GeoPAT 2.0 user's manual

Note: This manual is work in progress

`http://sil.uc.edu`

November 7, 2017

Contents

1 Introduction **4**
1.1 Basic concepts . . . 4

2 GeoPAT 2.0 architecture **5**
2.1 General description . . . 5
2.2 GeoPAT Modules . . . 6
2.2.1 Signature building . . . 6
2.2.1.1 gpat_gridhis . . . 6
2.2.1.2 gpat_gridts . . . 9
2.2.1.3 gpat_pointshis . . . 11
2.2.1.4 gpat_pointsts . . . 13
2.2.1.5 gpat_polygon . . . 14
2.2.2 Similarity measuring . . . 16
2.2.2.1 gpat_search . . . 16
2.2.2.2 gpat_compare . . . 18
2.2.2.3 gpat_segment . . . 20
2.2.2.4 gpat_distmtx . . . 23
2.2.3 Tools . . . 24
2.2.3.1 gpat_grd2txt . . . 24
2.2.3.2 gpat_globnorm . . . 25
2.2.3.3 gpat_segquality . . . 26

3 Basic workflow paths with examples **29**
3.1 Search . . . 30
3.1.1 Search on categorical maps . . . 31
3.1.2 Search on time series . . . 33
3.2 Change detection . . . 35
3.3 Segmentation . . . 36
3.3.1 Segmentation on categorical maps . . . 37
3.3.2 Segmentation on time series . . . 40
3.4 Clustering . . . 42
3.4.1 Clustering of individual motifels . . . 43
3.4.2 Clustering of a grid of motifels . . . 47
3.4.3 Clustering of segments / predefined irregular regions . . . 48

Appendices **50**

A GeoPAT 2.0 installation **50**
A.1 System requirements . . . 50
A.2 Windows installer . . . 50
A.3 Building from source code . . . 50

B Numerical signatures available in GeoPAT **52**
B.1 Cartesian product (*prod*) . . . 52
B.2 Class co-occurrence histogram (*cooc*) . . . 52
B.3 Decomposition histogram (*fdec* and *sdec*) . . . 52
B.4 Landscape indices vector (*lind* and *linds*) . . . 53
B.5 Shannon entropy (*ent*) . . . 55

C Normalization methods available in GeoPAT **56**
C.1 01 . . . 56
C.2 pdf . . . 56
C.3 N01 . . . 56
C.4 ind01 . . . 56
C.5 none . . . 56

D Dissimilarity measures available in GeoPAT **57**
D.1 Jensen Shannon Divergence (*jsd*) . . . 57
D.2 Euclidean distance (*euc*) . . . 57
D.3 Normalized euclidean distance (*eucn*) . . . 57
D.4 Wave-Hedges distance (*wh*) . . . 57
D.5 Jaccard distance (*jac*) . . . 58
D.6 Euclidean distance - time series (*tsEUC*) . . . 58
D.7 Euclidean distance - periodic time series (*tsEUCP*) . . . 58
D.8 Dynamic Time Warping distance - time series (*tsDTW*) . . . 58
D.9 Dynamic Time Warping distance - periodic time series (*tsDTWP*) . . . 58
D.10 Synchronized Dynamic Time Warping distance - time series (*tsDTWPa*) . . . 58

E Topologies available in GeoPAT **59**

References **62**

1 Introduction

GeoPAT 2.0 (Geospatial Pattern Analysis Toolbox) is a standalone suite of modules written in C and dedicated to analysis of large Earth Science datasets in their entirety using spatial and/or temporal patterns. Global scale, high resolution spatial datasets are available but are mostly used in small pieces for local studies. GeoPAT enables studying them in their entirety. GeoPAT's core idea is to tessellate global spatial data into grid of square blocks of original cells (pixels). This transforms data from its original form (huge number of cells each having simple content) to a new form (much smaller number of supercells/blocks with complex content). Complex cell contains a pattern of original variable. GeoPAT provides means for succinct description of such patterns and for calculation of similarity between patterns. This enables spatial analysis such as search, change detection, segmentation, and clustering to be performed on the grid of complex cells (local patterns).

1.1 Basic concepts

- **Grid** is a topological structure applied to input data. Grid is a lattice of cells defined by its geographical position, spatial extent, number of cells, and cell size which defines grid resolution. Each cell in the grid contains a vector of values.
- **Pattern signatures** - compact numerical descriptors of patterns. We use histograms of pattern's features as scene signatures.
- **Motifel** is an elementary unit of analysis - a square block of pixels representing local pattern. Motifel is represented by histogram of features (co-occurrence, decomposition). Distance between motifels is a distance between their histograms (for example using Jensen-Shannon Divergence).

2 GeoPAT 2.0 architecture

2.1 General description

The GeoPAT consists of twelve modules (Fig. 1). Five modules are designed for extracting pattern signatures from an original data grid, four modules for actual geoprocessing of the patterns, and three utility modules.

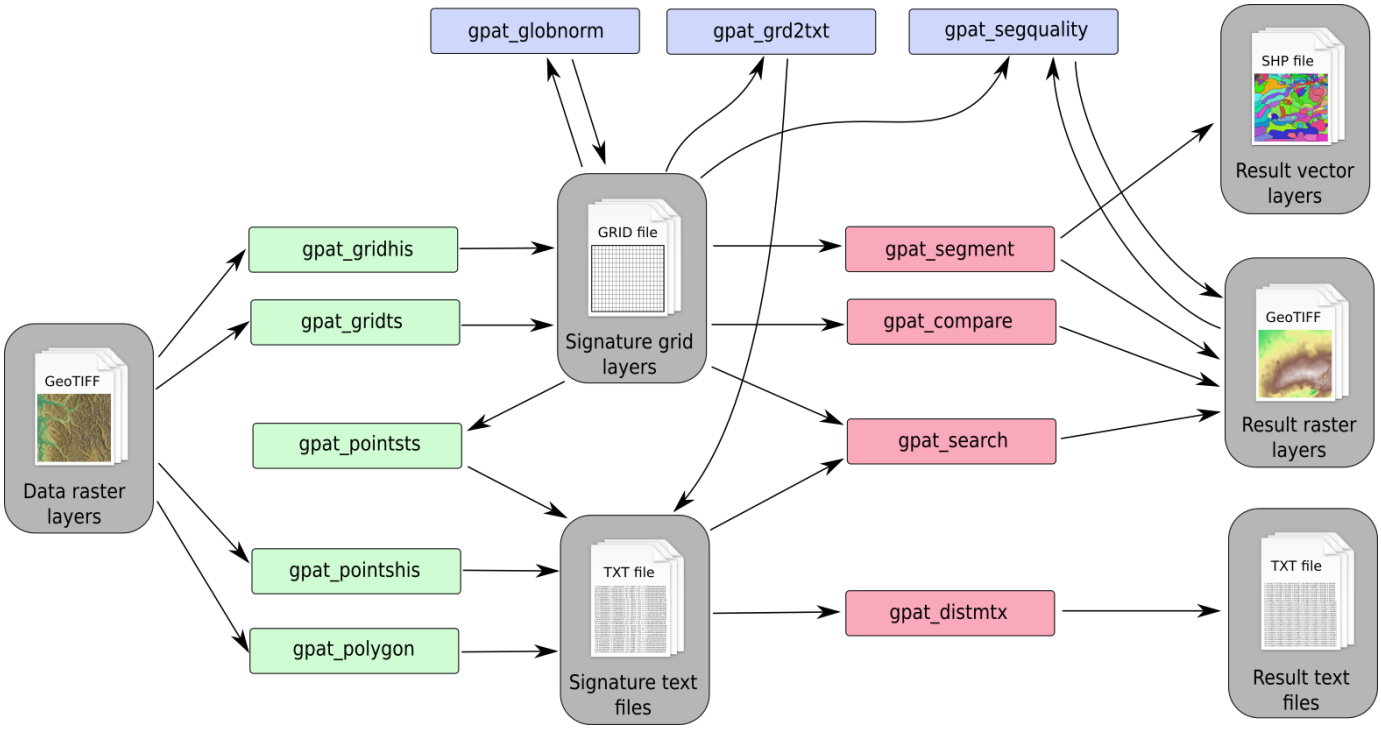

Figure 1: Outline of the GeoPAT 2.0 architecture

The role of signature extraction modules (`gpat_gridhis`, `gpat_gridts`, `gpat_pointshis`, `gpat_pointsts`, `gpat_polygon`) is to calculate pattern signatures for scenes defined:

- `gpat_gridhis` - by a neighborhood of each cell in a grid (spatial pattern)
- `gpat_gridts` - by a neighborhood of each cell in a grid (spatiotemporal pattern)
- `gpat_pointshis` - in the neighborhoods of selected points (spatial pattern)
- `gpat_pointsts` - in the neighborhoods of selected points (spatiotemporal pattern)
- `gpat_polygon` - over irregular polygons

The role of geoprocessing modules (`gpat_search`, `gpat_compare`, `gpat_segment`, `gpat_distmtx`) is to perform:

- `gpat_search` - searching

- `gpat_compare` - comparision
- `gpat_segment` - segmentation (based on pattern data generated by the signature extraction modules)
- `gpat_distmtx` - clustering

The role of utility modules (`gpat_grd2txt`, `gpat_globnorm`, `gpat_segquality`) is to:

- `gpat_grd2txt` - convert grid of signatures (grid-of-scenes) from a binary to text
- `gpat_globnorm` - normalization of grid of signatures (grid-of-scenes) values
- `gpat_segquality` - calculate the quality metrics (inhomogeneity and isolation) of a segmentation

2.2 GeoPAT Modules

2.2.1 Signature building

2.2.1.1 gpat_gridhis

Creates a binary grid of signatures from a categorical raster map(s).

Usage:

```
gpat_gridhis [−lh] −i <file_name> −o <file_name> [−s <signature_name>] [−−level=<n>] [−z <n>]
    [−f <n>] [−n <normalization_name>] [−t <n>]

−i, −−input=<file_name> name of input file (GeoTIFF)
−o, −−output=<file_name> name of output file (GRID)
−s, −−signature=<name> motifel's signature (use −l to list all signatures, default: 'cooc')
−−level=<n> full decomposition level (default: 0, auto)
−z, −−size=<n> motifel size in cells (default: 150)
−f, −−shift=<n> shift of motifels (default: 100)
−n, −−normalization=<name> signature normalization method (use −l to list all methods, default: 'pdf')
−l list all signatures and normalization methods
−t <n> number of threads (default: 1)
−h, −−help print this help and exit
```

Options:

input

Defines a categorical raster layer(s) which will be used as a source for extracting pattern signature. Layers must be a categorical one. For the Cartesian product method ('prod') there can be more than one input map. Other methods use a one map only (the first one provided). In order to provide more than one input map, type multiple input options ("-i map1.tif -i map2.tif" or "–input=map1.tif –input=map2.tif").

output

Output consists of two files: one of them is a dataset containing a grid of signatures in binary form, the other one is a header text file (the .hdr extension) containing a grid topology and an information about the input data parameters. Modifying the header is strongly discouraged as it may cause some calculations to fail. Structure of the header is as follows:

- dim – number of dimensions of each signature
- dims – size of each dimension
- type – type of data stored in the grid (integer, float, etc.)
- at0 – top left x
- at1 – w-e grid resolution
- at2 – rotation (0 if grid is north-up)
- at3 – top left y
- at4 – rotation (0 if grid is north-up)
- at5 – n-s grid resolution
- rows – number of rows in the grid
- cols – number of cols in the grid
- proj – projection in wkt style
- desc – command used to create the grid

size

Size of a motifel (local calculation window) expressed in the number of pixels. It defines the extent of a local pattern. It is the length of the side of square-shaped block of pixels (motifel). It must be at least 10 and cannot exceed the input map size.

shift

Parameter defines the shift between adjacent scenes along the grid in n-s and w-e directions. It describes the density of the output gird and defines a new topology of the grid. Formula original_resolution $\times$ shift = new_resolution explains how resolution of the original map will be reduced. If shift is set to the same value as 'size', the input map will be simply divided into a grid of non-overlapping motifels. Setting shift to a value smaller than 'size' parameter will result in grid of overlapping motifels. Shift cannot be larger than 'size', and cannot be smaller than 5.

signature

Defines method of calculating signatures of motifels. GeoPAT offers the following methods:

- prod – Cartesian product of input category lists
- cooc – Spatial coocurrence of categories
- sdec – Simple 2-level decomposition
- fdec – Full decomposition
- lind – Landscape indices vector
- linds – Selected landscape indices vector
- ent – Shannon entropy

See Appendix B for the details.

level

This option is used only if 'full decomposition' (fdec) is used. It defines the highest level of decomposition. See Appendix B for the details.

normalization

Specifies normalization method used on the local signatures. See Appendix C for the details. Normalization method used should depend on a selected signature method. For example, the default pdf normalization works well for the histogram-based signatures (pred, cooc, sdec, fdec), while normalization should be disabled (none) for the landscape indices signatures (lind, linds) and the Shannon entropy signature.

threads (-t)

The module is parallel, therefore use more than one processing thread can be used in order to speed up calculations. This option specifies how many threads will be used. The default is 1.

Description:

This module extracts a "grid-of-scenes" (grid of pattern signatures, grid of motifels). The output is a grid of the same spatial extent as the input raster map, but with a different cell size. Each cell in a new grid has only one attribute - the signature of its pattern. Pattern is calculated over a window centered on the cell and having a user-defined size. Resolution of the output grid equals to the resolution of the input raster map multiplied by the shift parameter. A signature of pattern for each scene is stored as a numerical vector in a binary form. The module outputs a header file (.hdr) containing a topology of a grid-of-scenes and a binary file containing signatures ordered by rows.

This module uses categorical raster data in GeoTIFF format as an input. It might be more than one map for the Cartesian product signature. Raster maps must be categorical and its size must be greater than the scene size specified by the user. Size defines the size of individual scene for which the histogram is calculated. Shift shows how scenes shift along the grid in n-s and w-e directions. Shift defines the resolution of the new grid. If window size

is bigger than shift (recommended), motifels will overlap. If shift and size is equal, windows will not overlap. Shift cannot be greater than size.

The output grid may be an input to one of the following GeoPAT modules: `gpat_search`, `gpat_compare`, `gpat_segment`, `gpat_segquality`, `gpat_grd2txt`, and `gpat_globnorm`.

2.2.1.2 gpat_gridts

Creates a grid of time series signatures from raster maps.

Usage:

```
gpat_gridts [−nh] −i <file_name> [−i <file_name>]... −o <file_name> [−d <n>]

−i, −−input=<file_name> name of input file(s) (GeoTIFF)
−o, −−output=<file_name> name of output file (GRID)
−d, −−dimension=<n> dimension of vector that describes time series element (default: 1)
−n, −−normalize normalize each vector coordinate to [0.0, 1.0] (default: no)
−h, −−help print this help and exit
```

input

Defines raster layers that will be used as a source for building time series grid. Number of input layers has to be equal to the time series length times the size of time series element. In order to provide input maps, type multiple input options ("-i map1.tif -i map2.tif" or "–input=map1.tif –input=map2.tif"). The order of the input layers is important. Following time series has length of six and each element contains three attributes:

$$ts = \begin{bmatrix} attr_{1,1}, & attr_{2,1}, & attr_{3,1} \\ attr_{1,2}, & attr_{2,2}, & attr_{3,2} \\ attr_{1,3}, & attr_{2,3}, & attr_{3,3} \\ attr_{1,4}, & attr_{2,4}, & attr_{3,4} \\ attr_{1,5}, & attr_{2,5}, & attr_{3,5} \\ attr_{1,6}, & attr_{2,6}, & attr_{3,6} \end{bmatrix} \quad (1)$$

For such time series input, GeoTiff (each GeoTiff contains one attribute) files should be in following order: $attr_{1,1}$, $attr_{2,1}$, $attr_{3,1}$, $attr_{1,2}$, $attr_{2,2}$, $attr_{3,2}$, ... $attr_{1,5}$, $attr_{2,5}$, $attr_{3,5}$, $attr_{1,6}$, $attr_{2,6}$, $attr_{3,6}$.

In this case a call of `gpat_gridts` should look like:

```
gpat_gridts −i attr11.tif −i attr21.tif −i attr31.tif −i attr12.tif, ... −o new_grid
```

output

Output consists of two files: one of them is a dataset containing the grid of time series in the binary form, the other one is a header text file (the .hdr extension) containing the grid topology and the information about the input data parameters. Modifying the header is strongly discouraged as it may cause some calculations to fail. Structure of the header is as follows:

- dim – number of dimensions (for time series it is 2)

- dims – size of each dimension (two numbers: size of time series element, length of time series)
- type – type of data stored in the grid (integer, float, etc.)
- at0 – top left x
- at1 – w-e grid resolution
- at2 – rotation (0 if grid is north-up)
- at3 – top left y
- at4 – rotation (0 if grid is north-up)
- at5 – n-s grid resolution
- rows – number of rows in the grid
- cols – number of cols in the grid
- proj – projection in the wkt style
- desc – command used to create the grid

dimension

The dimension of a time series element. This number describes length of vectors that builds a time series.

normalize

An option to normalize each attribute to $[0.0, 1.0]$ globally.

Description:

This module creates a grid of time series (grid of time pattern signatures). The output is a grid of the same spatial extent as the input raster map, with the same cell size. Time series are stored as numerical vectors in the binary form. The module outputs a header file (.hdr) containing a topology of the grid and a binary file containing the time series ordered by rows.

This module uses raster data in the GeoTIFF format as an input. Number of input files has to be equal to number of the time series element dimension.

The output grid may be an input to one of the following GeoPAT modules: `gpat_pointsts`, `gpat_search`, `gpat_compare`, `gpat_segment`, `gpat_segquality`, `gpat_grd2txt`, and `gpat_globnorm`.

2.2.1.3 gpat_pointshis

Calculates numerical signatures of individual motifels within a raster map.

Usage:

```
gpat_pointshis [−lah] −i <file_name> −o <file_name> [−s <signature_name>] [−−level=<n>] [−z <n>]
    [−n <normalization_name>] [−x <double>] [−y <double>] [−d <string>] [−−xy_file=<file_name>]

−i, −−input=<file_name> name of input file (GeoTIFF)
−o, −−output=<file_name> name of output file (TXT)
−s, −−signature=<name> motifel's signature (use −l to list all signatures, default: 'cooc')
−−level=<n> full decomposition level (default: 0, auto)
−z, −−size=<n> motifel size in cells (default: 150)
−n, −−normalization=<name> signature normalization method (use −l to list all methods, default: 'pdf')
−l list all signatures and normalization methods
−x <double> x coord
−y <double> y coord
−d, −−description=<string> description of the location
−−xy_file=<file_name> name of file with coordinates (TXT)
−a, −−append append results to output file
−h, −−help print this help and exit
```

Options:

input

Defines categorical raster map(s) in the GeoTIFF format, which will be used as a source for extracting pattern signature. For the Cartesian product method ('prod') there can be more than one input map. Other methods use one map only (the first one provided). In order to provide more than one input map, type multiple input options ("-i map1.tif -i map2.tif" or "–input=map1.tif –input=map2.tif").

output

Name of output text file containing signatures. In the output file, each line corresponds to a single motifel's signature. Each signature is preceded by its header. A header always consists of: coordinates of the midpoint of motifel (scene), name, number of dimensions, and length of each dimension. Modifying the signature file is strongly discouraged as it may cause some calculations to fail.

signature

Defines method for calculating signatures of motifels. GeoPAT offers the following methods:

- prod – Cartesian product of input category lists
- cooc – Spatial coocurrence of categories
- sdec – Simple 2-level decomposition
- fdec – Full decomposition
- lind – Landscape indices vector

- linds – Selected landscape indices vector
- ent – Shannon entropy

See Appendix B for details.

level

This option is used only when 'full decomposition' (fdec) is used. It defines the highest level of decomposition. See Appendix B for the details.

size

Size of a motifel (scene) for which signature is calculated. Scene is a square centered at the coordinate pair location, while its extent is defined by size.

normalization

Specifies normalization method used on the local signatures. See Appendix C for the details. Normalization method used should depend on a selected signature method. For example, the default pdf normalization works well for the histogram-based signatures (pred, cooc, sdec, fdec), while normalization should be disabled (none) for the landscape indices signatures (lind, linds) and the Shannon entropy signature.

x

X coordinate of the midpoint of motifel for which signature will be calculated.

y

Y coordinate of the midpoint of motifel for which signature will be calculated.

description

Description of a motifel. It can be a name of location, description of pattern, or anything else. If not provided, the default description is "location".

xy_file

Name of a text file with list of coordinates. This option is useful for calculating signatures for multiple motifels in a single run.

append (flag)

Append mode. Useful when using the coordinate pair mode and when scenes are processed one by one (see description below). If the append flag is used and the output file already exists, signatures will be appended at the end of file instead of overwriting it.

