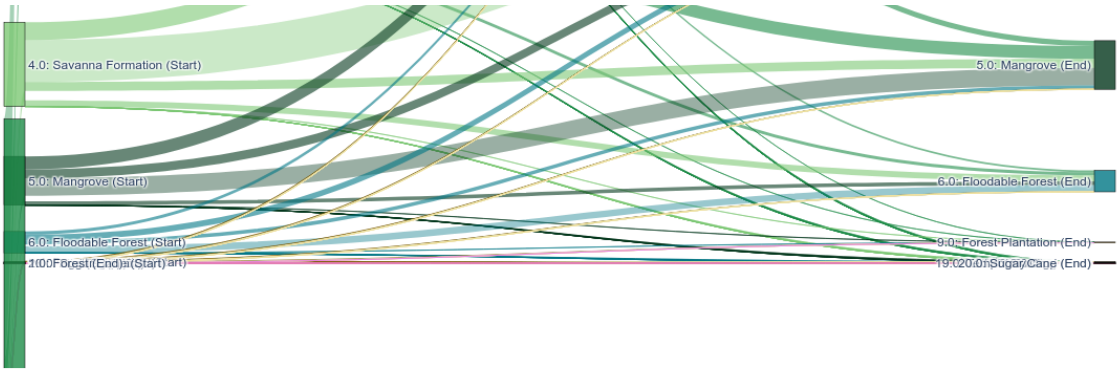
Creates Sankey diagrams showing land cover transitions across different time periods.



grid_42W_38W_22S_16S Land Cover Changes 2015-2023



/



`create_decadal_sankey_diagrams(root, output_dir)`

Creates Sankey diagrams specifically focused on decade-by-decade transitions.

Output Files

For each analyzed region, the script generates:

Data Files

- `data.zarr/` - Zarr dataset containing:
  - Original yearly land cover data
  - Change frequency map
  - Transition matrix
  - First year (1985) and last year (2023) snapshots
  - Geographic metadata

Visualizations

- `landcover_map.png` - Land cover map for 2023
- `landcover_map.pngw` - World file for GIS georeferencing
- `changes_map.png` - Heat map showing frequency of land cover changes
- `changes_map.pngw` - World file for GIS georeferencing

Transition Analysis

- `transitions_full.html` - Interactive Sankey diagram for entire period
- `transitions_full.csv` - CSV data of all transitions
- `transitions_1985_1995.html` - Interactive Sankey for 1985-1995
- `transitions_1995_2005.html` - Interactive Sankey for 1995-2005
- `transitions_2005_2015.html` - Interactive Sankey for 2005-2015
- `transitions_2015_2023.html` - Interactive Sankey for 2015-2023
- `change_matrix_{start}_{end}.png` - Graphical table of transitions
- `change_matrix_{start}_{end}.csv` - CSV data of transitions

Atlantic Forest change matrix

From/To	No data	Pasture	Forest Formation	Mosaic of Uses	Savanna Formation	Forest Plantation	Wetland	River Lake and Ocean	Coffee	Other Temporary Crops	Rocky Outcrop	Urban Area	Herbaceous Sandbank Vegetation	Wooded Sandbank Vegetation	Beach and Sand	Mangrove	Other non Vegetated Areas	Other Perennial Crops	Hypersaline Tidal Flat	Sugar Cane	Mining	Aquaculture	Grassland
No data	0	1609	4874	5050	65	0	27	110	11	4	135	10	0	2	4	0	7	1	0	0	0	0	0
Pasture	1377	0	720799	15089423	198064	955223	162981	126466	529307	424203	13663	131155	21174	3284	11905	60	17688	3347	26	0	1120	0	74
Forest Formation	5230	2811132	0	5097295	8611	1342558	8560	80158	68855	4881	3160	2783	6	244	30	8287	502	965	163	0	67	0	0
Mosaic of Uses	5401	18960527	3865583	0	288315	368681	241085	200353	119973	872412	16819	85884	25414	27539	29994	1326	23921	1729	678	0	1314	0	0
Savanna Formation	61	1413166	9807	330908	0	1408	0	7223	1056	0	326	581	0	0	0	0	471	16	0	0	729	0	3
Forest Plantation	1	206981	402589	68521	1495	0	87031	1470	600	490	0	743	816	30508	89	3318	195	5	54	0	5	0	0
Wetland	35	433502	6964	186989	0	77001	0	124835	20	33485	0	7885	1147	941	11759	4935	1905	0	3239	0	13	0	0
River Lake and Ocean	144	134705	74816	216587	5982	1291	117267	0	86	2137	2614	9249	1414	2326	14421	18243	15655	0	6132	0	1238	0	0
Coffee	12	266444	78262	109152	461	669	4	40	0	69	567	22	0	0	0	0	0	419	0	0	0	0	0
Other Temporary Crops	0	260547	3001	149419	1279	641	6293	1552	615	0	0	992	214	58	40	1	244	1	0	0	0	0	0
Rocky Outcrop	162	23492	1505	25224	197	0	2	3027	1083	0	0	175	0	0	0	0	1609	23	0	0	129	0	0
Urban Area	1	1256	53	2644	0	6	35	7138	1	19	0	0	36	39	7780	167	236	0	138	0	35	0	0
Herbaceous Sandbank Vegetation	0	39405	12	31397	0	630	1409	1261	0	683	0	4304	0	86	94554	26	1320	0	200	0	0	0	0
Wooded Sandbank Vegetation	0	23424	288	33984	0	18302	1334	2698	0	120	0	812	117	0	168	4441	107	0	83	0	0	0	0
Beach and Sand	4	8031	9	22366	0	22	8663	10024	0	1	0	8310	80709	81	0	13	10366	0	21	0	0	0	0
Mangrove	0	39	5879	862	0	3258	3824	15437	0	0	0	123	26	2816	21	0	46	0	1516	0	0	0	0
Other non Vegetated Areas	6	89550	4215	44960	350	4748	3213	19059	12	731	1163	4387	1092	348	12293	41	0	1	18	0	51	0	0
Other Perennial Crops	2	84900	37962	34592	191	45	20	1	2850	1	51	2	0	0	0	0	0	0	0	0	0	0	0
Hypersaline Tidal Flat	0	9	301	506	0	6	3397	5285	0	0	0	318	157	129	8	2882	53	0	0	0	0	0	0
Sugar Cane	0	3	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mining	0	288	13	580	239	0	0	491	0	0	74	35	0	0	0	0	41	0	0	0	0	0	0
Aquaculture	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Grassland	0	10	0	4	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