Description:

This module extracts signatures for a scene (motifel) or collection of scenes defined over square-shaped windows. User provides coordinates of the center of each scene and the size of the scene. The module outputs a list of scene-labeled signatures. As an input the module uses categorical raster map in the GeoTIFF format (or a few maps if user calculates the

Cartesian product of multiple maps), coordinates of the center of each scene and size of scenes. The coordinates must be in the input map's coordinate system. There are two ways of defining the scenes:

Definition by coordinate pair: This is the simplest mode designed for a batch or interactive processing. In this mode, scene is defined by a pair of its midpoint's coordinates (the *x* and *y* options) and scene size parameter (the *size* option). Only one scene can be processed at once. The *xy_file* option is not used in this mode. If used with the append flag (-a), signatures can be calculated one by one and added to the same output file. Additionally, the *description* option lets user name a scene. This name will be stored in the output file.

Definition by text file: In this mode, scene midpoint's coordinates are defined in a external text file in which each line contains X,Y coordinates using the following syntax: x_coordinate,y_coordinate. The name of a file with coordinates is provided using the *xy_file* option. Extent of a scene is defined by the *size* parameter. The *x, y* and *description* options are not used in this mode. Example of the content of coordinates file:

```
1260500, 1277638
1289493, 1251381
1304934, 1285000
1277700, 1261936
```

The output signature text file can be used as an input to the following modules: `gpat_search`, `gpat_distmtx`.

2.2.1.4 gpat_pointsts

Calculates numerical signatures of an individual cell within a time series data.

Usage:

```
gpat_pointsts [−ah] −i <file_name> −o <file_name> [−x <double>] [−y <double>] [−d <string>]
    [−−xy_file=<file_name>]

−i, −−input=<file_name> name of input file (GRID)
−o, −−output=<file_name> name of output file (TXT)
−x <double> x coord
−y <double> y coord
−d, −−description=<string> description of the location
−−xy_file=<file_name> name of file with coordinates (TXT)
−a, −−append append results to output file
−h, −−help print this help and exit
```

input

A grid file created by gpat_gridts is an input for gpat_pointsts.

output

Name of the output text file containing signatures. In the output file, each line corresponds to a single motifel's signature. Each signature is preceded by its header. Modifying the signature file is strongly discouraged as it may cause some calculations to fail.

x

X coordinate of the midpoint of motifel for which signature will be calculated.

y
Y coordinate of the midpoint of motifel for which signature will be calculated.

description
Description of a motifel. It can be a name of location, description of pattern, etc. If not provided, the default description is "location".

xy_file
Name of a text file with list of coordinates. This option is useful for calculating signatures for multiple motifels in a single run.

append (flag)
Append mode. Useful when using the coordinate pair mode and when scenes are processed one by one (see the description below). If the append flag is used and the output file already exists, signatures will be appended at the end of file instead of overwriting it.

Description:
This module extracts time series for a given location or collection of locations. User provides coordinates of each location and optionally description of location. The coordinates must be in the input grid's coordinate system. Coordinates can be defined in the same way as in gpat_pointshis.

The output text file can be used as an input to the following modules: `gpat_search`, `gpat_distmtx`.

2.2.1.5 gpat_polygon
Calculates numerical signatures of irregular regions.

Usage:

```
gpat_polygon [−lh] −i <file_name> −e <file_name> −o <file_name> [−s <signature_name>] [−n
    <normalization_name>] [−t <n>]

−i, −−input=<file_name> name of input file (GeoTIFF)
−e, −−segments=<file_name> name of input file (GeoTIFF, int)
−o, −−output=<file_name> name of output file (TXT)
−s, −−signature=<name> signature method (use −l to list all methods, default: 'cooc')
−n, −−normalization=<name> signature normalization method (use −l to list all methods, default: 'pdf')
−l list all signatures and normalization methods
−t <n> number of threads (default: 1)
−h, −−help print this help and exit
```

Options:

input
Categorical raster map(s) in the GeoTIFF format, which will be used as a source for extracting pattern signature. For the Cartesian product method ('prod') there can be more

than one input map. Other methods only use one map (the first one provided). In order to provide more than one input map, type multiple input options ("-i map1.tif -i map2.tif" or "–input=map1.tif –input=map2.tif").

segments

Categorical raster map in the GeoTIFF format which defines spatial division of area of interest into polygonal regions. In this map, regions should be defined as areas of unique ids. Layer must be a raster and its projection, extents and resolution should be identical to the main input map. This layer may be a result of the segmentation module `gpat_segment` or any other categorical map (e.g. map of ids of watersheds).

output

Name of output text file containing signatures. In the output file each line corresponds to a region's signature. Each signature is preceded by its header. A header always consists of: the coordinates of the midpoint of region, name, number of dimensions, and length of each dimension. Modifying the signature file is strongly discouraged as it may cause some calculations to fail.

signature

Defines method for calculating signatures of motifels. GeoPAT offers the following methods:

- prod – Cartesian product of input category lists
- cooc – Spatial coocurrence of categories
- sdec – Simple 2-level decomposition
- fdec – Full decomposition
- lind – Landscape indices vector
- linds – Selected landscape indices vector
- ent – Shannon entropy

See Appendix B for the details.

normalization

Specifies normalization method used on the local signatures. See Appendix C for the details. Normalization method used should depend on a selected signature method. For example, the default pdf normalization works well for the histogram-based signatures (pred, cooc, sdec, fdec), while normalization should be disabled (none) for the landscape indices signatures (lind, linds) and the Shannon entropy signature.

threads (-t)

The module is parallel, therefore use more than one processing thread can be used in order to speed up calculations. This option specifies how many threads will be used. The

default is 1.

Description:

Module extracts signatures for a collection of irregular regions using the same methods as in the `gpat_pointshis` and `gpat_gridhis` modules. As an input, apart from categorical raster map from which signatures are calculated, user provides a categorical raster map which defines spatial division of area of interest into polygonal regions. The module outputs a list of polygon labeled-signatures stored in the form of a text file.

The output signature text file can be used as an input to the following modules: `gpat_search`, `gpat_distmtx`.

2.2.2 Similarity measuring

2.2.2.1 gpat_search

Produces similarity maps which show similarity value between query motifels (scenes) and every motifel in the input grid.

Usage:

```
gpat_search [−dlh] −i <file_name> [−o <file_name>] −r <file_name> [−m <measure_name>]
    [−−type=Byte/....] [−p <file_name>] [−n <n>] [−t <n>]

−i, −−input=<file_name> name of input file (GRID)
−o, −−output=<file_name> name of output file (TIFF)
−r, −−reference=<file_name> reference data to calculate similarity (TXT)
−m, −−measure=<measure_name> similarity measure (use −l to list all measures; default 'jsd')
−l, −−list_measures list all measures
−−type=Byte/.... output data type (default: Float64)
−p, −−palette=<file_name> name of the file with colors definition (CSV)
−n, −−no_data=<n> output NO DATA value (default: none)
−t <n> number of threads (default: 1)
−h, −−help print this help and exit
```

input

Binary file containing grid of signatures (grid-of-scenes). Grid is mandatory. User has to provide a name of grid file (without the .hdr extension). Header file is read automatically. Grid is an output from either `gpat_gridhis` or `gpat_gridts` module. Header of the grid will define topology of the output layers. Do not modify header file.

output

Raster (GeoTIFF) file or files containing the map of similarity. The number of outputs depends on the number of scenes/polygons in the input reference file.

reference

Text file containing one or more signatures of either individual motifels (scenes) or irregular polygons. This option is mandatory. Input text file can be created using the `gpat_pointshis`, `gpat_pointsts` or `gpat_polygons` module. The number of outputs depends on the number of scenes in the input scene file. At least one scene is required.

measure

Defines method for calculating distance between motifels. GeoPAT offers the following measures:

- jsd – Jensen Shannon Divergence
- euc – Euclidean distance
- eucn – Normalized euclidean distance
- wh – Wave-Hedges distance
- tsEUC – time series - euclidean distance
- tsDTW – time series - Dynamic Time Warping distance
- tsDTWP – periodic time series - Dynamic Time Warping distance
- tsEUCP – periodic time series - euclidean distance
- tsDTWPa – periodic time series - Synchronized Dynamic Time Warping distance

See Appendix D for the details.

list_measures

Lists all the available measures.

type

Data type of the output raster (GeoTIFF) file. Types of Byte, UInt16, Int16, UInt32, Int32, Float32, Float64, CInt16, CInt32, CFloat32 and CFloat64 are supported.

palette

A text file containing colors definition. It enables user to add its own palette to the output raster file. The structure of the text file is as follow: ID of the file (the first row), # of colors max min (the second row), user comment (the third row), value red green blue alpha (the forth row).

For example:

```
PALETTE
16 100 0
my similarity
0 40 54 154 255
7 0 201 50 255
```

no_data

Specify NO DATA value of the output raster (GeoTIFF) file.

threads (-t)

The module is parallel, therefore use more than one processing thread can be used in order to speed up calculations. This option specifies how many threads will be used. The default is 1.

Description:

Searching task is dedicated to perform a query-by-pattern-similarity (QBPS). Searching procedure compares query/queries with the database in the scene-by-scene fashion and creates a layer(s) having the same topology as a grid-of-scenes but containing values of similarity between a query/queries and each scene in a grid-of-scenes. The results of QBPS can be visualized as a similarity map. Searching task is accomplished using `gpat_search` module. An input is a query scene (or a list of query scenes) is an output of `gpat_pointshis` or `gpat_polygon` modules and a grid-of-scenes (database to be queried) outputted by the `gpat_gridhis` module. Signature histogram of each query and signatures of histograms in a grid-of-scenes must be calculated using the same method (e.g. crossproduct must be used in both cases). The number of output map depends on the number of provided query scenes. Each output map gets its name from the name of query scene preceded by a prefix.

2.2.2.2 gpat_compare

Compares two grids-of-scenes (grid of histograms) in a scene-by-scene fashion.

Usage:

```
gpat_compare [−lh] −i <file_name> −i <file_name> −o <file_name> [−−type=Byte/....] [−p
    <file_name>] [−n <n>] [−m <measure_name>] [−t <n>]

−i, −−input=<file_name> name of input files (GRID)
−o, −−output=<file_name> name of output file with similarity (TIFF)
−−type=Byte/.... output data type (default: Float64)
−p, −−palette=<file_name> name of the file with colors definition (CSV)
−n, −−no_data=<n> output NO DATA value (default: none)
−m, −−measure=<measure_name> similarity measure (use −l to list all measures; default 'jsd')
−l, −−list_measures list all measures
−t <n> number of threads (default: 1)
−h, −−help print this help and exit
```

input

Two binary files containing grid of signatures (grid-of-scenes). User has to provide names of grid files (without the .hdr extension). Grid is an output from either `gpat_gridhis` or `gpat_gridts` module. Headers of the grids will define topology of the output layers. Do not modify header files.

output

Raster (GeoTIFF) file containing the map of similarity between two input grid of signatures.

type

Data type of the output raster (GeoTIFF) file. Types of Byte, UInt16, Int16, UInt32, Int32, Float32, Float64, CInt16, CInt32, CFloat32 and CFloat64 are supported.

palette

A text file containing colors definition. It enables user to add its own palette to the output raster file. The structure of the text file is as follow: ID of the file (the first row), # of colors max min (the second row), user comment (the third row), value red green blue alpha (the forth row).

For example:

```
PALETTE
16 100 0
my similarity
0 40 54 154 255
7 0 201 50 255
```

no_data

Specify NO DATA value of the output raster (GeoTIFF) file.

measure

Defines method for calculating distance between motifels. GeoPAT offers the following measures:

- jsd – Jensen Shannon Divergence
- euc – Euclidean distance
- eucn – Normalized euclidean distance
- wh – Wave-Hedges distance
- tsEUC – time series - euclidean distance
- tsDTW – time series - Dynamic Time Warping distance
- tsEUCP – periodic time series - euclidean distance
- tsDTWP – periodic time series - Dynamic Time Warping distance
- tsDTWPa – periodic time series - Synchronized Dynamic Time Warping distance

See Appendix D for the details.

list_measures

Lists all the available measures.

threads (-t)

The module is parallel, therefore use more than one processing thread can be used in order to speed up calculations. This option specifies how many threads will be used. The default is 1.

Description:

This module compares two grids-of-scenes (grid of histograms) in a scene-by-scene fashion. Both grids must be calculated using the same shift parameter and the same method. The output of `gpat_compare` is a raster having the same topology as the input grids. Each cell in the output file contains the value of similarity between corresponding pairs of scenes where one scene is coming from the first grid and the other scene is coming from the second grid. The module is useful for comparing data of the same area created at two different times. It is also useful for estimating the uncertainty of the results.

2.2.2.3 gpat_segment

Segments a grid-of-scenes into regions of uniform patterns.

Usage:

```
gpat_segment [−lcgaqh] −i <file_name> −o <file_name> [−v <file_name>] [−−size=<n>] [−m
    <measure_name>] [−−lthreshold=<double>] [−−uthreshold=<double>] [−−swap=<double>]
    [−−minarea=<n>] [−−maxhist=<n>] [−t <n>]

−i, −−input=<file_name> name of input files (GRID)
−o, −−output=<file_name> name of output file with segments (TIFF)
−v, −−vector=<file_name> name of output vector file with segments (SHP)
−−size=<n> output resolution modyfier (default: 1)
−m, −−measure=<name> similarity measure (use −l to list all measures; defult: jsd)
−l, −−list_measures list all measures
−−lthreshold=<double> minimum distance threshold to build areas (default: 0.1)
−−uthreshold=<double> maximum distance threshold to build areas (default: 0.3)
−−swap=<double> improve segmentation by swapping unmatched areas. −1 to skip (default: 0.001)
−−minarea=<n> minimum number of motifels in individual segment (default: 0)
−−maxhist=<n> create similarity/distance matrix for maxhist histograms; leave 0 to use all (default:
    200)
−c, −−complete use complete linkage (default is average)
−g, −−no_growing skip growing phase
−a, −−no_hierarhical skip hierarchical phase
−q, −−quad quad mode (rook topology)
−t <n> number of threads (default: 1)
−h, −−help print help and exit
```

input

Binary file containing grid of signatures (grid-of-scenes). Grid is mandatory. User has to provide a name of grid file (without the .hdr extension). Header file is read automatically. Grid is an output from either `gpat_gridhis` or `gpat_gridts` module. Header of the grid will define topology of the output layers. Do not modify header file.

output

Categorical raster map in the GeoTIFF format which defines spatial division of area of interest into polygonal regions (segments). In this map, regions (segments) are defined as areas of unique ids.

vector

Vector (shapefile) map of a segmentation. This data is automatically created by vector-

ization of the output file.

size

As the default, the resolution of the output map equals to the resolution of the input grid. The size option modifies the output resolution. For example, when the size is set to 2, the output map resolution would be increased by a factor of 2.

measure

Defines method for calculating distance between motifels. GeoPAT offers the following measures:

- jsd – Jensen Shannon Divergence
- euc – Euclidean distance
- eucn – Normalized euclidean distance
- wh – Wave-Hedges distance
- tsEUC – time series - euclidean distance
- tsDTW – time series - Dynamic Time Warping distance
- tsEUCP – periodic time series - euclidean distance
- tsDTWP – periodic time series - Dynamic Time Warping distance
- tsDTWPa – periodic time series - Synchronized Dynamic Time Warping distance

See Appendix D for the details.

list_measures

Lists all the available measures.

lthreshold

The purpose of lthreshold is to control (to some degree) sizes of the segments. It takes values between 0 and 1.

uthreshold

The purpose of uthreshold is to prevent the growth of inhomogeneous segments. It takes values between 0 and 1.

swap

Because the segment growing is a greedy process there is a chance that some motifels located at segments' boundaries show more affinity to motifels across the boundary than to motifels in its own segment. The swap parameter is used to alleviate this problem. It adjusts the boundaries between the segments. For each border motifel its linkage to the parent segment, D_{par}, as well as its linkage to the segment across the boundary, D_{neigh} is

calculated. If the difference between D_{par} and D_{neigh} is positive (the linkage to the segment across the boundary is smaller than the linkage to parent segment) the motifel is marked for possible reassignment (resulting in an adjustment to the boundary). The boundary between segments is adjusted only when the swap value is larger than the difference between D_{par} and D_{neigh}. The value of 0 forces repetition till all unmatched segments will be swapped (note: this could distinctly increase the calculation time).

minarea

Minimum number of cells in an individual segment not subject to removal process. The removal process is turned off if the minarea value equals to 0.

maxhist

Calculating linkage between two segments could be computationally expensive if the segments contain large numbers of motifels. When the number of motifels in the segment exceeds the maxhist value, so-called leveraging is used to speed up the calculation. This means that only a maxhist number of motifels, randomly selected from the segment, are used to calculate the linkage. 0 switches off the leveraging.

complete

As the default, the average linkage (Unweighted Pair Group Method with Arithmetic Mean) is used for calculation of a dissimilarity between two segments. This option changes the measure of dissimilarity between two segments to the complete linkage.

no_growing

Skips growing phase of the segmentation.

no_hierarchical

Skips hierarchical phase of the segmentation.

quad

As the default, the 6-connected brick wall topology is used, where square motifels are "laid" in alternative layers with each layer shifted a half motifel length with respect to the previous one like in masonry. This option sets quad mode, which means that the rectangular grid topology with 4-connectivity will be used. Important note: the rectangular grid topology performs calculations on an input grid of signatures. However, as a default, neighboring signatures are merged to create the brick wall topology. See Appendix E for the details.

threads (-t)

The module is parallel, therefore use more than one processing thread can be used in order to speed up calculations. This option specifies how many threads will be used. The default is 1.

Description:

This module segments a grid-of-scenes into regions of uniform patterns in the same fashion as traditional segmentation algorithms segment image (or other rasters) into segments of

uniform color and texture ([1]). The difference is that `gpat_segment` segments a grid-of-scenes rather than an ordinary grid, and uses scene signature as attribute rather than color or image texture to decide how to delineate the segments. The module uses a variant of the region growing algorithm ([2]–[4]). The segmentation provided by the `gpat_segment` module is typically used as an intermediate step in regionalization of the dataset. The goal of regionalization is to generalize and thus simplify spatial representation of data so it is more meaningful and easier to analyze. Examples of regionalization include delineation of landscape types ([1]) or delineation of physiographic units ([5]). Regionalization is achieved by clustering segments output by `gpat_segment` into a number of distinct pattern type classes. An output of the `gpat_distmtx` module could be used as a distance matrix for clustering in an external software.

2.2.2.4 gpat_distmtx

Computes a distance matrix between a collection of scenes.

Usage:

```
gpat_distmtx [−lsh] −i <file_name> −o <file_name> [−m <measure_name>]

−i, −−input=<file_name> name of input file with signatures (TXT)
−o, −−output=<file_name> name of output file (CSV) with similarity matrix
−m, −−measure=<name> similarity measure (use −l to list all measures; default 'jsd')
−l, −−list_measures list all measures
−s, −−similarity output is a similarity matrix
−h, −−help print this help and exit
```

input

Text file with signatures created using either `gpat_pointshis`, `gpat_pointsts`, `gpat_grid2txt`, or `gpat_polygon` module.

output

Text file which contain a distance matrix. The first row and first column contains names of objects. The rest of the file has values of distance for the pairs of objects.

measure

Defines method for calculating distance between motifels. GeoPAT offers the following measures:

- jsd – Jensen Shannon Divergence
- euc – Euclidean distance
- eucn – Normalized euclidean distance
- wh – Wave-Hedges distance
- tsEUC – time series - euclidean distance
- tsDTW – time series - Dynamic Time Warping distance

- tsEUCP – periodic time series - euclidean distance
- tsDTWP – periodic time series - Dynamic Time Warping distance
- tsDTWPa – periodic time series - Synchronized Dynamic Time Warping distance

See Appendix D for the details.

list_measures

Lists all the available measures.

similarity

Description:

This module computes a distance matrix between a collection of scenes. It uses signatures output by `gpat_pointshis`, `gpat_pointsts`, or `gpat_polygon` and selected distance measure. The resultant distance matrix is usually used as an input for scene clustering, which could results in discovering structures in the data. Clustering itself is not implemented in GeoPAT 2.0; we recommend using clustering algorithms implemented in R ([6]). An example of `gpat_distmtx` usage would be the clustering of a collection of cities on the basis of the patterns of land cover classes within their boundaries. See section 3.4 for the details.

2.2.3 Tools

2.2.3.1 gpat_grd2txt

Converts a binary file containing grid of signatures (grid-of-scenes) into a text file.

Usage:

```
gpat_grd2txt [−h] −i <file_name> −o <file_name>

−i, −−input=<file_name> name of input file (GRID)
−o, −−output=<file_name> name of output file (TXT)
−h, −−help print this help and exit
```

input

Binary file containing grid of signatures (grid-of-scenes). Grid is mandatory. User has to provide a name of grid file (without the .hdr extension). Header file is read automatically. Grid is an output from either `gpat_gridhis` or `gpat_gridts` module. Header of the grid will define topology of the output layers. Do not modify header file.

output

Text file with a grid of signatures. It can be an input for the `gpat_distmtx` module.

Description:

This module converts a binary file containing grid of signatures (grid-of-scenes) into a text file.

2.2.3.2 gpat_globnorm

Normalize a grid of signatures globally.

Usage:

```
gpat_globnorm [−lh] −i <file_name> −o <file_name> [−m <method_name>] [−t <n>]

−i, −−input=<file_name> name of input file (GRID)
−o, −−output=<file_name> name of output file (GRID)
−m, −−method=<method_name> normalization method (use −l to list all methods, default: '01')
−l, −−list_methods list all methods
−t <n> number of threads (default: 1)
−h, −−help print this help and exit
```

input

Binary file containing grid of signatures (grid-of-scenes). Grid is mandatory. User has to provide a name of grid file (without the .hdr extension). Header file is read automatically. Grid is an output from either `gpat_gridhis` or `gpat_gridts` module. Header of the grid will define topology of the output layers. Do not modify header file.

output

Normalized binary file containing grid of signatures (grid-of-scenes) and the header text file (the .hdr extension) containing a grid topology and an information about the input data parameters.

method

Specifies normalization method used on the whole grid of signatures.

- 01 – normalize coordinates to [0,1]
- N01 – normalize coordinates to N(0,1)
- ind01 – normalize coordinates to [0,1] for 72 landscape indices

See Appendix C for the details.

list_methods

Lists all the available normalization methods.

threads (-t)

The module is parallel, therefore use more than one processing thread can be used in order to speed up calculations. This option specifies how many threads will be used. The default is 1.

Description:

This module allows for a global normalization of a grid of signatures. It could be useful when pattern signatures should have the same scale of values before searching, comparison, segmentation, or clustering.

2.2.3.3 gpat_segquality

Computes quality metrics of a segmentation

Usage:

```
gpat_segquality [−lcqwh] −i <file_name> −s <file name> [−g <file_name>] [−o <file_name>] [−m
    <measure_name>] [−−maxhist=<n>] [−t <n>]

−i, −−input=<file_name> name of input file (GRID)
−s, −−segments=<file name> name of input segmentation map (TIFF)
−g, −−inhomogeneity=<file_name> name of output file with segment inhomogeneity (TIF
F)
−o, −−isolation=<file_name> name of output file with segment isolation (TIFF)
−m, −−measure=<name> similarity measure (use −l to list all measures; defult: jsd)
−l, −−list_measures list all measures
−−maxhist=<n> create similarity/distance matrix for maxhist histograms; leave 0 to use all (default:
    200)
−c, −−complete use complete linkage (default is average)
−q, −−quad quad mode (rook topology)
−w, −−no_weight switch off edge−based weighting in isolation
−t <n> number of threads (default: 1)
−h, −−help print help and exit
```

input

Binary file containing grid of signatures (grid-of-scenes). Grid is mandatory. User has to provide a name of grid file (without the .hdr extension). Header file is read automatically. Grid is an output from either `gpat_gridhis` or `gpat_gridts` module. Header of the grid will define topology of the output layers. Do not modify header file.

segments

Categorical raster map in the GeoTIFF format which defines spatial division of area of interest into polygonal regions. In this map, regions should be defined as areas of unique ids. Layer must be a raster and its projection, extents and resolution should be identical to the main input map. This layer may be a result of the segmentation module gpat segment or any other categorical map (e.g. map of ids of watersheds).

inhomogeneity

The name of a raster (GeoTIFF) file containing the values of segment's inhomogeneity. Inhomogeneity is a property of a single segment; it measures the degree of mutual dissimilarity between all motifels within the segment. Inhomogeneity has a range between 0 and 1, smaller values are better.

isolation

The name of a raster (GeoTIFF) file containing the values of segment's isolation. It is calculated as the average dissimilarity between the focus segment and all of its immediate neighbors. It measures how much the segments stand out from its surroundings. Isolation has a range between 0 and 1, larger values are better

measure

Defines method for calculating distance between motifels. GeoPAT offers the following measures:

- jsd – Jensen Shannon Divergence
- euc – Euclidean distance
- eucn – Normalized euclidean distance
- wh – Wave-Hedges distance
- tsEUC – time series - euclidean distance
- tsDTW – time series - Dynamic Time Warping distance
- tsEUCP – periodic time series - euclidean distance
- tsDTWP – periodic time series - Dynamic Time Warping distance
- tsDTWPa – periodic time series - Synchronized Dynamic Time Warping distance

See Appendix D for the details.

list_measures

Lists all the available measures.

maxhist

Calculating linkage between two segments could be computationally expensive if the segments contain large numbers of motifels. When the number of motifels in the segment exceeds the maxhist value, so-called leveraging is used to speed up the calculation. This means that only maxhist motifels, randomly selected from the segment, are used to calculate the linkage.

complete

As the default, the average linkage (Unweighted Pair Group Method with Arithmetic Mean) is used for calculation of a dissimilarity between two segments. This option changes the measure of dissimilarity between two segments to the complete linkage.

quad

As the default, the 6-connected brick wall topology is used, where square motifels are "laid" in alternative layers with each layer shifted a half motifel length with respect to the previous one like in masonry. This option sets quad mode, which means that the rectangular grid topology with 4-connectivity will be used.

no_weight

As the default, weights related to the distance between adjacent segments are applied. This option switch off edge-based weighting in the calculations of isolation.

threads (-t)

The module is parallel, therefore use more than one processing thread can be used in order to speed up calculations. This option specifies how many threads will be used. The default is 1.

Description:

This module computes inhomogeneity and isolation of grid of signatures (grid-of-scenes for each segment (region). It uses signatures output by `gpat_gridhis` and selected distance measure. The resultant raster files could be also used to calculate the quality of a segmentation or regionalization. Quality is equal to 1 - (inhomogeneity/isolation). It has a range between 0 and 1 and larger numbers are better.

3 Basic workflow paths with examples

In this section, the basic GeoPAT procedures are presented:

- **Search** - search for areas similar to a query
- **Change detection** - comparison of local patterns between two maps
- **Segmentation** - division of a map into regions of cohesive patterns
- **Clustering** - grouping patterns that are similar to each other

The procedures are explained using several workflow schemes and examples. The examples how to process categorical data are shown on a 42×69 km (1400×2300 px) part of the National Land Cover Dataset (NLCD) covering area around the city of Augusta, GA (Figure 2). This area is characterized by a high diversity of land cover categories and patterns. Thus it is perfect for various pattern analyzes.

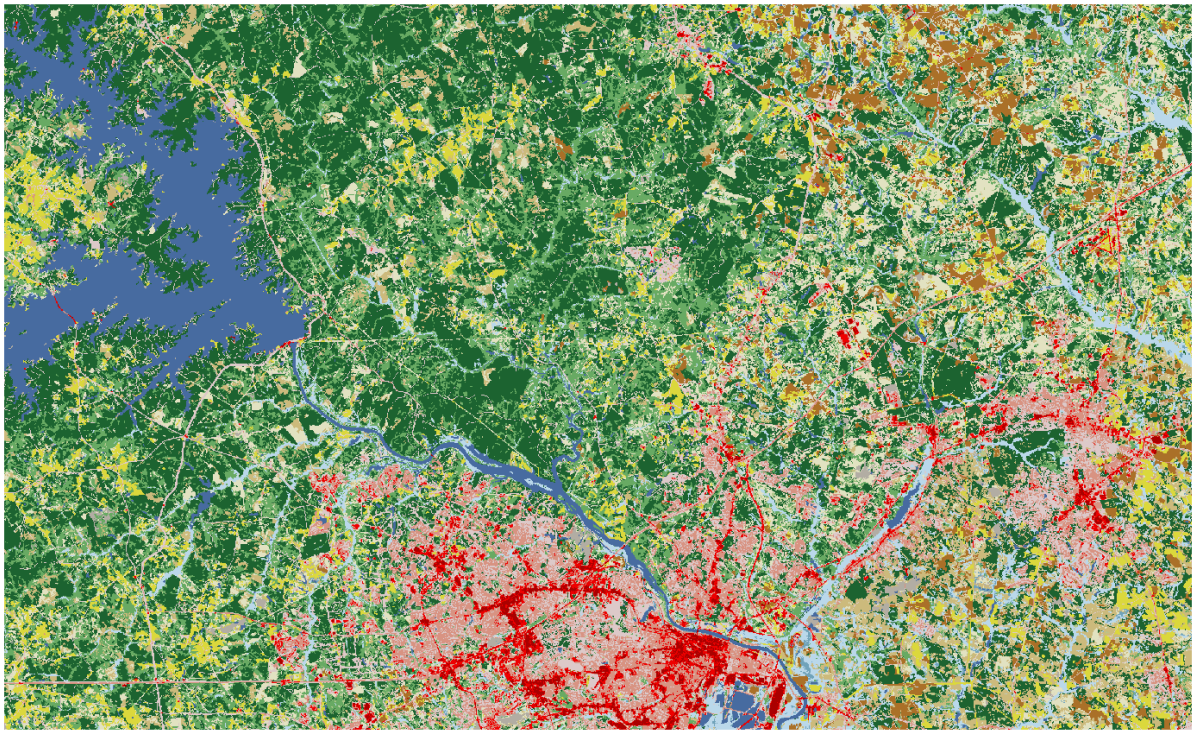

Figure 2: Part of NLCD, covering area around Augusta, GA used in the examples

Examples of time-series analysis are based on monthly sums of precipitation for area of Great Britain from the WorldClim database ([7]; Figure 3). It consists of twelve raster files, where each file has 240 rows and 319 columns. Precipitation values are expressed in millimeters.

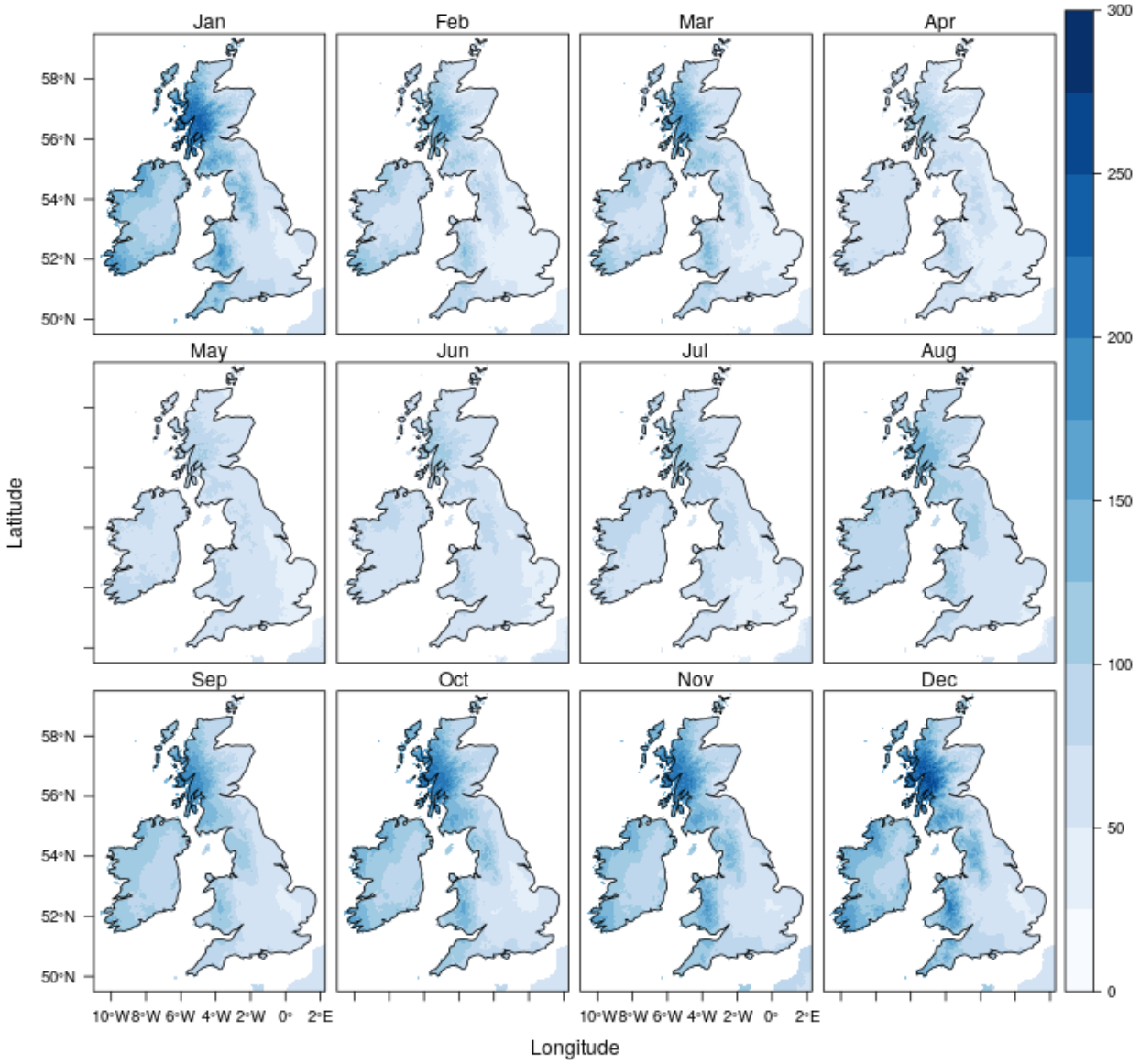

Figure 3: Monthly sums of precipitation for area of Great Britain used in the time-series examples

3.1 Search

Search functionality enables users to produce maps of similarity. These maps show the level of similarity between a specified motifel (query) and a grid of motifels. The input is one or more GeoTIFF raster maps (depending on a data and signature type; see appendix B for more information), and XY coordinates of one or more points in space. The result is one or more GeoTIFF raster maps that have the same extent as a grid of motifels specified by user. The number of output maps is the same as the number of points provided. The workflows for categorical and time series maps differ.

3.1.1 Search on categorical maps

Figure 4 presents general workflow path for producing similarity maps using a categorical raster data.

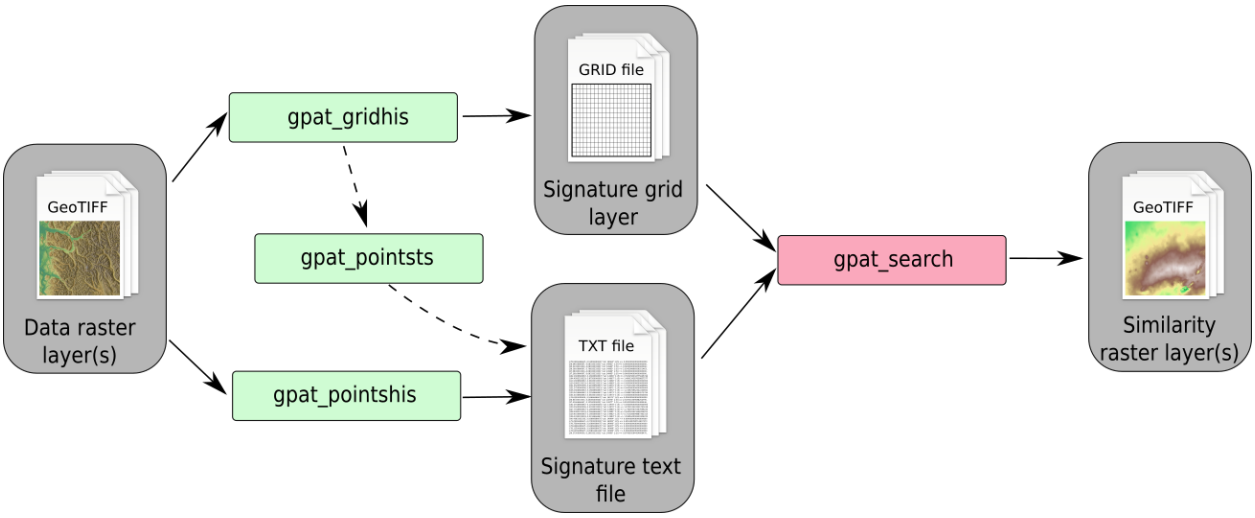

Figure 4: Workflow path for search on categorical maps

The first step is to prepare signature files for both, grid of motifels and query motifels, using two separate modules, `gpat_gridhis` and `gpat_pointshis` respectively. The second step is to use these signature files as inputs to `gpat_search` module in order to the produce similarity maps.

Example:

```
gpat_gridhis -i Augusta2011.tif -o grid -s cooc -z 50 -f 50 -n pdf
gpat_pointshis -i Augusta2011.tif -o query_signatures.txt -s cooc -z 50 -n pdf
      --xy_file=coordinates.txt
gpat_search -i grid -r query_signatures.txt
```

For example, we can search for the most similar motifels to those located in the four points in the Figure 5. Firstly, we need to build a grid of a motifels of a desired signature, size, shift, etc. Secondly, query motifels are extracted using a set of coordinates as an input. Finally, we can obtain the maps of similarity by comparing the grid from the first step with the query motifels from the second step. The four final maps (Figure 6) shows similarity to each of the given points, where the brightest color indicates the most similar areas and the darkest color indicates the least similar areas.