From/To	No data	Pasture	Forest Formation	Mosaic of Uses	Savanna Formation	Forest Plantation	Wetland	River Lake and Ocean	Coffee	Other Temporary Crops	Rocky Outcrop	Urban Area	Herbaceous Sandbank Vegetation	Wooded Sandbank Vegetation	Beach and Sand	Mangrove	Other non Vegetated Areas	Other Perennial Crops	Hypersaline Tidal Flat	Sugar Cane	Mining	Aquaculture	Grassland
No data	0	5846	19393	18405	632	112	93	668	100	13	692	25	0	3	8	0	31	1	0	0	0	0	0
Pasture	6573	0	3139137	58964633	959706	4532371	501583	500916	2566346	1569717	49390	444788	88355	17406	23880	164	128007	51616	49	120723	4723	14	102
Forest Formation	18345	6326411	0	14660443	29537	2152552	21516	242267	322912	15828	8162	10232	15	1009	89	19160	6999	5170	911	3	464	0	0
Mosaic of Uses	23649	55192339	16829759	0	955939	2271296	621913	999812	1194748	1482987	65641	289126	88961	108769	75338	4037	180096	28801	1686	5283	2788	18	27
Savanna Formation	562	2244730	32854	817173	0	14489	0	16707	3244	1197	1072	756	0	0	0	0	3533	175	0	0	1021	0	5
Forest Plantation	69	669665	1147885	689200	2647	0	119717	2569	3231	3039	3	4578	923	39971	96	4875	676	18	56	206	7	0	0
Wetland	103	1024486	19784	610205	0	134443	0	497381	3906	125334	0	17754	3529	2391	28597	18150	13896	67	16591	64	86	34	0
River Lake and Ocean	666	410671	262286	1018505	16579	3018	520169	0	193	37420	8565	31202	3851	9432	39284	62847	89866	1	21180	3	1964	433	0
Coffee	65	1146794	259324	751081	2484	5503	2560	146	0	99922	2593	214	1	16	0	0	54	1094	0	0	0	0	0
Other Temporary Crops	1	1527384	15080	579879	2290	7304	108841	24640	89970	0	3	25339	5030	2058	716	4	3529	1014	0	4302	0	0	0
Rocky Outcrop	3161	55624	7990	81342	832	410	2	9118	3653	1	0	351	0	0	0	0	10440	53	0	0	373	0	0
Urban Area	10	6181	385	11512	7	28	254	22583	20	538	4	0	175	215	21315	652	1446	0	386	0	169	4	0
Herbaceous Sandbank Vegetation	0	117867	17	108290	0	928	3942	3849	0	5829	0	11514	0	343	241529	117	16153	0	591	0	0	0	0
Wooded Sandbank Vegetation	1	42352	989	90387	0	21328	2982	9240	13	476	0	2803	397	0	392	12638	761	0	122	0	0	0	0
Beach and Sand	7	24737	60	87455	0	129	33772	39941	0	227	0	36321	306562	365	0	58	93463	0	36	0	0	0	0
Mangrove	0	164	18040	3618	0	4357	18343	60212	0	2	0	649	130	10492	80	0	173	0	8299	0	0	0	0
Other non Vegetated Areas	37	127804	5542	142318	2455	5117	14665	79626	47	1209	13811	15708	12628	564	70384	155	0	1	85	0	770	0	4
Other Perennial Crops	3	86284	38527	36193	193	70	20	1	3160	34	58	3	0	0	0	0	1	0	0	0	0	0	0
Hypersaline Tidal Flat	0	77	1299	1498	0	16	14932	18744	0	0	0	979	563	237	117	11152	128	0	0	0	0	23	0
Sugar Cane	0	7048	2	1336	0	322	9	0	0	2191	0	0	0	0	0	0	2	0	0	0	0	0	0
Mining	0	376	21	682	247	0	9	509	0	0	77	35	0	0	0	0	68	0	0	0	0	0	0
Aquaculture	0	2	0	13	0	0	12	210	0	0	0	3	0	0	0	0	0	0	26	0	0	0	0
Grassland	0	34	0	11	91	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