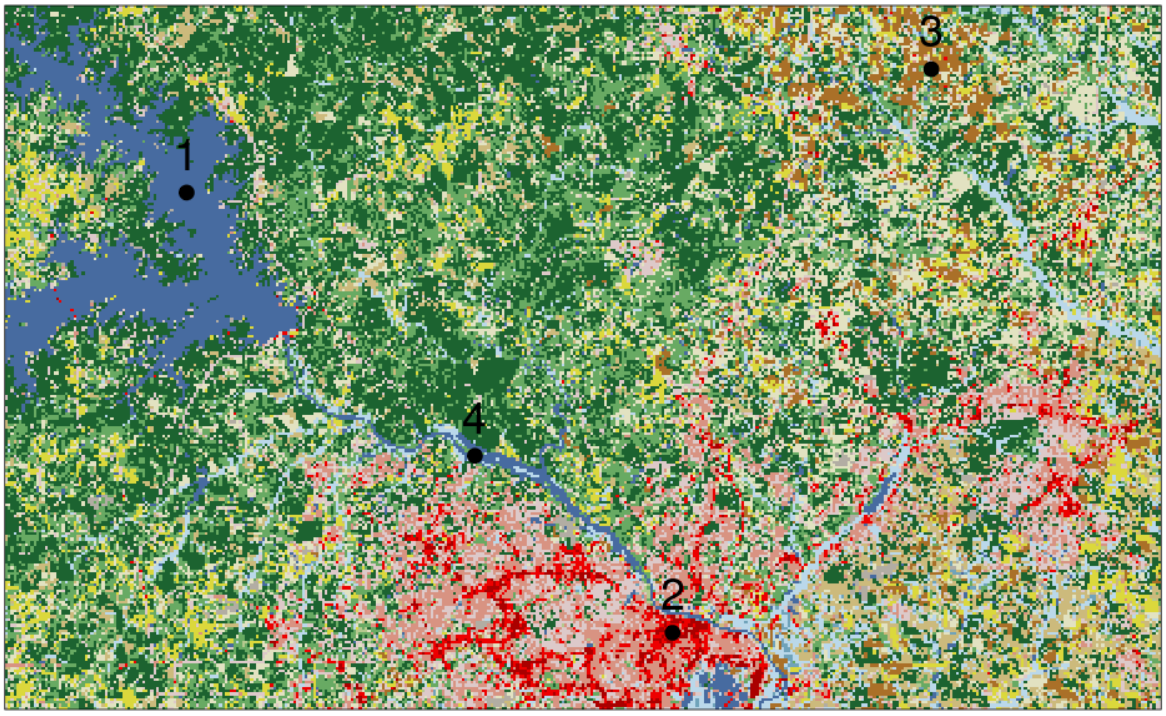

Figure 5: Points of interest on the top of a land cover map

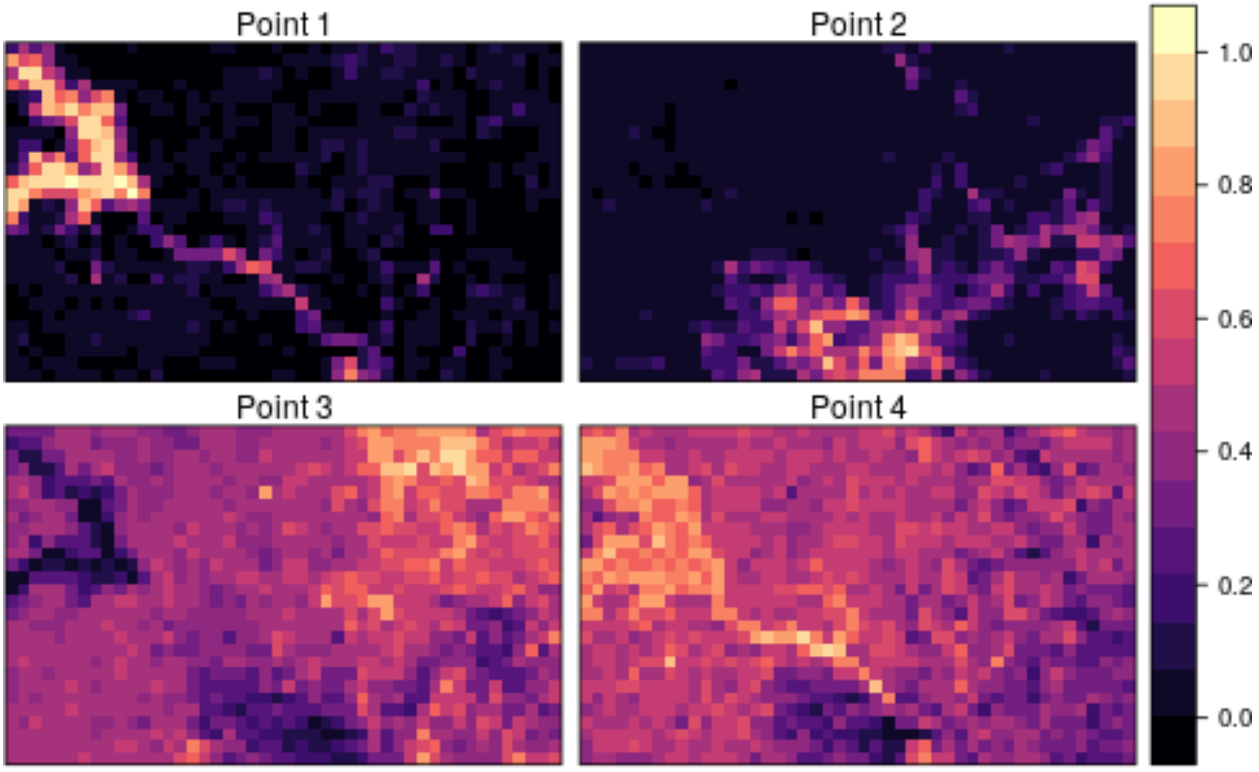

Figure 6: Output similarity maps

3.1.2 Search on time series

Figure 7 presents general workflow path for producing similarity maps using a time-series raster data.

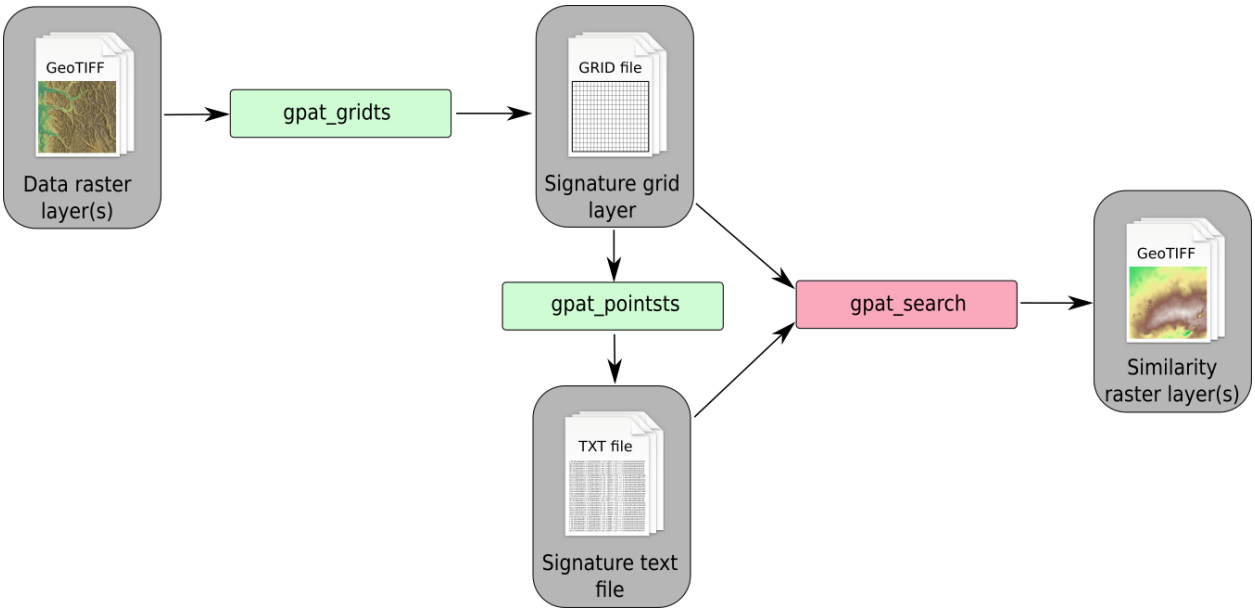

Figure 7: Workflow path for search on time series

The first step is to prepare signature files for both, grid of motifels and query motifels, using two separate modules, `gpat_gridts` and `gpat_pointsts` respectively. The second step is to use these signature files as inputs to `gpat_search` module in order to the produce similarity maps.

Example:

```
gpat_gridts -i GB_pr01.tif -i GB_pr02.tif -i GB_pr03.tif -i GB_pr04.tif -i GB_pr05.tif -i GB_pr06.tif
    -i GB_pr07.tif -i GB_pr08.tif -i GB_pr09.tif -i GB_pr10.tif -i GB_pr11.tif -i GB_pr12.tif -o
    GB_pr_grid -n
gpat_pointsts -i GB_pr_grid -o query_signatures_ts.txt --xy_file=coordinates_gb.txt
gpat_search -i GB_pr_grid -r query_signatures_ts.txt -m tsEUC
```

For example, we want to find out which cities in Great Britain have the most (and the least) similar temporal pattern of precipitation. For this purpose, we need to have maps of precipitation (e.g. twelve rasters with monthly sums of precipitation from January to December) and coordinates of our points of interest (Figure 8).

Figure 8: Location of the points of interest

Firstly, a grid which contains temporal patterns of precipitation need to be built based on the precipitation rasters. Secondly, query signatures are extracted based on the coordinates of points of interest - in our case cities of London, Glasgow, Cardiff, Fort William, and Dublin. Finally, we can create the maps of similarity (one for each city). To do so, we need to specify the similarity measure proper for time series data, such as *tsEUC* (time series - euclidean distance). Our final five maps show where the annual precipitation patterns are the most and the least similar to those in our selected cities (Figure 9).

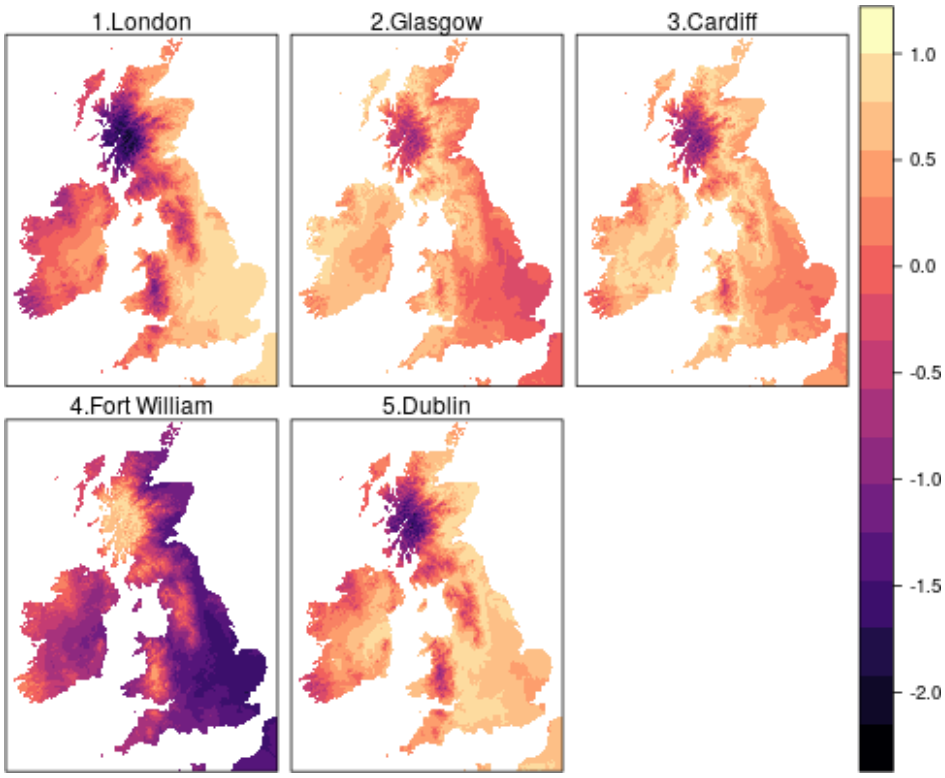

Figure 9: Maps of the similarity in terms of precipitation time-series

3.2 Change detection

Change detection module provides a way to compare patterns between two grids-of-scenes. The output map show the level of similarity between two inputs. Figure 10 presents general workflow path for producing maps of change.

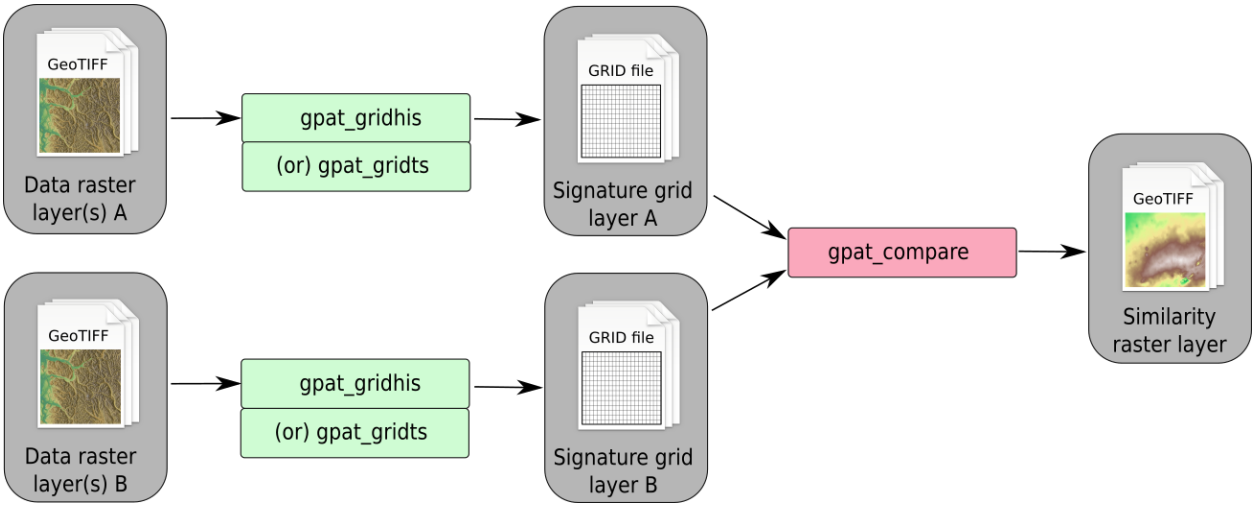

Figure 10: Workflow path for change detection

The first step is to prepare signature files for both dataset, using either `gpat_gridhis` for categorical data or `gpat_gridts` for time series data. The second step is to use these signature files as inputs to the `gpat_compare` module in order to the produce change maps.

Example:

```
gpat_gridhis -i Augusta2006.tif -o Augusta2006_grid100 -z 100 -f 100
gpat_gridhis -i Augusta2011.tif -o Augusta2011_grid100 -z 100 -f 100
gpat_compare -i Augusta2006_grid100 -i Augusta2011_grid100 -o Augusta0611_compared.tif
```

For example, we want to compare a change in land cover patterns between years 2006 and 2011. For this purpose, we can use a land cover data from NLCD for 2006 and 2011 (Figure 11). Both of these datafiles need to have the same extent and resolution.

Firstly, we need to create a grid of signatures for both GeoTIFF files. It is important to remember that both of the created grid of signatures must by created using the same size, shift, and signature.

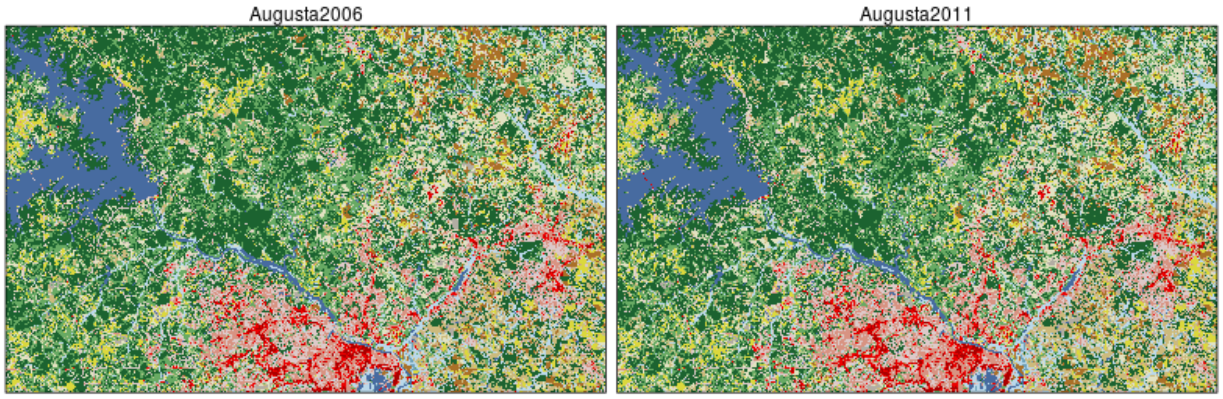

Figure 11: Land cover maps of Augusta for year 2006 and 2011

Secondly, the output map is created based on the grids of signatures. The final map shows where a land cover pattern stayed the same (value of 1) and where it changed. The lowest values indicate the biggest change of a land cover pattern.

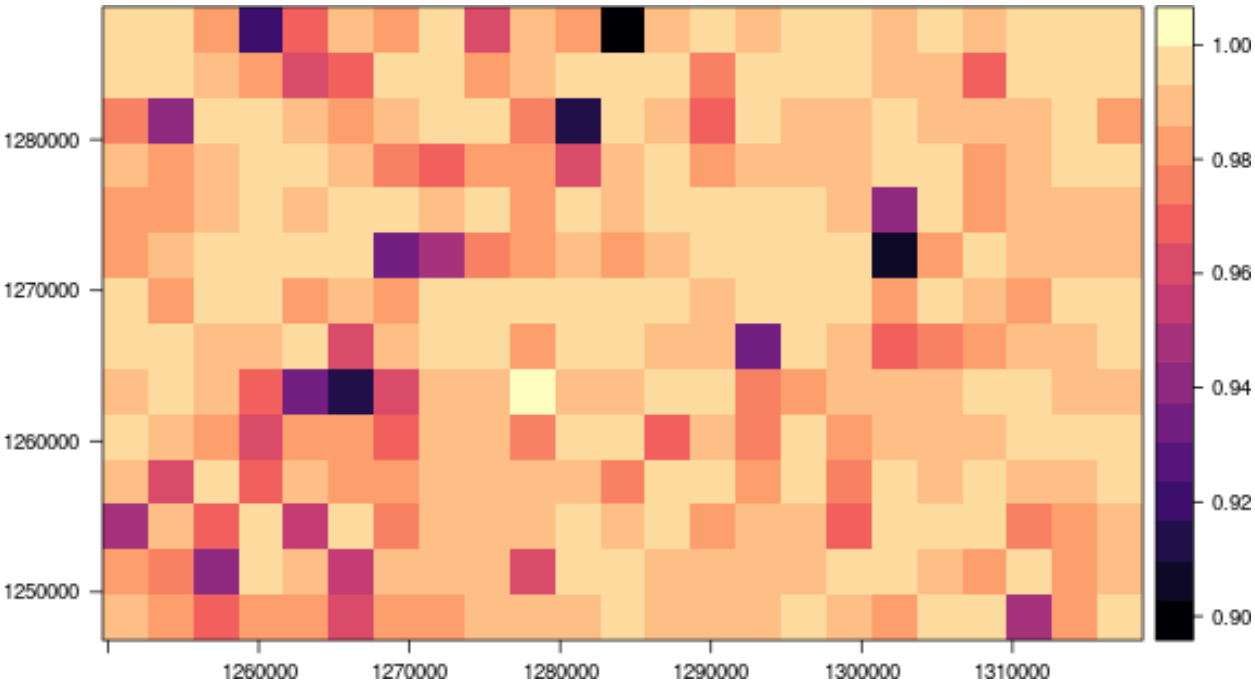

Figure 12: Maps of the similarity between a land cover of Augusta between 2006 and 2011

3.3 Segmentation

Segmentation module enables user to create maps of segments. Each of the output segment is internally homogeneous in terms of encapsulated pattern and, at the same time, different from the adjacent segments. The input is a grid of signatures. The result is a raster (GeoTIFF) and vector (ESRI Shapefile) file containing the segments. Quality of segmentation

could be also determined with raster maps of inhomogeneity, isolation, and overall quality. Figure 13 presents general workflow path for segmentation.

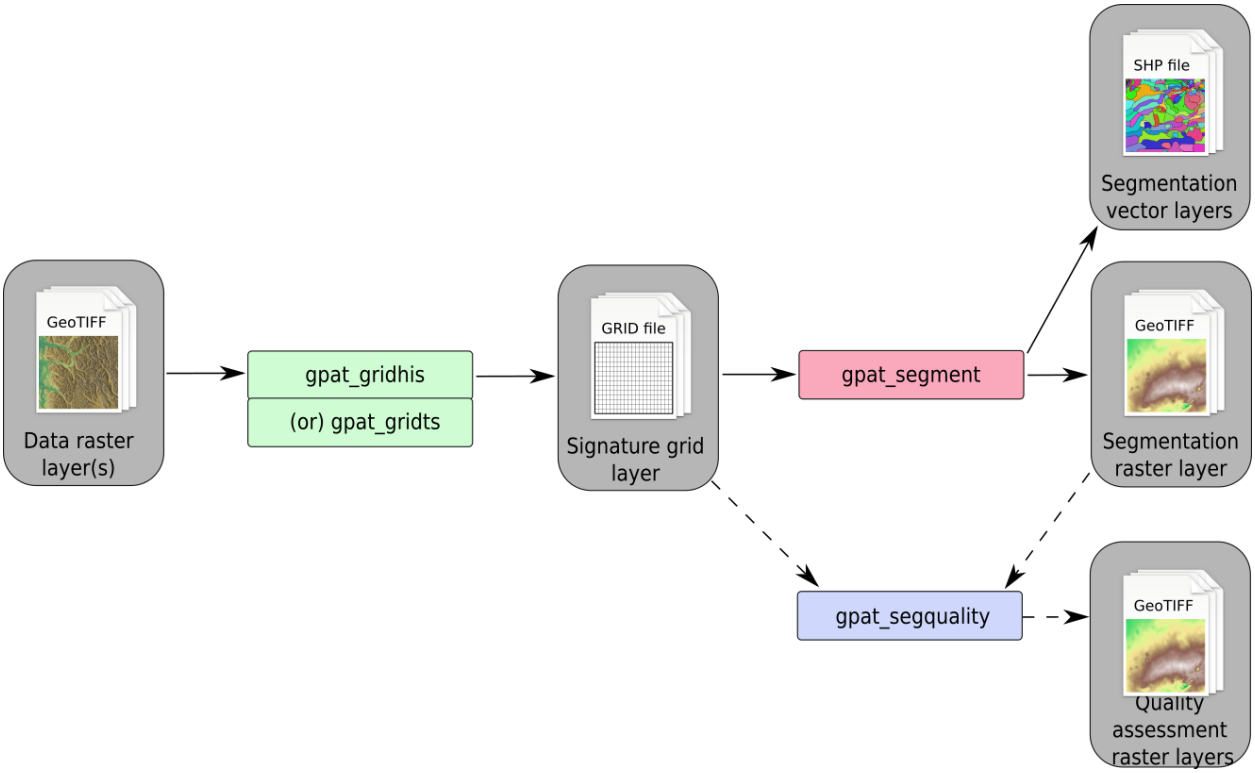

Figure 13: Workflow path for segmentation

3.3.1 Segmentation on categorical maps

Example:

```
gpat_gridhis -i Augusta2011.tif -o Augusta2011_grid100 -z 100 -f 100
gpat_segment -i Augusta2011_grid100 -o Augusta2011_seg100.tif -v Augusta2011_seg100.shp
    --lthreshold=0.1 --uthreshold=0.3
gpat_segquality -i Augusta2011_grid100 -s Augusta2011_seg100.tif -g Augusta2011_seg100_ih.tif -o
    Augusta2011_seg100_is.tif
```

The first step is to prepare a signature file for the dataset, using the `gpat_gridhis` module. The second step is to use this signature file as input to the `gpat_segment` module in order to the produce a segmentation map. The third, optional, step is to determine a quality of segmentation using the `gpat_segquality` module.

For example, we want to find and delineate areas of similar land cover patterns.

Firstly, we need to create a grid of signatures. Several options could be specified, such as the size and shift, type of motifel's signature, and normalization method. Values of size and shift could be used to delineate patterns in different spatial scale, while the motifel's signature type and normalization method need to be selected based on the output data type. We want to create segments with a spatial scale of six kilometers. To do so, we must to set adequate size and shift. As a default, segmentation process uses the brick topology, in which rectangular areas are merged to create each brick. Therefore, to create segments with a spatial scale of six kilometers (area of 36 sq km), we need to set both size and shift to 100. It is because one cell has resolution of 30 meters, and firstly using `gpat_gridhis` it is combined into a grid of signature of 3 kilometers (30 meters x 100 = 3 kilometers), and secondly using `gpat_segment` it is merged into bricks of four grids of signatures (6 by 6 kilometers). See Appendix E for the details. We left the rest of the parameters to the default values, with the signature in form of spatial coocurrence of the categories and the pdf normalization.

Secondly, the output of the previous step is used for segmentation. There we can modify several parameters of segmentation, such as the similarity measure, and minimum and maximum distance threshold to build areas. The threshold values could be used to control a number of final segments. As a result, we get the output segmentation as a raster (GeoTIFF) and vector (ESRI Shapefile) (Figure 14).

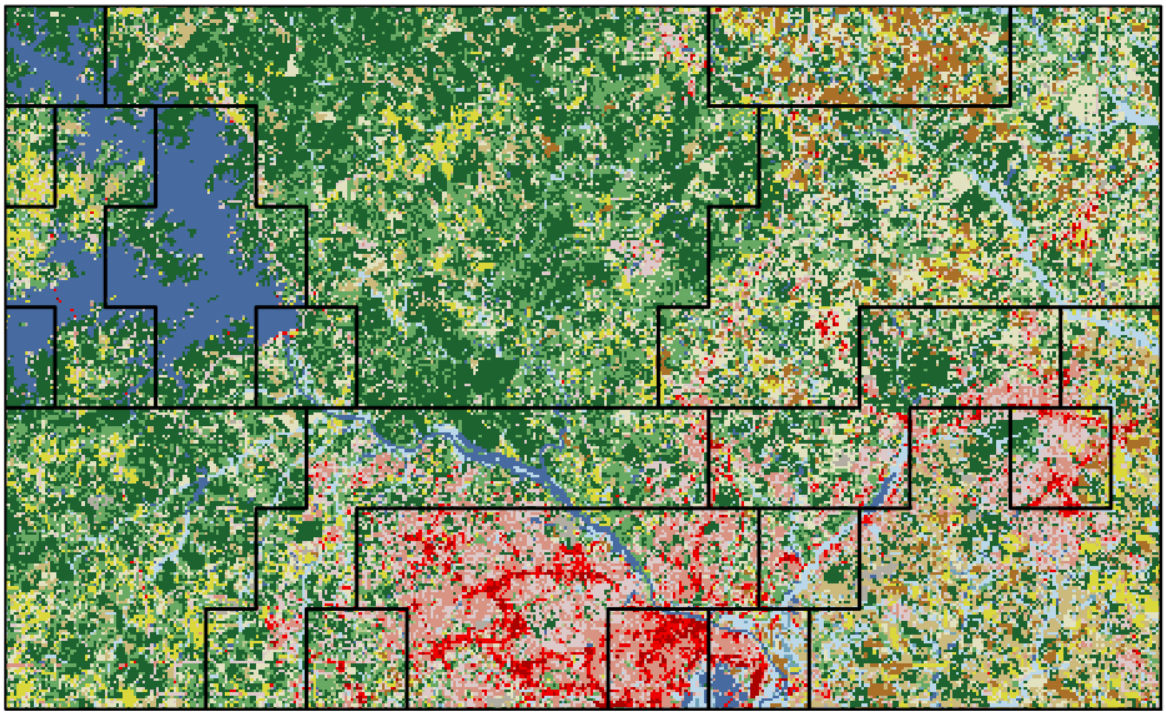

Figure 14: Segments of land cover patterns of the 6km scale for Augusta

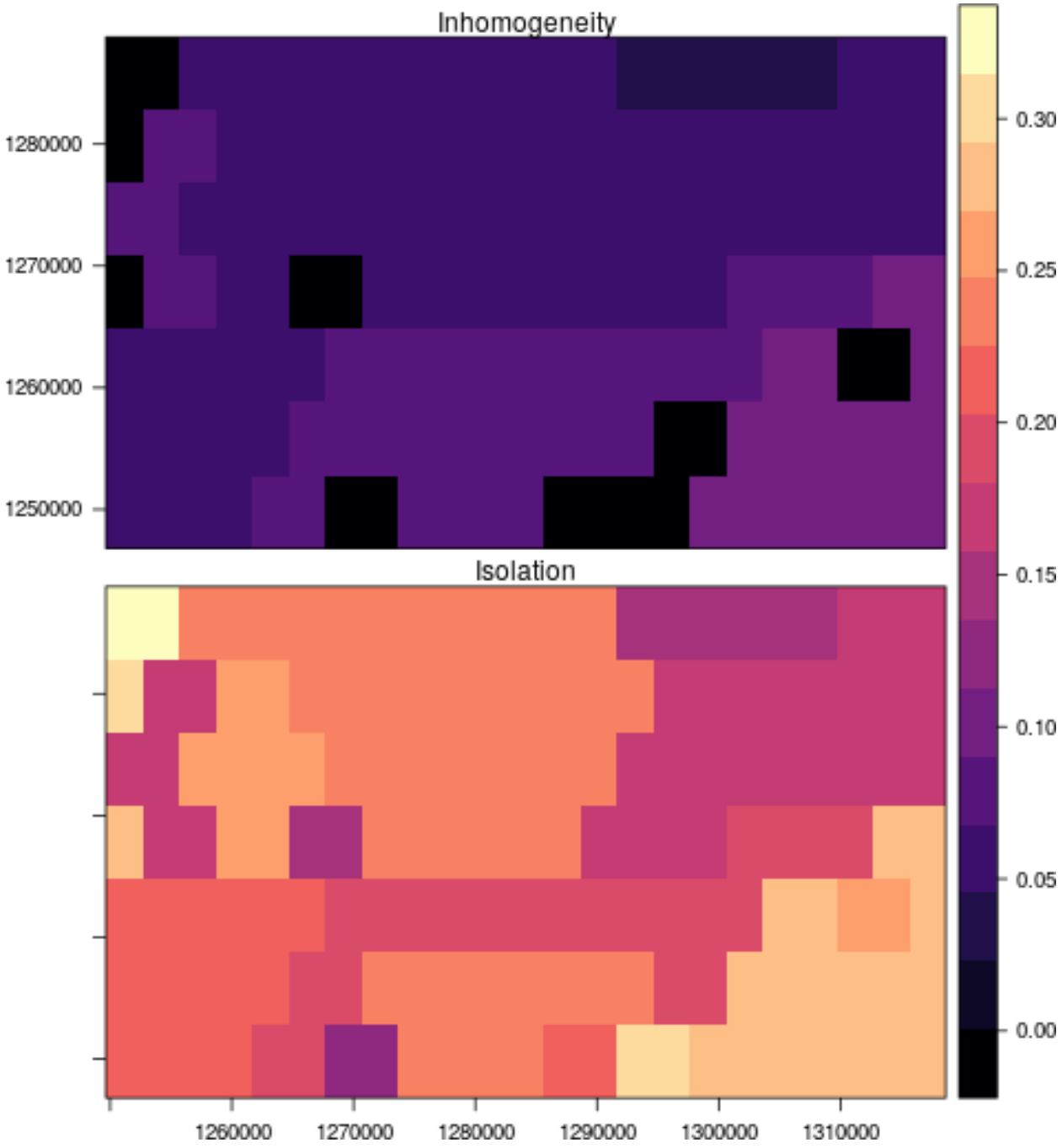

Figure 15: Inhomogenety and isolation metrics for the segmentation output

We can evaluate the quality of a segmentation using two metrics - inhomogeneity and isolation (Figure 15). Inhomogeneity indicates how much each segment is internally diverse (lower is better). Isolation shows how much each segment is different from theirs neighbors (higher is better). Based on these two raster, it is possible to calculate the segmentation quality raster ($quality = 1 - (inhomogeneity/isolation)$) (higher is better; Figure 16). We can also calculate the overall quality as an average quality of all the segments. In this example it is 0.82.

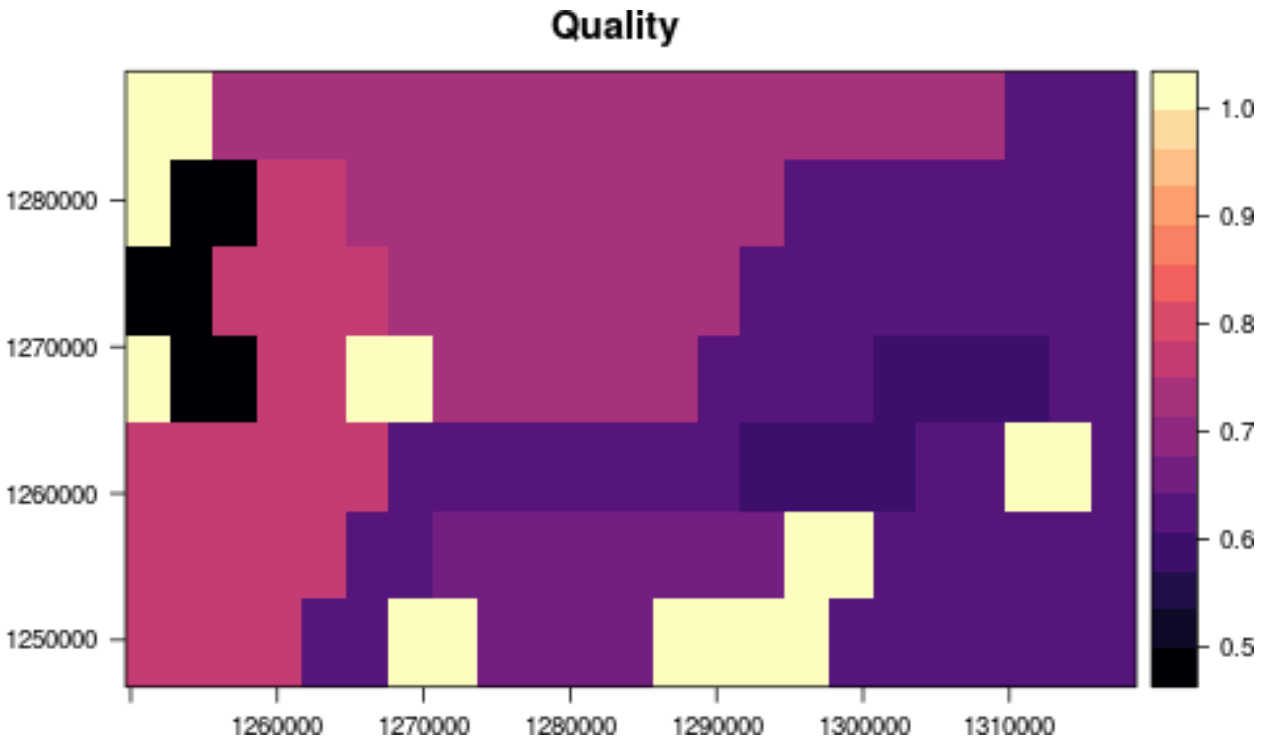

Figure 16: Quality of the segmentation of land cover patterns

3.3.2 Segmentation on time series

Example:

```
gpat_gridts -i GB_pr01.tif -i GB_pr02.tif -i GB_pr03.tif -i GB_pr04.tif -i GB_pr05.tif -i GB_pr06.tif
    -i GB_pr07.tif -i GB_pr08.tif -i GB_pr09.tif -i GB_pr10.tif -i GB_pr11.tif -i GB_pr12.tif -o
    GB_pr_grid -n
gpat_segment -i GB_pr_grid -o GB_pr_seg.tif -v GB_pr_seg.shp -m tsEUC --lthreshold=0.5
    --uthreshold=1
gpat_segquality -i GB_pr_grid -s GB_pr_seg.tif -g GB_pr_seg_ih.tif -o GB_pr_seg_is.tif -m tsEUC
```

The first step is to prepare a signature file for the dataset, using the `gpat_gridts` module. The second step is to use this signature file as input to the `gpat_segment` module in order to the produce a segmentation map. The third, optional, step is to determine a quality of segmentation using the `gpat_segquality` module.

For example, we want to find and delineate areas of similar time-series patterns of precipitation in Great Britain.

Firstly, we need to create a grid of signatures using twelve raster with monthly average values of precipitation. Depending on the input data it could be important to normalize data using the *–normalize* option.

Secondly, the output of the previous step is used for segmentation. As a result, we get the output segmentation as a raster (GeoTIFF) and vector (ESRI Shapefile) (Figure 17).

We can evaluate the quality of a segmentation using two metrics - inhomogeneity and isolation (Figure 18). Based on these two raster, it is possible to calculate the segmentation quality raster (Figure 19). We can also calculate the overall quality as an average quality of all the segments. In this example it is 0.67.

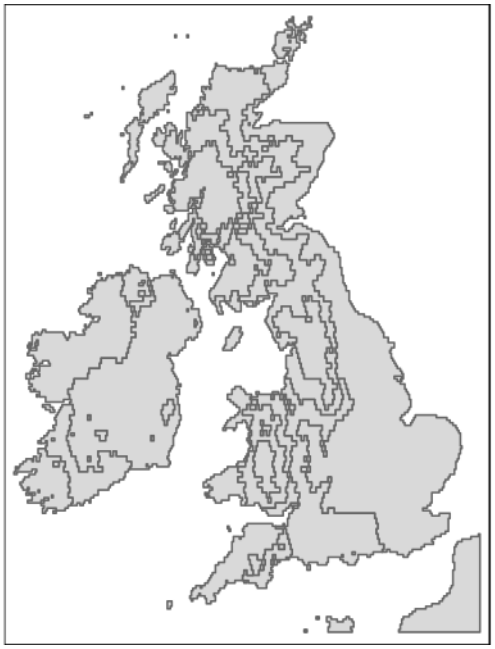

Figure 17: Segments of the monthly precipitation temporal patterns for Great Britain

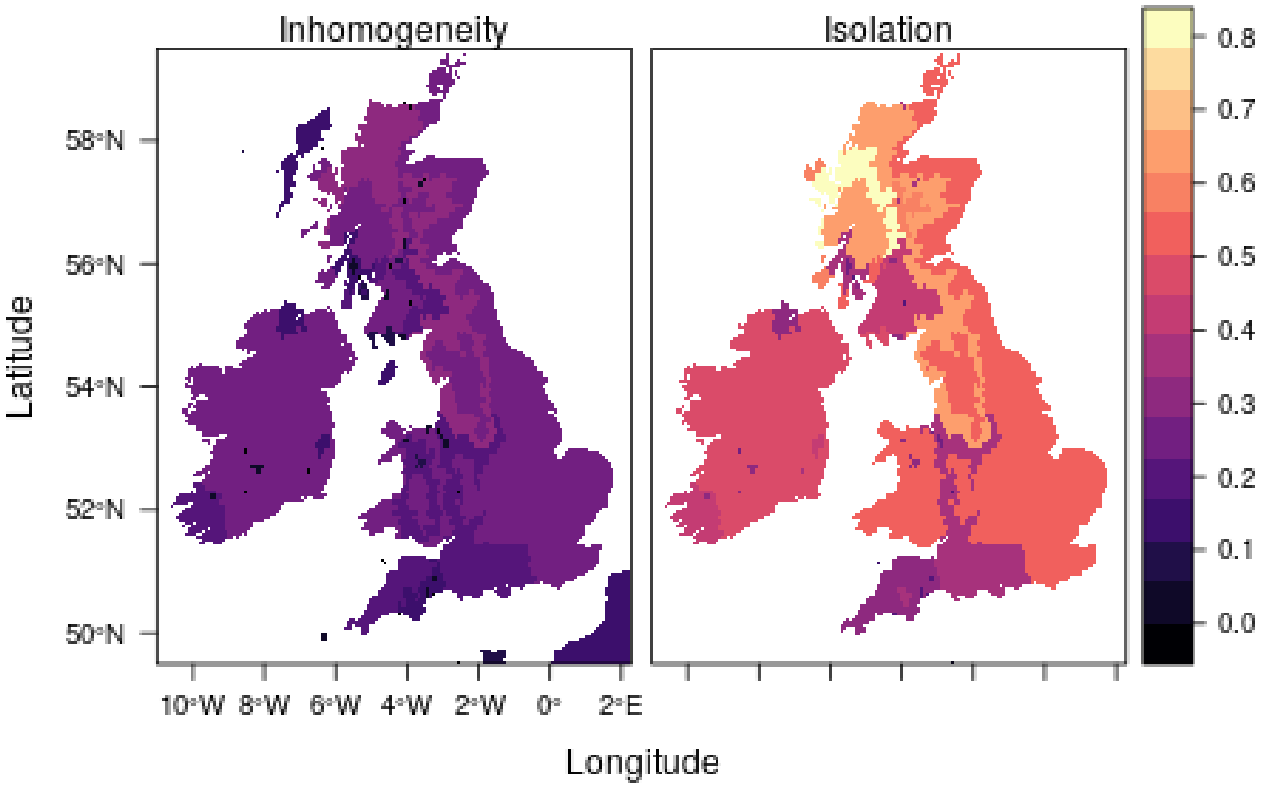

Figure 18: Inhomogenety and isolation metrics for the segmentation of the monthly precipitation temporal patterns for Great Britain