From/To	No data	Pasture	Forest Formation	Mosaic of Uses	Savanna Formation	Forest Plantation	Wetland	River Lake and Ocean	Coffee	Other Temporary Crops	Rocky Outcrop	Urban Area	Herbaceous Sandbank Vegetation	Wooded Sandbank Vegetation	Beach and Sand	Mangrove	Other non Vegetated Areas	Other Perennial Crops	Hypersaline Tidal Flat	Sugar Cane	Mining	Aquaculture	Grassland
No data	0	940	4089	3776	245	37	14	287	27	2	182	7	0	0	1	0	13	0	0	0	0	0	0
Pasture	1600	0	830768	12509474	135967	683340	27431	97476	432180	123872	10710	85053	23031	3483	948	28	74895	47447	5	115081	1436	14	0
Forest Formation	3360	741119	0	2231126	3468	194045	2481	52805	45999	3390	923	1535	2	194	3	2168	4463	3947	260	2	121	0	0
Mosaic of Uses	8089	9037510	4423131	0	118767	519421	33498	317497	277890	29770	13475	51124	19576	29182	3270	665	83763	26498	197	2477	378	18	9
Savanna Formation	234	237465	7745	182495	0	7941	0	2348	747	268	363	46	0	0	0	0	1458	159	0	0	94	0	1
Forest Plantation	18	136172	38469	135698	56	0	2044	90	1069	952	1	612	63	345	1	40	95	4	0	206	2	0	0
Wetland	12	113926	4159	109141	0	25761	0	139739	383	2419	0	1848	698	418	1225	3771	4284	67	3731	52	1	34	0
River Lake and Ocean	187	52480	52741	224049	2257	631	96313	0	4	1364	1758	5984	676	2122	2363	7731	26925	1	2878	3	107	433	0
Coffee	26	256939	59614	202596	509	1810	1268	30	0	64451	483	170	1	11	0	0	19	515	0	0	0	0	0
Other Temporary Crops	1	237827	3641	47022	190	2694	8077	2393	60821	0	3	11304	1390	44	46	0	699	1013	0	2145	0	0	0
Rocky Outcrop	2646	7918	2411	17584	244	408	0	1753	640	0	0	86	0	0	0	0	4888	28	0	0	8	0	0
Urban Area	5	1630	143	3215	2	9	61	3261	9	17	2	0	42	56	1064	80	507	0	47	0	41	4	0
Herbaceous Sandbank Vegetation	0	31152	0	22633	0	87	469	643	0	280	0	2646	0	65	8014	25	6806	0	128	0	0	0	0
Wooded Sandbank Vegetation	0	3611	164	12732	0	939	285	2670	8	120	0	444	100	0	32	891	314	0	10	0	0	0	0
Beach and Sand	1	2444	8	8289	0	102	3887	3936	0	39	0	2309	35143	53	0	4	10682	0	1	0	0	0	0
Mangrove	0	45	5020	861	0	2	4907	13161	0	0	0	167	27	1729	5	0	54	0	3119	0	0	0	0
Other non Vegetated Areas	19	10802	518	28280	1404	58	1774	26446	6	97	8910	3503	4737	82	3472	18	0	0	39	0	106	0	0
Other Perennial Crops	1	440	161	703	0	12	0	0	150	33	5	0	0	0	0	0	1	0	0	0	0	0	0
Hypersaline Tidal Flat	0	14	633	150	0	0	2745	3367	0	0	0	60	70	54	0	1224	4	0	0	0	0	23	0
Sugar Cane	0	1736	2	377	0	59	2	0	0	257	0	0	0	0	0	0	2	0	0	0	0	0	0
Mining	0	32	4	41	5	0	0	9	0	0	3	0	0	0	0	0	12	0	0	0	0	0	0
Aquaculture	0	2	0	13	0	0	12	210	0	0	0	3	0	0	0	0	0	0	26	0	0	0	0
Grassland	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Amazon Forest change matrix