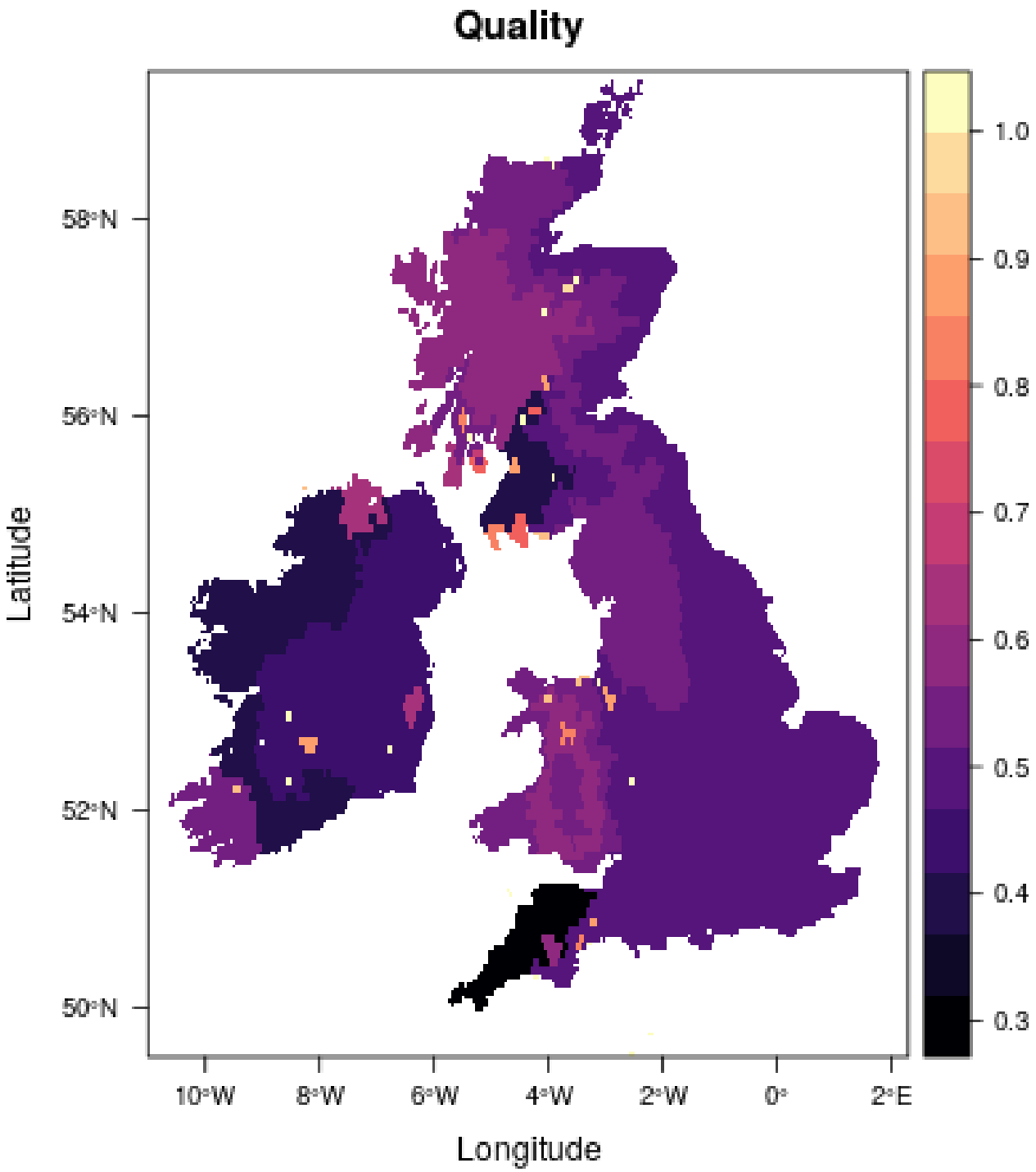

Figure 19: Quality of the segmentation of the monthly precipitation temporal patterns for Great Britain

3.4 Clustering

Clustering module enables users to create distance matrices, which can be later used in a statistical software (such as R) to perform a clustering. Distance matrix can be created for

selected motifels, a grid of motifels, and predefined regions (e.g. segments).

Examples in this section are using R ([6]) for a clustering and visualization of the results. They require the latest versions of R (`https://cloud.r-project.org/`) and a set of packages installed by executing the below code in R:

```
install.packages("sf")
install.packages("tmap")
install.packages("nowosad/rgeopat2")
```

3.4.1 Clustering of individual motifels

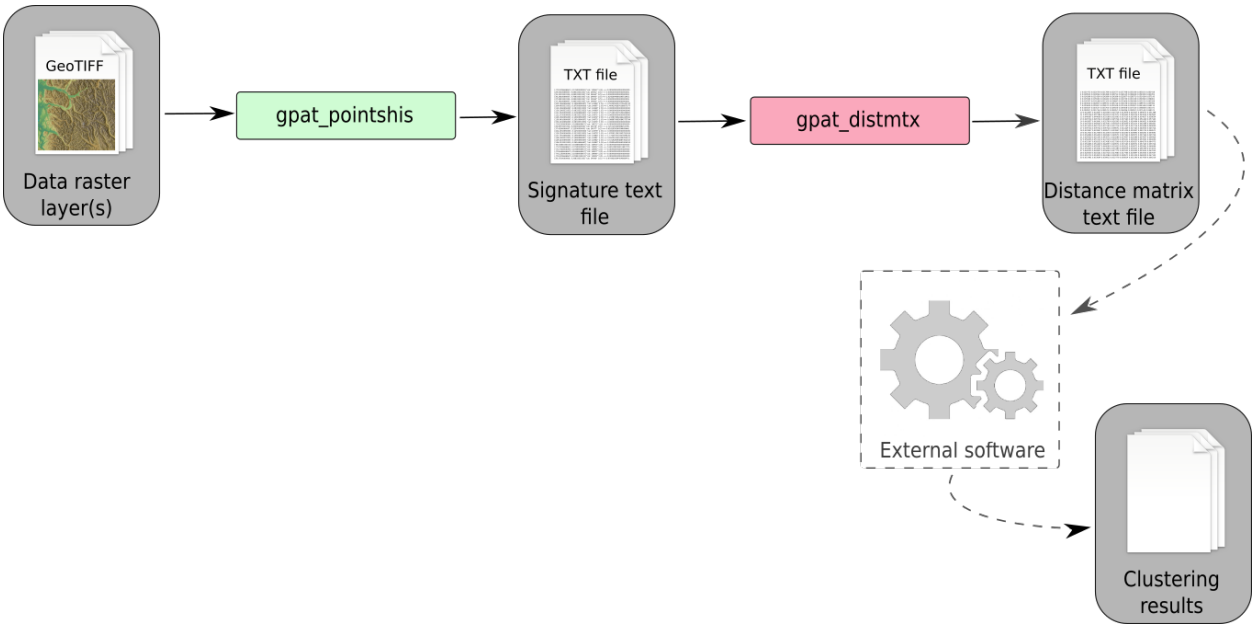

Figure 20: Workflow path for a clustering of motifels

The first step is to prepare a signature file for query motifels using the `gpat_pointsts` module. The second step is to calculate a distance matrix based on a selected similarity measure using `gpat_distmtx` (Figure 20).

Example:

```
gpat_pointshis -i Augusta2011.tif -o Augusta2011_selected.txt -s cooc -z 50 -n pdf
    --xy_file=Augusta2011_sel_points.txt
gpat_distmtx -i Augusta2011_selected.txt -o Augusta2011_matrix.csv
```

For example, we create signatures based on a given size (e.g. 50) and signature type for selected motifels using a text file with the coordinates of points of interest. Next, we calculate distances between all of these motifels using selected similarity measure (in this example, we used the default one - 'jsd' Jensen-Shannon Divergence).

Created distance matrix can be used in R:

```
library(rgeopat2)
dist_matrix = gpat_read_distmtx("Augusta2011_matrix.csv")
hclust_result = hclust(d = dist_matrix, method = "ward.D2")
plot(hclust_result, labels = FALSE)
sel_points = read.csv("Augusta2011_sel_points.txt", header = FALSE)
sel_points$class = cutree(hclust_result, k = 5)
plot(augusta2011)
points(sel_points, pch = 20, cex = 3, col = sel_points$class)
```

The `gpat_read_distmtx` function reads a distance matrix from a file. Next, this file is used for a clustering. There are several clustering methods in R that accept a distance matrix as an input, such as hierarchical clustering (`hclust`) and partitioning around medoids (`cluster::pam`). Here, we use and visualize a result of hierarchical clustering (left panel of figure 21). Based on the dendrogram, data is divided into five groups and the information about class numbers is added to the object containing our selected points. Finally, we visualize the results (right panel of figure 21).

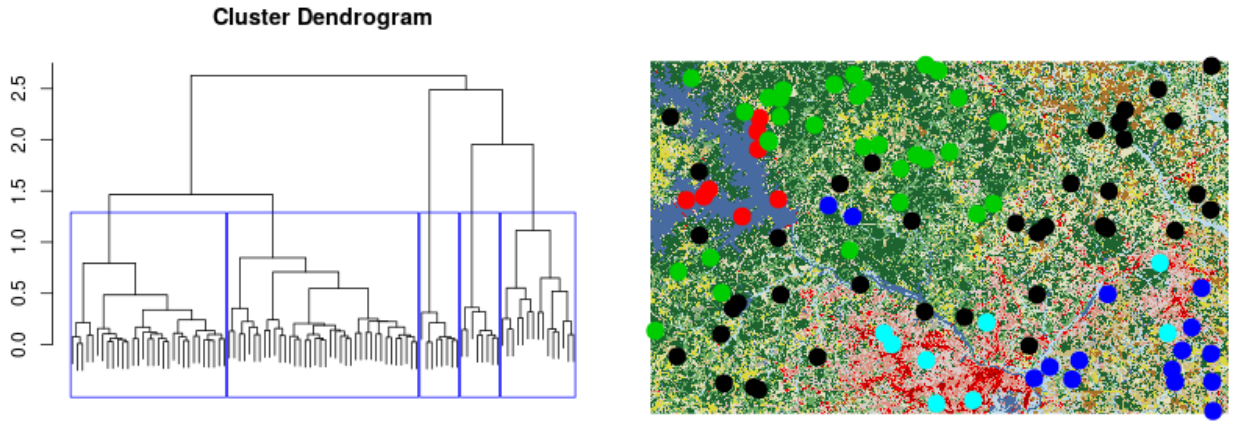

Figure 21: Example of a clustering of motifels

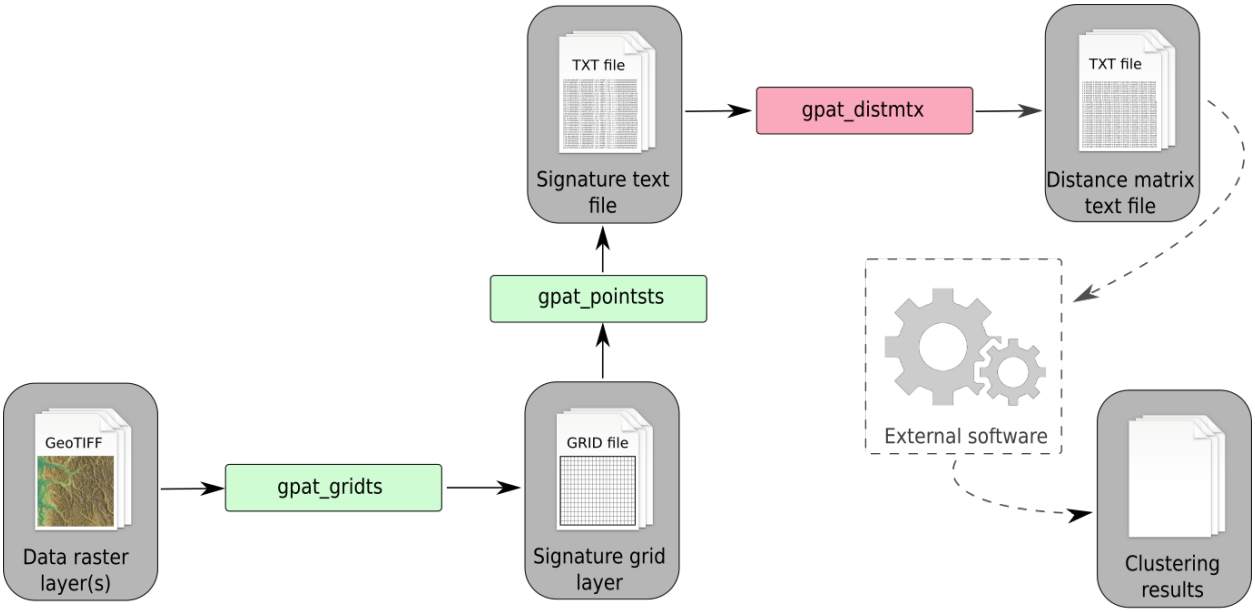

Figure 22: Workflow path for a clustering of time series motifels

Procedure for time-series data is slightly different (Figure 22). The first step is to create a signature grid file using a time-series of rasters using the `gpat_gridts` module. The second step is to extract only selected motifels using `gpat_pointsts`. The third step is to calculate and export a distance matrix with `gpat_distmtx`.

Example:

```
gpat_gridts -i GB_pr01.tif -i GB_pr02.tif -i GB_pr03.tif -i GB_pr04.tif -i GB_pr05.tif -i GB_pr06.tif
    -i GB_pr07.tif -i GB_pr08.tif -i GB_pr09.tif -i GB_pr10.tif -i GB_pr11.tif -i GB_pr12.tif -o
    GB_pr_grid -n
gpat_pointsts -i GB_pr_grid -o GB_pr_points.txt --xy_file=GB_cities.csv
gpat_distmtx -i GB_pr_points.txt -o GB_pr_distmat.csv -m tsEUC
```

For example, we want to find cities in Great Britain with very similar temporal patterns of a monthly sum of precipitation. For this purpose, we need to have twelve rasters with values of a monthly sum of precipitation for the whole area of Great Britain and the list of the cities coordinates (the GB_cities.csv file). Firstly, we can create a signature grid. Next step is to extract the signature only for the locations in our GB cities.csv file. Finally, we can calculate and export distance matrix using euclidean distance *tsEUC*.

Created distance matrix can be used in R:

```
library(sf)
library(tmap)
library(rgeopat2)
dist_matrix = gpat_read_distmtx("GB_pr_distmat.csv")
hclust_result = hclust(d = dist_matrix, method = "ward.D2")
plot(hclust_result, labels = FALSE)
sel_points = read.csv("GB_cities.csv", header = FALSE)
sel_points$class = as.factor(cutree(hclust_result, k = 4))
sel_points_st = st_as_sf(sel_points, coords = c("V1", "V2"))
tm_shape(british_isles) +
        tm_polygons() +
        tm_shape(sel_points_st) +
        tm_dots(col = "class", size = 0.25, title = "Class: ")
```

The `gpat_read_distmtx` function reads a distance matrix from a file. Next, we use and visualize a result of hierarchical clustering (left panel of figure 23). Based on the dendrogram, data is divided into four groups and the information about class numbers is added to the object containing our selected points. Using the `st_as_sf` function, we transform our selected points into spatial object. Finally, we visualize the results using functions from the `tmap` package (right panel of figure 21). For this purpose, we use a map of British Isles *british_isles* from the `rgeopat2` package and the spatial object with our selected points (*sel_points_st*).

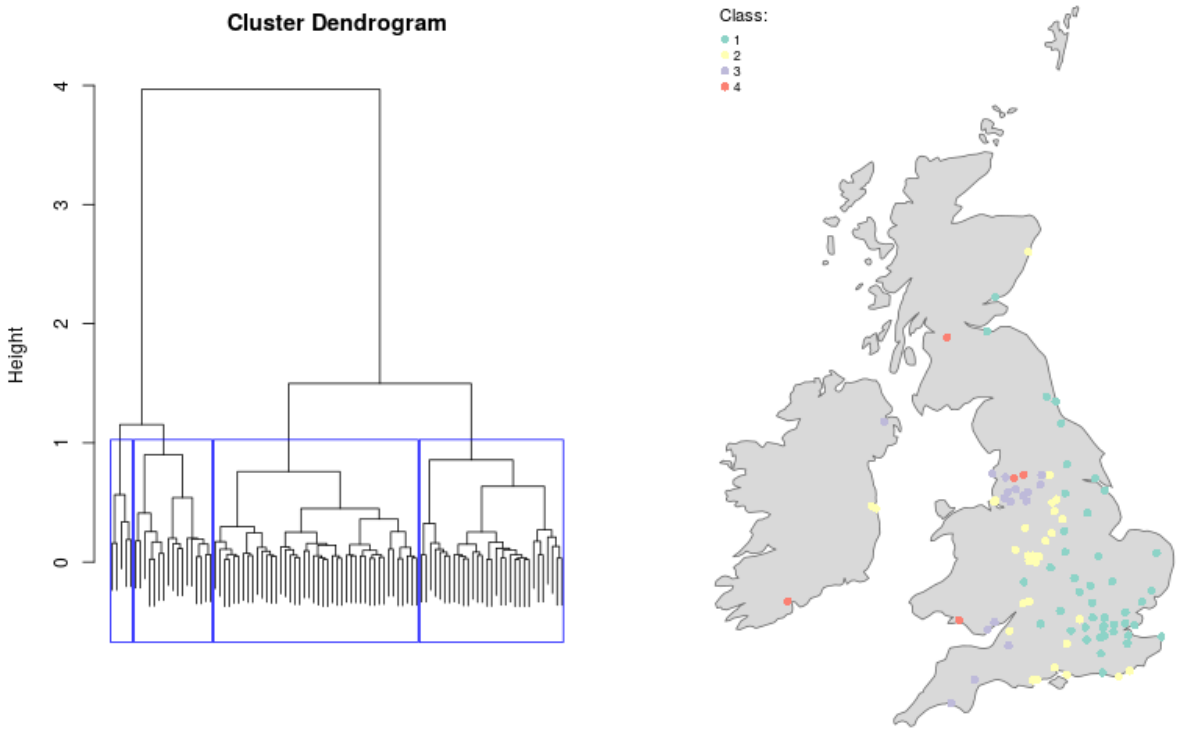

Figure 23: Example of a clustering of time series motifels

3.4.2 Clustering of a grid of motifels

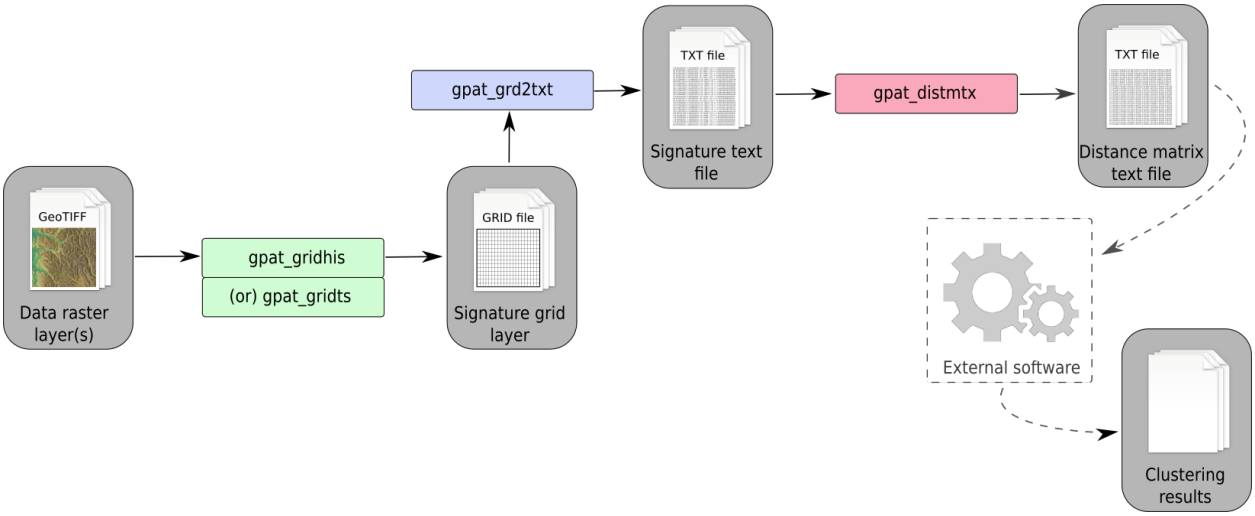

Figure 24: Workflow path for a clustering of a grid of motifels

The first step is to create a signature grid using either the `gpat_gridhis` or `gpat_gridts` module. The second step is to convert the obtained signature grid from binary to text format with `gpat_grd2txt`. The final step is to calculate and export a distance matrix using the `gpat_distmtx` module (Figure 24).