From/To	Forest Formation	Pasture	Floodable Forest	River Lake and Ocean	Wetland	Savanna Formation	Other Temporary Crops	Grassland	Soybean	Urban Area	Rocky Outcrop	Mining	Other non Vegetated Areas	Forest Plantation	No data
Forest Formation	0	11221119	4974	23081	125446	21013	418	15261	0	11930	560	24445	74	256	12
Pasture	3382576	0	391420	77090	189131	20335	45	90052	0	41503	63	7854	3505	115	1
Floodable Forest	2940	676475	0	520114	858396	18	992	111	0	1128	3	6817	7	65	1
River Lake and Ocean	10816	51857	243076	0	1963631	54	13	110730	0	928	31	8722	1	36	1
Wetland	129500	234459	715316	2899584	0	326	38	11742	0	1953	26	5817	515	181	0
Savanna Formation	3122	45897	11	108	226	0	0	439	0	424	24	2682	25	0	0
Other Temporary Crops	1441	23	792	6	76	17	0	25	0	0	0	0	0	0	0
Grassland	7912	114144	84	69637	21345	469	9	0	0	3256	412	9522	3355	8	1
Soybean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Urban Area	260	921	38	1064	65	20	0	38	0	0	0	422	12	0	0
Rocky Outcrop	230	134	0	38	22	23	0	199	0	3	0	0	0	0	0
Mining	4240	11774	814	14291	135	31	0	561	0	419	0	0	0	0	0
Other non Vegetated Areas	27	3836	9	15	678	2	0	2308	0	2921	0	0	0	20	0
Forest Plantation	0	7	2	5	11	0	0	0	0	2	0	0	0	0	0
No data	9	3	1	2	0	0	0	1	0	0	0	0	0	0	0

From/To	Forest Formation	Pasture	Floodable Forest	River Lake and Ocean	Wetland	Savanna Formation	Other Temporary Crops	Grassland	Soybean	Urban Area	Rocky Outcrop	Mining	Other non Vegetated Areas	Forest Plantation	No data
Forest Formation	0	65889171	22240	85278	214433	105468	95367	71663	3674	15783	4108	79436	6226	328	20
Pasture	17346896	0	1680360	390104	743867	201927	2220442	284577	47997	96972	821	100143	92197	162	9
Floodable Forest	18122	2756785	0	1521917	1849003	117	11383	361	104	2365	15	28409	2356	100	7
River Lake and Ocean	62269	339489	1253750	0	8679551	357	20590	386692	281	3639	174	41301	3412	89	6
Wetland	190328	743075	1854948	9853859	0	1222	86617	34438	484	3212	72	11009	17313	266	0
Savanna Formation	21416	349945	130	552	1180	0	472	2466	25	2122	108	7119	469	0	0
Other Temporary Crops	20255	817278	7278	21096	58758	244	0	6059	1674909	5944	19	726	858	0	0
Grassland	28962	391711	328	320105	97225	2856	18139	0	882	4548	2379	21236	11417	9	2
Soybean	651	4511	74	116	100	1	480553	107	0	1303	0	0	56	0	0
Urban Area	669	2555	205	3520	273	42	352	156	105	0	0	3044	670	0	0
Rocky Outcrop	1988	3053	6	179	74	168	21	2027	0	3	0	307	0	0	0
Mining	10155	30394	4715	38999	2663	198	16	4301	0	1811	42	0	11	0	0
Other non Vegetated Areas	741	30428	676	1714	12007	49	481	6770	92	5964	0	10	0	20	0
Forest Plantation	3	18	7	28	28	0	4	2	0	2	0	0	1	0	0
No data	15	24	3	6	1	1	0	3	0	0	0	0	0	0	0

From/To	Forest Formation	Pasture	Floodable Forest	River Lake and Ocean	Wetland	Savanna Formation	Other Temporary Crops	Grassland	Soybean	Urban Area	Rocky Outcrop	Mining	Other non Vegetated Areas	Forest Plantation	No data
Forest Formation	0	18823898	5377	33148	31162	21601	18645	31682	2357	928	1836	47764	6057	31	4
Pasture	3827344	0	281302	123897	240009	68773	320558	107359	29001	14284	355	61553	76849	20	1
Floodable Forest	5569	632028	0	400679	339621	20	3404	128	99	723	12	15437	2302	21	2
River Lake and Ocean	17725	65884	300456	0	2146234	141	6190	43352	186	1131	40	20390	2896	4	1
Wetland	22379	166230	427552	1842634	0	176	10549	7344	274	493	19	2824	14287	11	0
Savanna Formation	7423	148181	89	214	246	0	228	819	25	362	50	2380	407	0	0
Other Temporary Crops	12012	289974	3127	8450	18108	29	0	1960	1035758	2438	10	91	453	0	0
Grassland	9862	138533	116	38616	12651	1386	4697	0	198	290	1412	3525	6079	0	0
Soybean	610	3367	69	59	73	1	214571	22	0	1182	0	0	48	0	0
Urban Area	147	628	84	1094	75	12	117	39	33	0	0	681	510	0	0
Rocky Outcrop	749	2298	6	52	18	104	7	1039	0	0	0	133	0	0	0
Mining	1125	3973	294	13497	321	49	16	581	0	425	14	0	8	0	0
Other non Vegetated Areas	632	19901	642	1025	9312	12	173	1778	85	2496	0	10	0	0	0
Forest Plantation	1	5	0	10	2	0	4	0	0	0	0	0	0	0	0
No data	5	5	2	2	1	0	0	1	0	0	0	0	0	0	0

Error Handling

The script implements several error handling mechanisms:

1. **Retry Logic**: Failed read operations are retried with exponential backoff
2. **Window Validation**: Ensures requested windows are within raster bounds
3. **Worker Management**: Tracks and reports failed tiles without stopping the process
4. **Memory Monitoring**: Adjusts processing based on available system memory
5. **Data Validation**: Checks for valid data and handles exceptions appropriately

Performance Considerations

- Processing an 8×8 degree region (~880×880 km) requires approximately 32GB RAM
- Full processing including visualizations takes 3-5 hours on a modern system
- The script dynamically adjusts to available system resources
- Parallel processing reduces time but increases memory usage
- VRT-specific optimizations improve performance with large datasets

Usage

```
# Define the VRT file path
VRT_FILE = '/path/to/mapbiomas_coverage_1985_2023.vrt'

# Define the output directory
OUTPUT_BASE_DIR = 'amazon_analysis_results'

# Define the analysis region (coordinates in lon, lat format)
AMAZON_REGION = [((-55, -2), (-55, -7), (-51, -7), (-51, -2), (-55, -2))]

# Run the analysis
zarr_path, grid_output_dir = extract_grid_data_optimized(
    VRT_FILE, AMAZON_REGION, OUTPUT_BASE_DIR
)

# Create visualizations
create_visualizations(zarr_path, grid_output_dir)
```

Example Analysis: Central Amazon Region (2° to 7°S, 51° to 55°W)

This region encompasses parts of the states of Pará and Amazonas, including sections of the Tapajós National Forest and areas affected by the BR-163 highway development corridor. The analysis reveals:

1. **Forest Loss Patterns**: Deforestation fronts advancing from the southeast
2. **Transition Hotspots**: Primary forest to pasture conversion dominant near highways
3. **Temporal Patterns**: Acceleration of land cover change after 2005
4. **Land Use Intensification**: Agricultural expansion in previously cleared areas
5. **Protected Area Effectiveness**: Lower change frequencies within protected boundaries

Troubleshooting

- **Memory Errors**: Reduce `MAX_WORKERS` or increase tile size
- **Failed Tiles**: Check the log for specific error messages
- **VRT Issues**: Ensure the VRT file correctly references all source TIFFs
- **Visualization Errors**: If Sankey diagrams fail, check for missing class colors

---

Created by Leandro Biondo, UBCO - Sustainability, 2025

Brazil Color Table Amazon Analysis Documentation

1. Introduction and Setup

This script performs analysis of land cover changes in the Amazon region of Brazil using MapBiomas data. It processes multi-year raster data, identifies land cover transitions, and creates visualizations of the changes.

Key Dependencies

```
import rasterio
import numpy as np
import zarr
import multiprocessing
from concurrent.futures import ProcessPoolExecutor, as_completed
import psutil
import matplotlib.pyplot as plt
import plotly.graph_objects as go
```

Configuration Constants

```
# Memory and performance tuning parameters
TILE_SIZE = 256               # Size of processing tiles in pixels
MAX_WORKERS = 10              # Maximum number of parallel worker processes
MEMORY_BUFFER_GB = 5          # Memory buffer to maintain in GB
MAX_READ_RETRIES = 3          # Maximum retries for failed reads
```

```
RETRY_DELAY = 0.05            # Delay between retries in seconds
(exponential backoff)
```

2. MapBiomas Classification System

The script uses the official MapBiomas land cover classification system with 50+ classes. Each class has an assigned color and label for visualization.

```
# Color mapping (simplified example)
COLOR_MAP = {
    0: "#ffffff",                 # No data
    1: "#1f8d49", 3: "#1f8d49",   # Forest types
    4: "#7dc975",                 # Savanna
    15: "#edde8e",                # Pasture
    24: "#d4271e",                # Urban Area
    33: "#259fe4"                 # Water bodies
    # ... many other classes ...
}

# Class labels
LABELS = {
    0: "No data",
    1: "Forest", 3: "Forest Formation",
    4: "Savanna Formation",
    # ... many other classes ...
}
```

3. Robust Read Functionality

The script implements fault-tolerant reading of geospatial data with retries and validation.

```
def robust_read(src, band, window, retries=MAX_READ_RETRIES):
    """Attempt to read a tile with retries on failure."""
    for attempt in range(retries):
        try:
            # Validate window is within bounds
            valid_window = window.intersection(Window(0, 0, src.width,
src.height))
            if valid_window is None or valid_window.width <= 0 or
valid_window.height <= 0:
                raise ValueError("Requested window is outside the raster
bounds")

            # Read data
            data = src.read(band, window=valid_window)

            # Validate data
            if np.all(data == src.nodata):
                raise ValueError("All values are nodata")
```

```
            return data
        except Exception as e:
            # Retry with exponential backoff
            if attempt == retries - 1:
                raise
            time.sleep(RETRY_DELAY * (attempt + 1))

    raise RuntimeError(f"Failed after {retries} attempts")
```