Example:

```
gpat_gridhis −i Augusta2011.tif −o Augusta2011_grid100 −z 100 −f 100
gpat_grd2txt −i Augusta2011_grid100 −o Augusta2011_grid100.txt
gpat_distmtx −i Augusta2011_grid100.txt −o Augusta2011_matrix_grid.csv
```

For example, we want to find groups of areas with similar pattern of a land cover in scale of 3 kilometers. Firstly, we need to calculate a signature grid with the size and shift options set to 100 (30 meters resolution multiplied by 100 is 3 kilometers). Next, we convert the signature grid to a text file. This new file will be an output to for the calculations of a distance matrix. A default similarity measure, Jensen-Shannon divergence, can be used here, as it is suitable for a land cover data.

Created distance matrix can be used in R:

```
library(sf)
library(rgeopat2)
dist_matrix = gpat_read_distmtx("Augusta2011_matrix_grid.csv")
hclust_result = hclust(d = dist_matrix, method = "ward.D2")
plot(hclust_result, labels = FALSE)
hclust_cut = cutree(hclust_result, 5)
my_grid = gpat_gridcreate("Augusta2011_grid100.hdr")
my_grid$class = hclust_cut
plot(my_grid)
```

The `gpat_read_distmtx` function reads a distance matrix from a file. Next, we use and visualize a result of hierarchical clustering (left panel of figure 25). Based on the dendrogram, data is divided into five groups and the information about class numbers is added to the

object containing our selected points. Using the `gpat_gridcreate` function, we create a new grid polygon and add the class numbers to it. Finally, we visualize the results (right panel of figure 25).

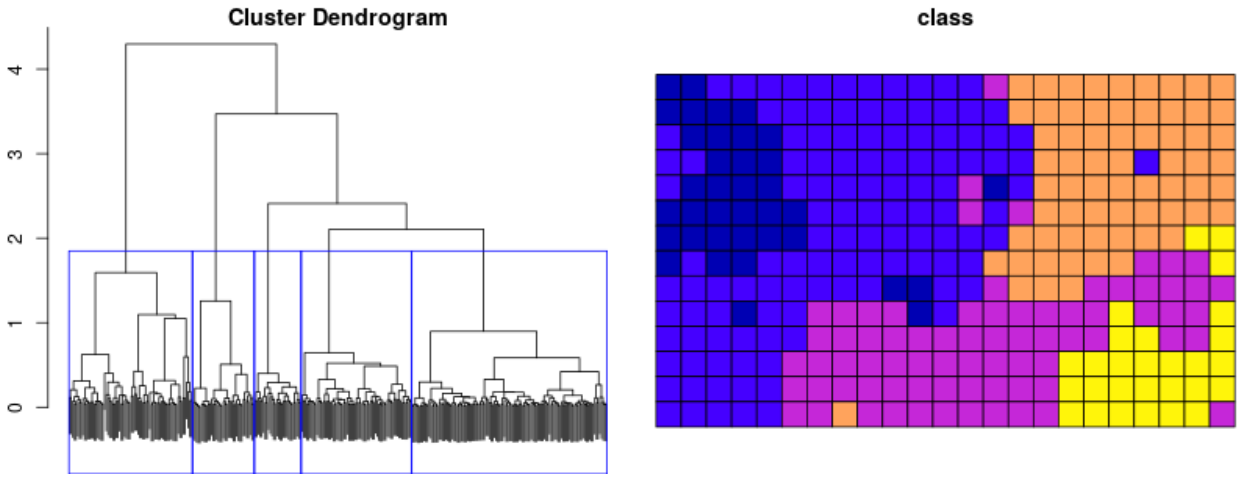

Figure 25: Example of a clustering of a grid of motifels

3.4.3 Clustering of segments / predefined irregular regions

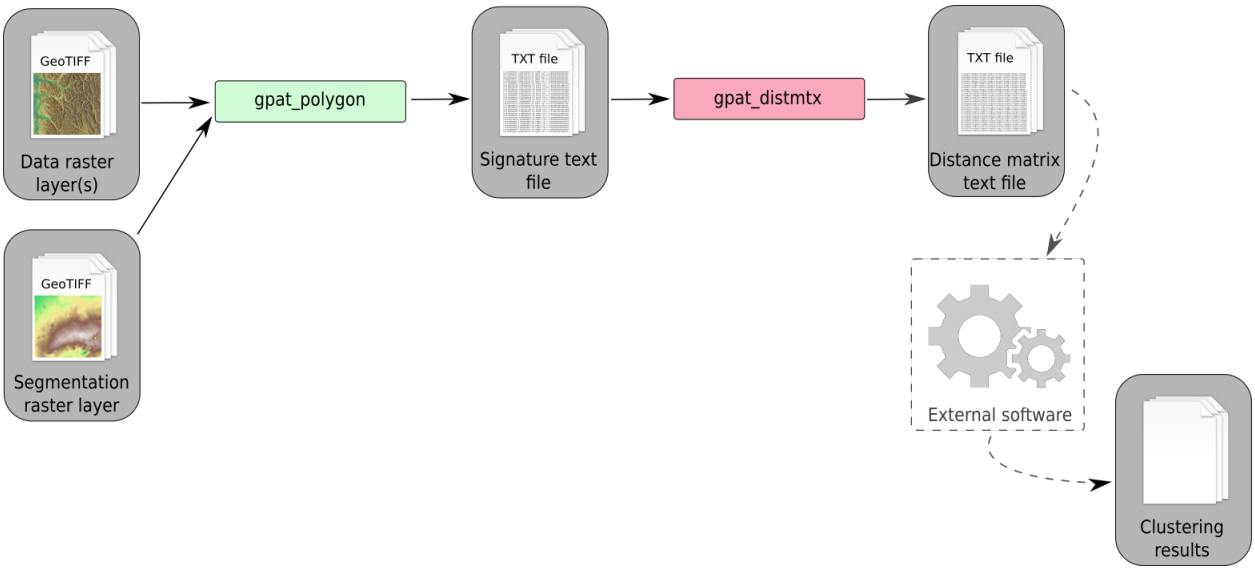

Figure 26: Workflow path for a clustering of segments (regions)

The first step is to unify the resolution and extent of the input raster data and the segment file, which can be done using a GIS software, such example `gdalwrap`. The second step is to calculate the signature file based on a raster data which contain a pattern to analyze and a raster data with segments (or any other irregular regions) using the `gpat_polygon` module. The third step is to create and export a distance matrix using `gpat_distmtx` (Figure 26).

Example:

```
gdalwarp -tr 30 30 Augusta2011_seg100.tif Augusta2011_seg100_res.tif
gpat_polygon -i Augusta2011.tif -e Augusta2011_seg100_res.tif -o Augusta2011_psign.txt
gpat_distmtx -i Augusta2011_psign.txt -o Augusta2011_matrix_seg.csv
```

For example, in the section 3.3 we created segments with homogeneous patterns of land cover. Now we can find and group similar non-adjacent segments with clustering. Firstly, we need to change the resolution of the segment file to 30 meters (this is the resolution of the land cover data). Next, we need to create a signature text file for each of the segments. Based on the new file we can calculate a distance between each pair of the segments using selected dissimilarity metric, such as Jensen-Shannon divergence.

Created distance matrix can be used in R:

```
library(sf)
library(rgeopat2)
dist_matrix = gpat_read_distmtx("Augusta2011_matrix_seg.csv")
hclust_result = hclust(d = dist_matrix, method = "ward.D2")
plot(hclust_result, label = FALSE)
hclust_cut = cutree(hclust_result, 3)
segm = st_read("Augusta2011_seg100.shp")
segm$class = hclust_cut
plot(segm["class"])
```

The `gpat_read_distmtx` function reads a distance matrix from a file. Next, we use and visualize a result of hierarchical clustering (left panel of figure 27). Based on the dendrogram, data is divided into three groups and the information about class numbers is added to the object containing our selected points. Using the `st_read` function, we read our polygons into R and add the class number to each polygon. Finally, we visualize the results (right panel of figure 27).

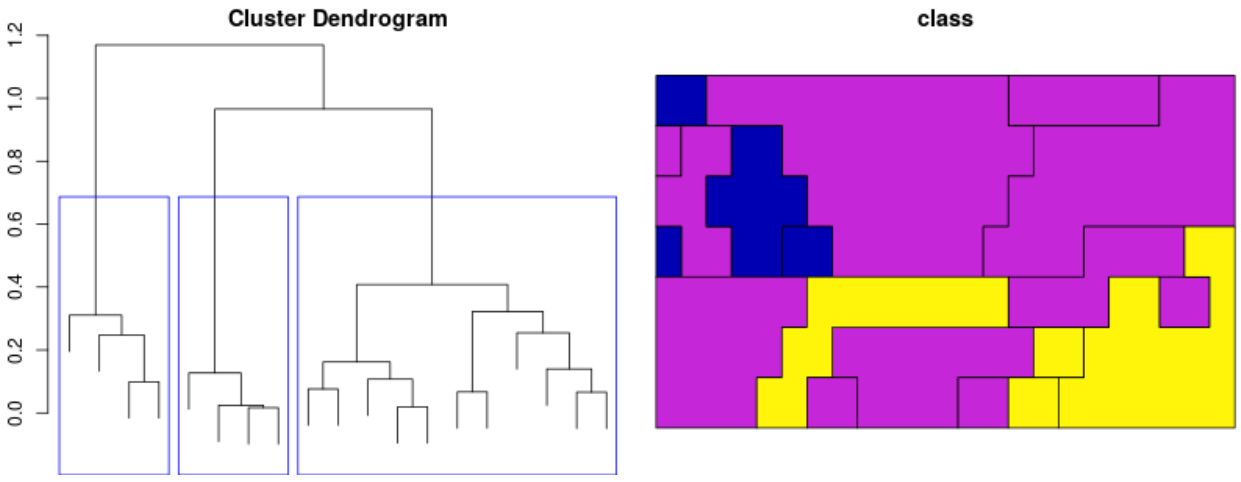

Figure 27: Example of a clustering of a grid of segments (regions)

Appendices

A GeoPAT 2.0 installation

A.1 System requirements

The Windows installer and Linux binaries provide all necessary components. To build GeoPAT 2.0 from the source code, the user needs to install the GDAL library and compile and install the ezGDAL and SML libraries.

A.2 Windows installer

The Windows installer works under 64 bit versions of Windows 7, 8.1, and 10. The installer provides four components:

- GDAL library package
- ezGDAL and SML libraries
- GeoPAT 2.0 software
- Microsoft Visual C 13 runtime libraries (optional)

To start the installation process user has to run GPAT20setup.exe. The setup program should be run in "Run as administrator" mode. If an antivirus software is running on the computer, user should turn it off temporary for the time of GeoPAT installation. The installer will create the directory for GDAL and GeoPAT and copy all necessary files. Optionally, GPAT20setup.exe will start the Microsoft Visual C runtime libraries installer. When installation is completed, the user can find "SIL GeoPAT 2.0" in the Windows start menu with two files - "GeoPAT console" and "Uninstall GeoPAT".

The installer for Windows x64 is available at:

```
http://sil.uc.edu/cms/data/uploads/software_data/GPAT2/GPAT20setup.exe
```

A.3 Building from source code

The building from source code procedure is presented using the Fedora 25 Linux distribution. This procedure could diffet between Linux distributions, therefore the user should modify it to their system.

To build GeoPAT 2.0 from the source, the user has to do the four following steps:

1. Install the developement version of GDAL
2. Build and install the ezGDAL library
3. Build and install the SML library

4. Build and install the GeoPAT 2.0 software

The dnf package manager can be used in a Fedora system to install the developement version of GDAL:

```
dnf install gdal-devel
```

To install the ezGDAL library the user has to download the ezGDAL source code and unpack it. The source code of ezGDAL is available at:

```
http://pawel.netzel.pl/gitlist/index.php/ezGDAL/zipball/master
```

Next, the user has to compile the code by calling the following command in an unpacked source code directory:

```
make
```

And install it using:

```
make install
```

By default, the library is placed in the /usr/local/lib directory and the include file is placed in /usr/local/include. The user can change the destination directory by adding the PREFIX parameter.

```
make PREFIX=/my/destination/directory
```

When PREFIX is provided, the library is placed in /my/destination/directory/lib and the include file is placed in /my/destination/directory/include.

The installation procedure of the SML library is similar. The source code of SML is available at:

```
http://pawel.netzel.pl/gitlist/index.php/SML/zipball/master
```

After an extraction of the source code of SML, the user should call:

```
make
make install
```

The command "make install" should be called using the sudo command or in the root user context. After finishing libraries installation procedure the user has to ensure that PREFIX/lib is on the library search path.

The last step of the installation procedure is a compilation of the GeoPAT 2.0 source code. The source code of GeoPAT 2.0 is available at:

```
https://github.com/Nowosad/geopat2/archive/master.zip
```

GeoPAT 2.0 depends on the GDAL, SML, and ezGDAL libraries. Therefore, after the installation of the above libraries, the user has to unpack, compile and install the GeoPAT 2.0. The installation procedure is similar to the installation procedures of the ezGDAL and SML libraries.

```
make
make install
```

The "make" and "make install" commands should be run in the main GeoPAT 2.0 source code directory. PREFIX parameter works in the same way.

B Numerical signatures available in GeoPAT

A signature is a numerical description of a motifel.

B.1 Cartesian product (*prod*)

This method calculates signature as a k-dimensional histogram using k primitive features assigned to each cell. Examples of such features include cell class, the size of the clump to which the cell belongs, the shape of the clump and its spatial orientation ([8]). Because all features must be categorical (so the histogram can be formed), numerical features need to be categorized. For example, clump sizes need to be categorized into size categories from the smallest to the largest. The number of bins in the crossproduct histogram is $N_1 \times N_2 \times \ldots \times N_k$, where N_i is the number of categories of i-th feature. Crossproduct signature is designed to be effective in encapsulating spatial structures with clear geometric quality (having relatively low complexity); an example of a dataset with such a structure is the land cover raster.

B.2 Class co-occurrence histogram (*cooc*)

This method uses a color co-occurrence histogram ([9], [10]), a variant of the Gray-Level Co-occurrence Matrix (GLCM) originally introduced by [11] to characterize texture in grayscale images. In GeoPAT, color is replaced by cell class and a single cell separation of one pixel is used to calculate a co-occurrence histogram. This results in a single primitive feature - a pair of classes assigned to two neighboring cells; eight-connectivity is assumed for establishing the existence of a neighborhood relationship between the two cells. Thus, eight features are calculated for each cell, but their total number is halved as the same feature is generated twice by the pairs of neighboring cells. For a scene with k cell classes, the co-occurrence histogram has $(k^2 + k)/2$ bins, k of them correspond to same-class pairs, which measure the composition of the classes in the scene, and $(k^2 - k)/2$ bins correspond to different-class pairs, which measure the configuration of the classes in the scene. The co-occurrence signature is designed to be effective in encapsulating spatial structures exhibiting high complexity patterns like the ones resulting from a geomorphons-based classification of a DEM.

B.3 Decomposition histogram (*fdec* and *sdec*)

This signature method, inspired by the work of [12], is used to describe a scene using a set of sub-scenes having a hierarchy of sizes. For the decomposition method to work best the scene should be a square having a linear size of 2^D cells. The scene with k cell classes is scanned without overlap by a series of square moving windows with sizes $w = 2^i$ cells where $i = 2, \ldots, D$ are the decomposition levels. The size of the maximum scanning window, 2^D, is the size of the scene. At the smallest decomposition level $i = 2$ a scene is scanned by a window having a size of $4 \times 4 = 16$ cells. At each scanning position the percentages, $p_1, \ldots, p_k$, of the window's area occupied by cells having classes $1, \ldots, k$, respectively are recorded and a window area is assigned a list of k tags (one for each class) representing those percentages. These tags are classified into one of three categories, 1 if the percentage is below $\frac{1}{4}$, 2 if it is between $\frac{1}{4}$ and $\frac{1}{2}$, and 3 if it is above $\frac{1}{2}$. Tallying all tags results in a

histogram with $3 \times k$ bins (three bins for each class). The decomposition signature is designed to be effective for patterns of all levels of complexity, however, we have not yet accumulated sufficient experience working with this signature to offer definitive advice on the types of datasets to which it can be best applied.

B.4 Landscape indices vector (*lind* and *linds*)

Two types of landscape indices are present in GeoPAT - landscape level indices and class level indices. Each of landscape level indices produce one value for each "landscape" (basic unit of analysis - it could be a motifel or a polygon). Class level indices produce as many values as there are classes in the input dataset. For example, class percentage of landscape *pland* gives the proportional abundance of each class for each "landscape" in percentages. Therefore, it produces one value for each class for each "landscape". The complete list of landscape indices supported by GeoPAT is in Table 1.

GeoPAT provides two ways to calculate landscape indices - *lind* (Landscape indices vector) and *linds* (Selected landscape indices vector). The *lind* signature calculates all of the landscape level indices first, and all of the class level indices second. Class level indices are arranged in such a way that the first index (*pland*) is calculated for all classes, then the second index (*lpi*) for all classes, etc. The *linds* signature calculates the landscape level indices and only one of the class level indices - class percentage of landscape *pland*. The reasoning for why using the aforementioned selected set of landscape indices (*linds*) for comparing landscapes is reasonable can be found in [13]. This method is still experimental though.

ID	Name	Description	Landscape level	Class level
1	pland	class percentage of landscape	0	1
2	lpi	largest patch index	1	1
3	ed	edge density	1	1
4	area_mn	patch area mean	1	1
5	area_am	patch area area-weighted mean	1	1
6	area_md	patch area median	1	1
7	area_ra	patch area range	1	1
8	area_sd	patch area standard deviation	1	1
9	area_cv	patch area coeff. of variation	1	1
10	gyrate_mn	radius of gyration mean	1	1
11	gyrate_am	radius of gyration area-weighted mean	1	1
12	gyrate_md	radius of gyration median	1	1
13	gyrate_ra	radius of gyration range	1	1
14	gyrate_sd	radius of gyration standard deviation	1	1
15	gyrate_cv	radius of gyration coeff. of variation	1	1
16	pafrac	perimeter-area fractal dimension	1	1
17	para_mn	perimeter-area ratio mean	1	1
18	para_am	perimeter-area ratio area-weighted mean	1	1
19	para_md	perimeter-area ratio median	1	1
20	para_ra	perimeter-area ratio range	1	1
21	para_sd	perimeter-area ratio standard deviation	1	1

22	para_cv	perimeter-area ratio coeff. of variation	1	1
23	shape_mn	shape index mean	1	1
24	shape_am	shape index area-weighted mean	1	1
25	shape_md	shape index median	1	1
26	shape_ra	shape index range	1	1
27	shape_sd	shape index standard deviation	1	1
28	shape_cv	shape index coeff. of variation	1	1
29	frac_mn	fractal index mean	1	1
30	frac_am	fractal index area-weighted mean	1	1
31	frac_md	fractal index median	1	1
32	frac_ra	fractal index range	1	1
33	frac_sd	fractal index standard deviation	1	1
34	frac_cv	fractal index coeff. of variation	1	1
35	contig_mn	contiguity index mean	1	1
36	contig_am	contiguity index area-weighted mean	1	1
37	contig_md	contiguity index median	1	1
38	contig_ra	contiguity index range	1	1
39	contig_sd	contiguity index standard deviation	1	1
40	contig_cv	contiguity index coeff. of variation	1	1
41	contag	contagion index	1	0
42	iji	interspersion & juxtaposition index	1	1
43	pladj	percentage of like adjacencies	1	1
44	ai	aggregation index	1	1
45	lsi	landscape shape index	1	1
46	cohesion	patch cohesion index	1	1
47	pd	patch density	1	1
48	split	splitting index	1	1
49	division	landscape division index	1	1
50	mesh	effective mesh size	1	1
51	prd	patch richness density	1	0
52	rpr	relative patch richness	1	0
53	shdi	Shannon's diversity index	1	0
54	sidi	Simpson's diversity index	1	0
55	msidi	modified Simpson's diversity index	1	0
56	shei	Shannon's evenness index	1	0
57	siei	Simpson's evenness index	1	0
58	msiei	modified Simpson's evenness index	1	0

Table 1: Landscape metrics implemented in GeoPAT 2.0

More information about landscape indices can be found in the FRAGSTATS documentation ([14]).