4. Tile Processing

The script processes large raster data in manageable tiles using parallel workers:

```
def process_tile(task):
    """Process a single tile for changes and transitions."""
    y, x, path, window, temp_dir = task

    try:
        with rasterio.open(path) as src:
            changes = np.zeros((window.height, window.width),
dtype='uint8')
            transitions = np.zeros((256, 256), dtype='uint64')

            # Read first band/year
            prev_data = src.read(1, window=window, fill_value=FILL_VALUE)

            # Process subsequent bands/years
            for band_idx in range(2, src.count + 1):
                curr_data = src.read(band_idx, window=window,
fill_value=FILL_VALUE)

                # Identify changed pixels
                changed = prev_data != curr_data
                changes += changed

                # Record transitions
                if np.any(changed):
                    from_vals = prev_data[changed]
                    to_vals = curr_data[changed]
                    for f, t in zip(from_vals, to_vals):
                        transitions[f, t] += 1

                prev_data = curr_data

            return changes, transitions, (y, x)

    except Exception as e:
        logging.warning(f"Tile {y},{x} failed: {str(e)}")
        return None
```

5. Optimized Data Extraction

The main data extraction function processes a geographic region by dividing it into manageable tiles:

```
def extract_grid_data_optimized(vrt_path, polygon_coords, output_dir):
    """Optimized extraction with VRT-aware settings."""
    # VRT-specific configuration
    VRT_BLOCK_SIZE = 512  # Matches the BlockXSize/BlockYSize in VRT
    TILE_SIZE = 2048      # Multiple of VRT_BLOCK_SIZE for optimal
performance

    # Important: Getting window aligned to VRT blocks improves performance
    def align_to_blocks(value, block_size, max_value, align_type='floor'):
        if align_type == 'floor':
            return max(0, (value // block_size) * block_size)
        else:
            return min(max_value, ((value + block_size - 1) // block_size)
* block_size)

    # Calculate window based on geographic coordinates
    # ... coordinate conversion code ...

    # Process data in parallel using tiles
    with ProcessPoolExecutor(max_workers=MAX_WORKERS) as executor:
        futures = []
        for task in tasks:
            futures.append(executor.submit(process_tile, task))

        # Process results as they complete
        for future in progress:
            result = future.result()
            if result is not None:
                changes, transitions, (y, x) = result
                # ... store results ...
            else:
                failed_tiles.append((y, x))

    # Store data in Zarr format with compression
    zarr_path = os.path.join(grid_output_dir, 'data.zarr')
    root = zarr.open(zarr_path, mode='w')

    # Important: Chunking matters for performance
    root.zeros(
        'changes',
        shape=full_changes.shape,
        chunks=(VRT_BLOCK_SIZE, VRT_BLOCK_SIZE),




# Iguaçu National Park Land Cover Analysis
```

```
![alt text](image.png)

## 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

```
- MapBiomas 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 MapBiomas project:

- **Source**: MapBiomas Collection (recommended Collection 8 or later)
- **Resolution**: 30m
- **Projection**: EPSG:4326 (WGS84)
- **Time Range**: 1985-2023
- **Classification**: MapBiomas 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 MapBiomas 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:
 ```python
 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

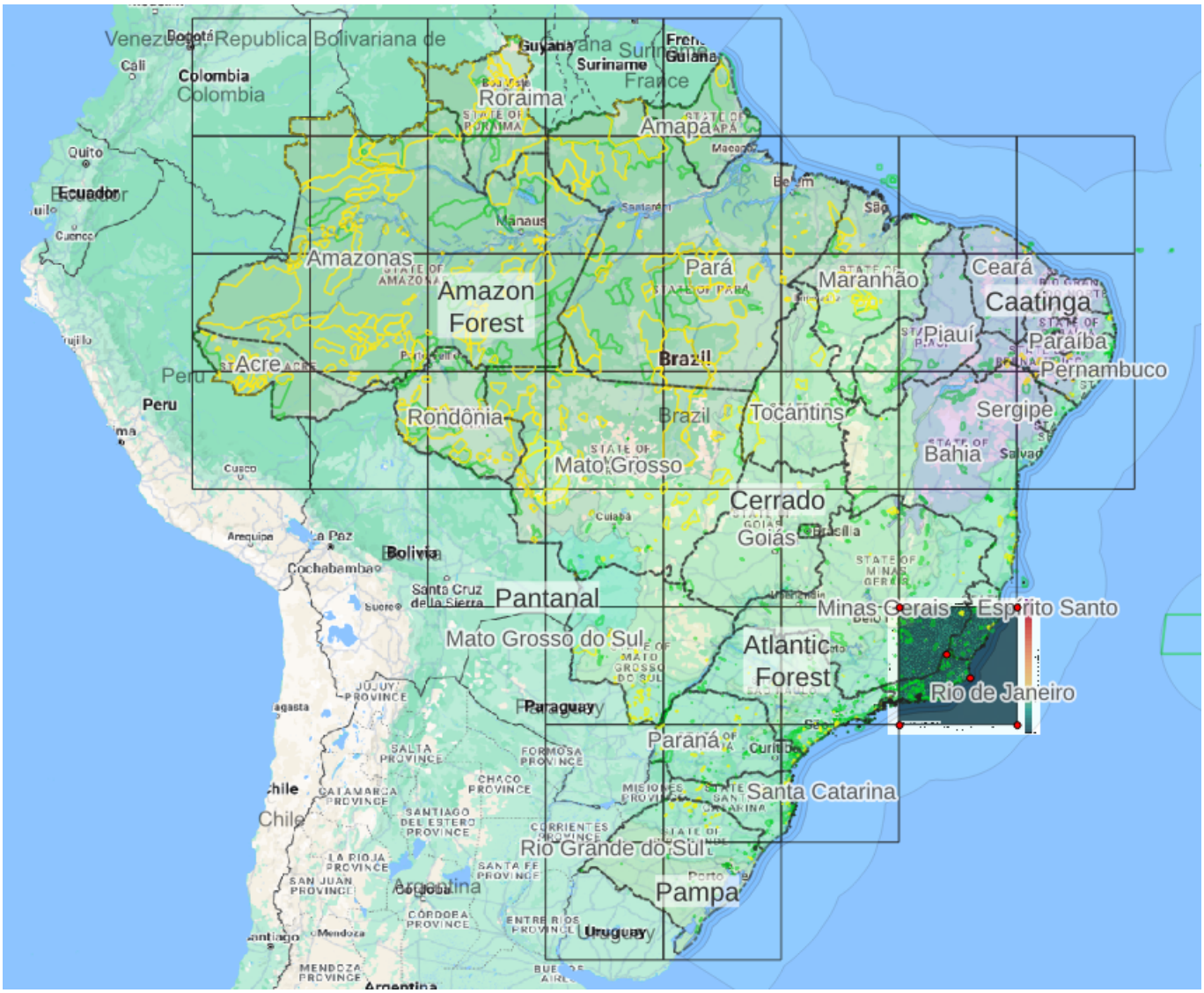
- Processing the full dataset requires approximately 2-3GB of RAM
- The script automatically filters out small, insignificant transitions (less than 0.1% of total)
- Visualization includes annotations with statistics about change patterns
- All visualizations include proper geographic coordinates in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

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

MapBiomas Brazil Grid Analysis

Overview

This Python script analyzes land cover change patterns across Brazil using MapBiomas 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
```

- MapBiomas Brazil Collection 8 data as a VRT file

Data Source

This script processes the MapBiomas 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)

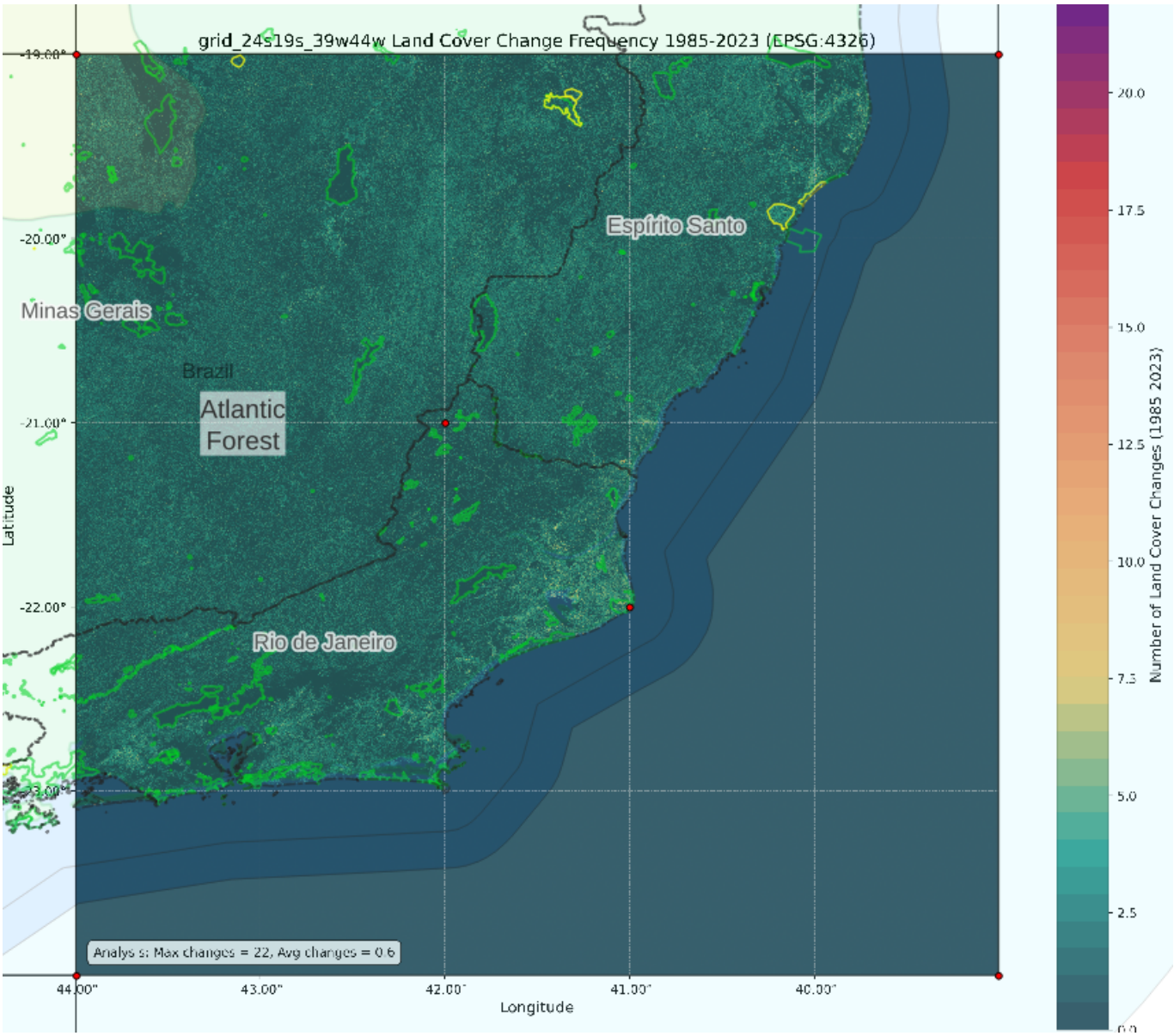
Grid System

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 MapBiomas 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