B.5 Shannon entropy (*ent*)

This signature contains three values:

1. Shannon entropy
2. Number of unique values in a motifel/segment
3. Size of a motifel/segment

This signature was not designed for purposes of geoprocessing (searching, comparision, segmentation, clustering), but rather for diagnostic purposes.

C Normalization methods available in GeoPAT

There are two ways to normalize pattern signatures. The first one (used in `gpat_gridhis`, `gpat_pointshis` and `gpat_polygon`) works independently on each of the pattern signatures. The role of the second one (used in `gpat_globnorm` is to normalize each of the signature elements for all the grid. Decision on the normalization method and its type should depends on both, numerical signatures and dissimilarity measures used. For example, the class co-occurrence histogram signature should be locally normalized (used in `gpat_gridhis`, `gpat_pointshis` and `gpat_polygon`) before a segmentation using the Jensen Shannon Divergence measure. However, in case of a segmentation based on the landscape indices a global normalization `gpat_globnorm` is recommended, while a local normalization should be avoided.

C.1 01

Normalization to the [0,1] interval.

C.2 pdf

Normalization to pdf (sum(hi) = 1).

C.3 N01

Normalization to N(0,1) (hi = (hi - avg) / std).

C.4 ind01

Normalization to [0,1] for 72 landscape indices. Used in the `gpat_globnorm` module only. It accepts only the grids built using the selected landscape indices vector signature (*linds*).

C.5 none

Normalization process is skipped.

D Dissimilarity measures available in GeoPAT

Distance, which assesses the degree of dissimilarity between two scenes, is the opposite of similarity. When the value of distance function is equal to zero identical histograms are indicated, and thus scenes have identical or very similar patterns, whereas large values of the distance function indicate very different histograms and scenes having significantly different patterns. Note that all histogram distance measures are heuristic and no single measure will work well with all signatures.

D.1 Jensen Shannon Divergence (*jsd*)

This measure ([15]) expresses the informational distance between two histograms P and Q by calculating a deviation between the Shannon entropy of the mixture of the two histograms $(P+Q)/2$ (the second term in eq. 2 below) and the mean of their individual entropies (the second term in eq. 2). The value of the Jensen-Shannon divergence is given by the following formula,

$$d_{JSD} = \sqrt{\sum_{i=1}^{d}\left[\frac{P_i log_2 P_i + Q_i log_2 Q_i}{2} - \left(\frac{P_i+Q_i}{2}\right) log_2\left(\frac{P_i+Q_i}{2}\right)\right]} \tag{2}$$

where d is the number of bins (the same for both histograms) and P_i and Q_i are the values of ith bin in the two histograms. We have found Jensen-Shannon divergence works well (yields dissimilarity values in agreement with human visual perception) for comparison of land cover patterns as encapsulated by crossproduct signatures ([16]–[18]).

D.2 Euclidean distance (*euc*)

D.3 Normalized euclidean distance (*eucn*)

D.4 Wave-Hedges distance (*wh*)

This measure is designed to work with co-occurrence signature histograms that tend to be dominated by bins corresponding to adjacent cells having the same class. The value of the Wave Hedges distance is given by the following formula ([19]),

$$d_{WH} = \sum_{i=0}^{d} e_i \frac{|P_i - Q_i|}{max(P_i, Q_i)} \tag{3}$$

where d is the maximum number of possible bins and $e_i = 1$ if $max(P_i, Q_i) > 0$ or $e_i = 0$ otherwise. In other words, only pattern features present in at least one of the two scenes contribute to the value of the distance. In Wave Hedges distance formula all present features contribute to the overall value of distance with the same weights regardless of feature abundance in the scenes. In the case of the co-occurrence histogram this means that composition-related features and configuration-related features contribute equally to the distance value despite the heavy dominance of composition-related features in histograms stemming from all realistic scenes. This makes the Wave Hedges distance particularly suitable for comparison of terrain scenes as encapsulated by co-occurrence signatures ([20]).

D.5 Jaccard distance (jac)

This measure is an extension of the Jaccard similarity coefficient ([21]) , originally developed to assess a similarity between two sets, but used here for assessing the dissimilarity between two histograms. The value of the Jaccard distance is given by the following formula ([19]),

$$d_J = 1 - \frac{\sum_{i=1}^{d} P_i Q_i}{\sum_{i=1}^{d} P_i^2 + \sum_{i=1}^{d} Q_i^2 - \sum_{i=1}^{d} P_i Q_i} \tag{4}$$

We have found that the Jaccard distance works well for comparison of land cover patterns as encapsulated by decomposition signatures.

D.6 Euclidean distance - time series ($tsEUC$)

D.7 Euclidean distance - periodic time series ($tsEUCP$)

D.8 Dynamic Time Warping distance - time series ($tsDTW$)

D.9 Dynamic Time Warping distance - periodic time series ($tsDTWP$)

D.10 Synchronized Dynamic Time Warping distance - time series ($tsDTWPa$)

E Topologies available in GeoPAT

Basic concept in GeoPAT 2.0 is to divide a raster into a regular grid of motifels. This grid encapsulate a complex content - a pattern corresponding to composition and spatial configuration of cells in each motifels. For the geoprocessing operations, such as search and comparison, a grid of motifels has a simple rectangular grid topology (left panel in Figure 28).

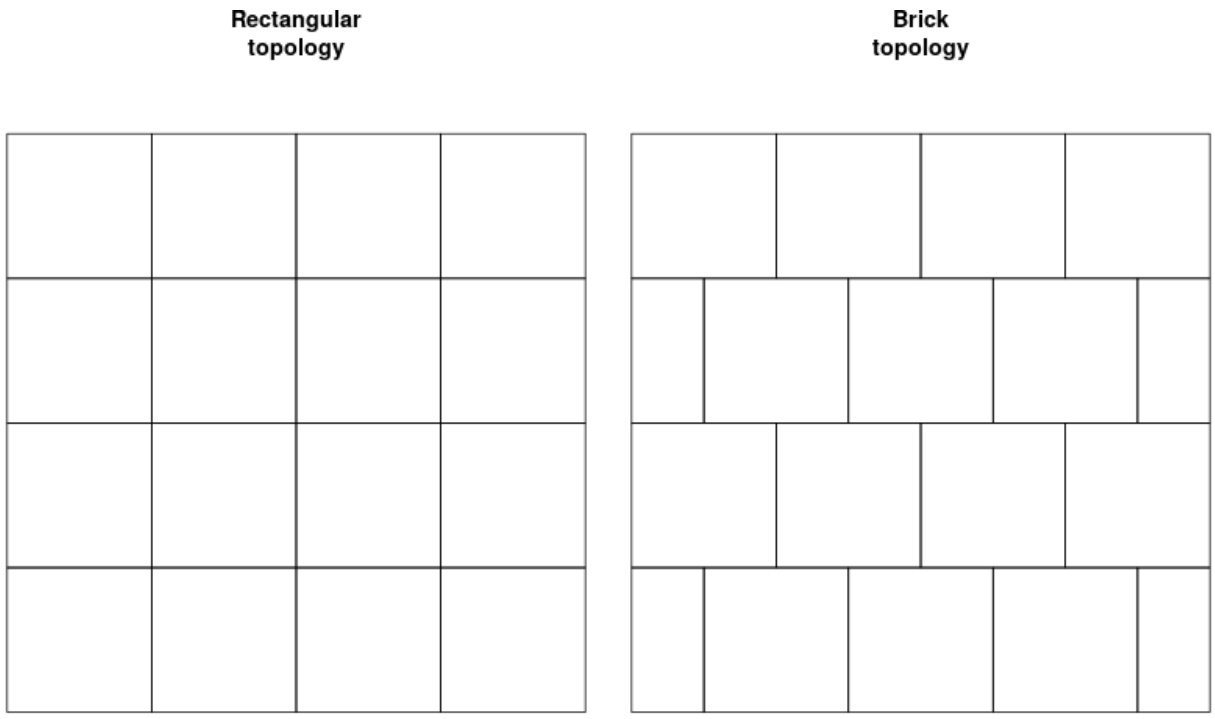

Figure 28: Two possible topologies for a grid of motifels: rectangular (4-connected grid) and brick (6-connected grid)

This representation is simple and useful in many cases. However, it also has significant shortcomings for task such as segmentation. Due to its 4-connectivity, segmentation based on the rectangular topology often results in a large number of small (usually one-motifel size) segments. To solve this issues, the 6-connected brick topology was created (right panel in Figure 28). Square motifels are arranged in alternative layers, where each layer is shifted by a half of size of output motifel.

The `gpat_gridhis` module creates objects only in the rectangular (4-connected grid) topology. However, `gpat_segment` uses the brick (6-connected grid) topology during its segmentation procedure as a default. (Note: it is possible to use the rectangular topology in the segmentation procedure by adding the *-q* flag). It this process, neighboring motifels are merged to create the brick wall topology (Figure 29). Importantly, this influence the size and shift of motifels and therefore change a size of analyzed area.

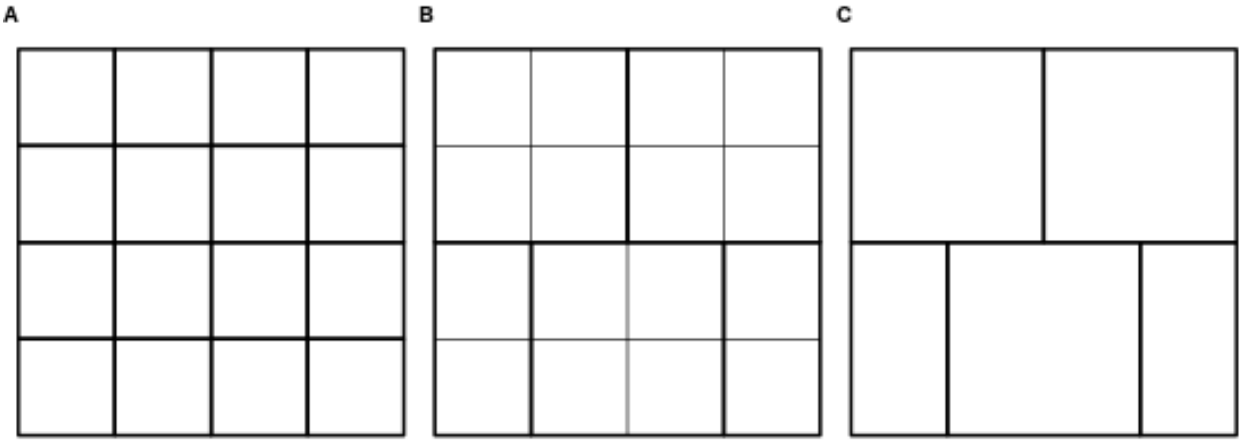

Figure 29: Process of merging elements of the rectangular topology to the brick topology: (A) input grid in the rectangular topology, (B) neighboring motifels in the rectangular topology are merged to create the brick wall topology, (C) output grid in the brick topology

For example, we created a new grid using `gpat_gridhis` with the size and shift arguments set to 100. This means that each motifel encapsulates pattern in an area of 9 sq km (one cell size is 30 meters; 30 meters x 100 = 3 km; 3 km x 3km = 9 sq km). However, the segmentation module uses the brick topology as a default. Thus, neighboring motifels are merged and create a new "brick" motifels. This time each motifel encapsulates pattern in an area of 36 sq km (9 sq km motifels are combined into quadruples). This means that if we want to segment patterns in scale of 9 sq km, we need to create a new grid of the size and shift twice smaller - 50.

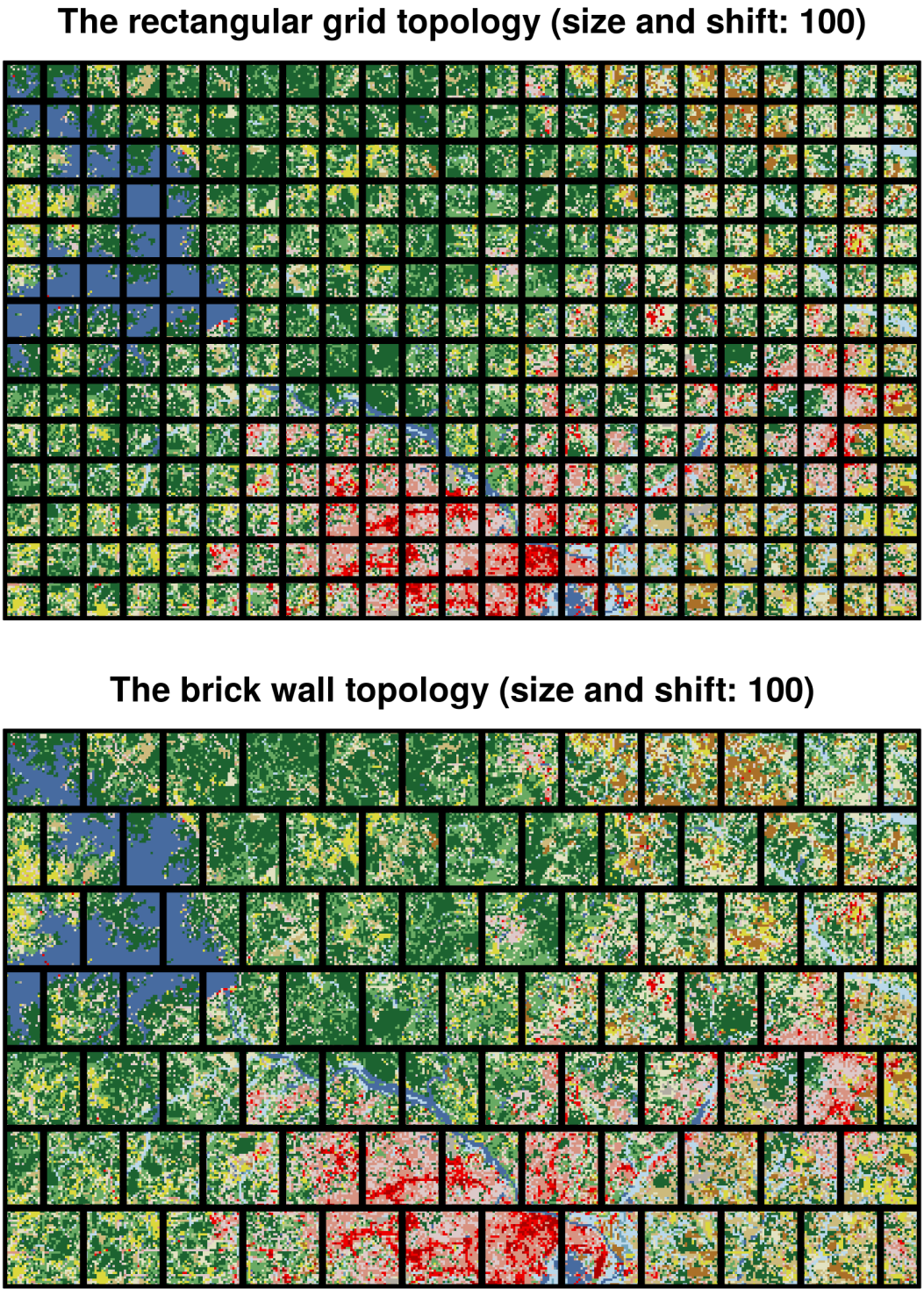

Figure 30: Comparision between the rectangular grid topology and the brick topology for Augusta

References

[1] J. Niesterowicz and T. F. Stepinski, "Regionalization of multi-categorical landscapes using machine vision methods", *Applied Geography*, vol. 45, pp. 250–258, Dec. 2013, ISSN: 01436228.

[2] G. Câmara, R. C. M. Souza, U. M. Freitas, and J. Garrido, "Spring: Integrating remote sensing and gis by object-oriented data modelling", *Comput. Graph.*, vol. 20, no. 3, pp. 395–403, May 1996, ISSN: 00978493. DOI: 10.1016/0097-8493(96)00008-8.

[3] J. Li and R. Narayanan, "Integrated spectral and spatial information mining in remote sensing imagery", *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 673–685, 2004.

[4] T. Blaschke, "Object based image analysis for remote sensing", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, pp. 2–16, 2010, ISSN: 09242716.

[5] J. Jasiewicz, P. Netzel, and T. F. Stepinski, "Landscape similarity, retrieval, and machine mapping of physiographic units", *GEOMORPHOLOGY*, vol. 221, pp. 104–112, 2014, ISSN: 0169555X.

[6] R Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available: https://www.R-project.org/.

[7] R. J. Hijmans, S. E. Cameron, J. L. Parra, P. G. Jones, and A. Jarvis, "Very high resolution interpolated climate surfaces for global land areas", *International journal of climatology*, vol. 25, no. 15, pp. 1965–1978, 2005.

[8] E. A. Williams and E. A. Wentz, "Pattern Analysis Based on Type, Orientation, Size, and Shape", *Geogr. Anal.*, vol. 40, no. 2, pp. 97–122, Apr. 2008, ISSN: 0016-7363. DOI: 10.1111/j.1538-4632.2008.00715.x.

[9] M. Barnsley and S. Barr, "Inferring urban land use from satellite sensor images using kernel-based spatial reclassification", *Photogrammetric engineering and remote sensing*, vol. 62, no. 8, pp. 949–958, 1996.

[10] P. Chang and J. Krumm, "Object recognition with color cooccurrence histograms", in *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, June 23-25, 1999*, IEEE Computer Society Conference, 1999.

[11] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification", *Syst. Man Cybern. IEEE Trans.*, vol. 3, no. 6, pp. 610–621, Nov. 1973, ISSN: 0018-9472. DOI: 10.1109/TSMC.1973.4309314.

[12] T. K. Remmel and F. Csillag, "Mutual information spectra for comparing categorical maps", *International Journal of Remote Sensing*, vol. 27(7), pp. 1425–1452, 2006.

[13] J. Niesterowicz and T. F. Stepinski, "On using landscape metrics for landscape similarity search", *Ecological Indicators*, vol. 64, pp. 20–30, 2016.

[14] K. McGarigal, "Fragstats help: Documentation for fragstats 4", 2014.

[15] J. Lin, "Divergence measures based on the Shannon entropy", *IEEE Transactions on Information Theory*, vol. 31(1), pp. 145–151, 1991.

[16] J. Jasiewicz, T. Stepinski, and P. Netzel, "Content-based landscape retrieval using geomorphons", in *Geomorphometry 2013*, Nanjing, China, 2013, pp. 1–4.

[17] T. Stepinski, P. Netzel, and J. Jasiewicz, "LandEx - A GeoWeb tool for query and retrieval of spatial patterns in land cover datasets", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7(1), N. Xiao, M.-P. Kwan, and H. Lin, Eds., pp. 257–266, 2014.

[18] P. Netzel and T. F. Stepinski, "Pattern-based assessment of land cover change on continental scale with application to NLCD 2001-2006", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53(4), pp. 1773–1781, 2015.

[19] S. Cha, "Comprehensive survey on distance/similarity measures between probability density functions", *Int. J. Math. Model. Methods Appled Sci.*, vol. 1(4), no. 4, pp. 300–307, 2007.

[20] J. Jasiewicz, P. Netzel, and T. F. Stepinski, "Landscapes similarity, retrieval, and machine mapping of physiographic units", *Geomorphology*, vol. 221, pp. 104–112, 2014.

[21] P. Jaccard, " Nouvelles recherches sur la distribution florale", *Bull. Soc. Vaudoise Sci. Nat.*, vol. 44, pp. 223–270, 1908.