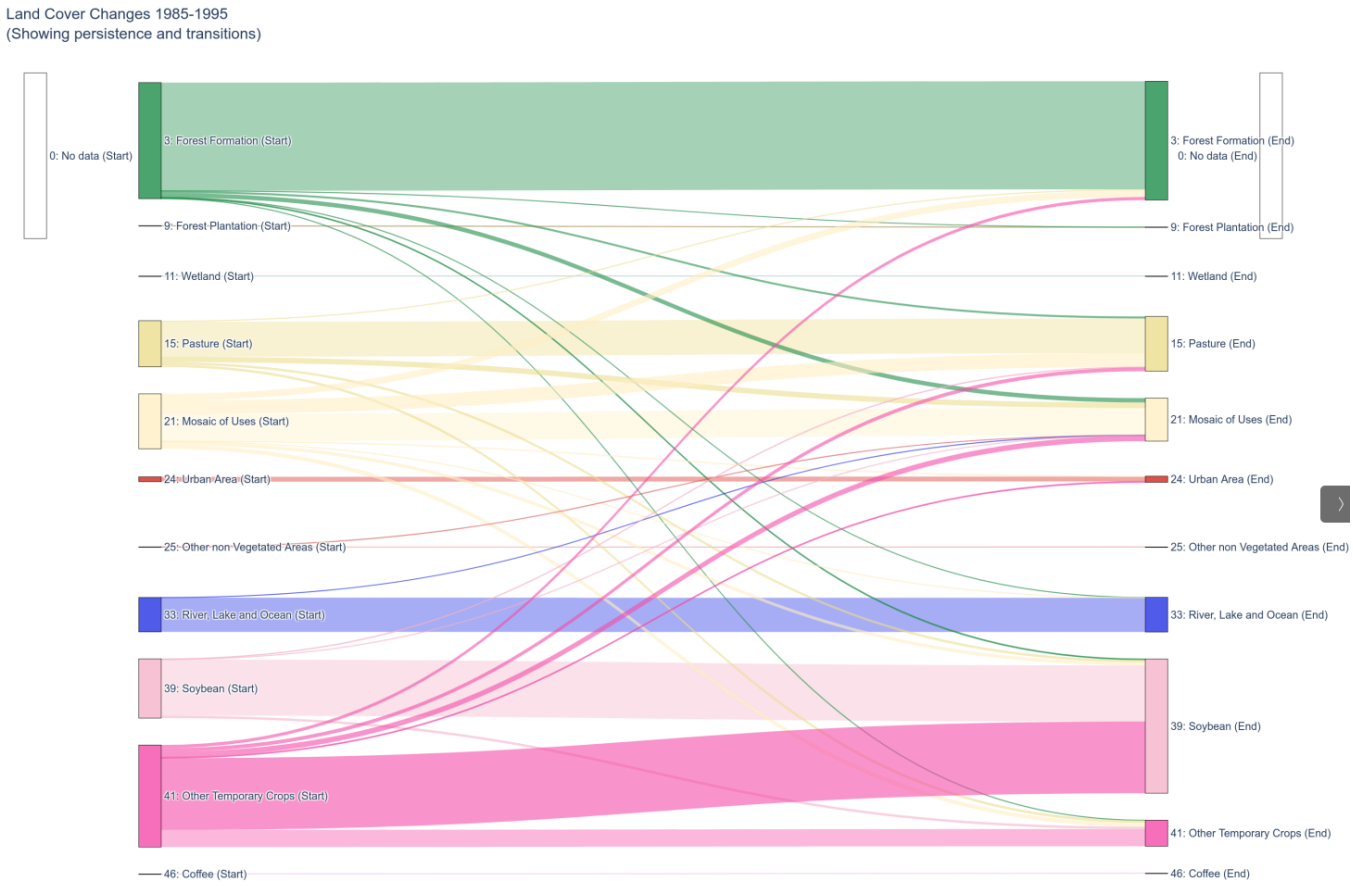
---

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 MapBiomas grid datasets in Zarr format

Data Sources

This script expects pre-processed MapBiomas 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.