
biota Documentation

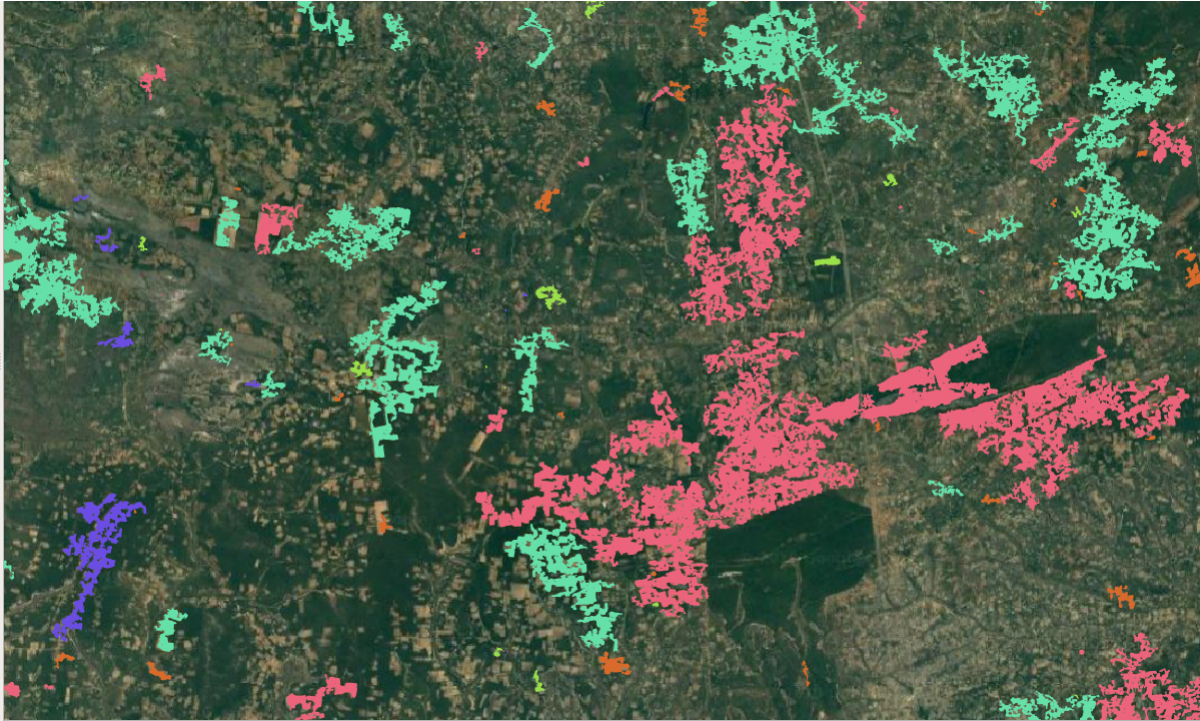
Release 1

Samuel Bowers

Jun 15, 2021

CONTENTS

1	Who do I talk to?	3
2	Contents:	5
3	Indices and tables	15



`acacia` is a Python tool that produces clustered polygons from two rasters. If this seems like a general statement, it's because it is. Although originally designed to take input from our other tool `biota`, `acacia` works the same regardless of the data you feed it. All you need to run `acacia` are:

- One raster of a given continuous quantity (e.g. biomass in 2010), representing the initial state of your data.
- One raster representing the change between your initial state and the final state of your data (e.g. change in biomass between 2010 and 2015).

Using these 2 rasters, `acacia` produces polygons of significant negative change, clusters them according to their properties (both geometrical and taken from the rasters) and outputs a shapefile summarizing all these analyses, as well as a few handy figures.

`acacia` offers the possibility of adding a third input in the form of a shapefile, but we'll come to it later.

WHO DO I TALK TO?

Written and maintained by Guillaume Goodwin (g.c.h.goodwin@sms.ed.ac.uk) and Samuel Bowers (sam.bowers@ed.ac.uk).

CONTENTS:

2.1 Setup instructions

2.1.1 Preamble

acacia is written in Python and was developed on a Linux platform. Provided you can use Python and can install modules on your computer, it should be possible to run on most OS or a Linux server.

On this page, we explain how to set up your Linux or Windows machine to use *acacia*.

Note: *acacia* requires Python 3. It may work with Python 2.7, but it's best not to try your luck.

2.1.2 Installing Anaconda

We recommend running *acacia* in Anaconda.

If you are using a Linux machine, open a terminal window, change directory to the location you'd like to install Anaconda Python, and run the following commands:

```
wget https://repo.anaconda.com/archive/Anaconda3-2018.12-Linux-x86_64.sh
bash Anaconda3-2018.12-Linux-x86_64.sh
```

Once complete, you'll need to add this version of Python to your `.bashrc` file as follows (replacing `~/` with your installation directory):

```
# Substitute root for the path to your system's installation and .bashrc file.
echo 'export PATH="$~/anaconda3/bin:$PATH"' >> ~/.bashrc
exec -l $SHELL
```

If this has functioned, on executing `python` in a terminal window, you should see the following:

```
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>>
```

For Windows users, go to the [Anaconda website](#)) and download the installer for your version of Windows (64-bit or 32-bit). Once the download is finished, run the installer. This may take a while, but when it is done you will be able to open the *Anaconda Prompt*, which functions like a linux terminal and has Anaconda conveniently installed.

2.1.3 Setting up your Anaconda environment

To ensure you are working with the appropriate version of Python as well as the correct modules, we recommend that you create an Anaconda virtual environment set up for running *acacia*. This is done by running the following commands in your terminal or the *Anaconda prompt* (recommended procedure):

```
conda create -n acacia -c conda-forge python=3.7 tqdm scikit-learn scikit-image pyshp ↵  
↵ gdal
```

Note: the GDAL package is notoriously temperamental. If this step fails, try again and add `openssl=1.0` at the end of the line

Activate the *acacia* environment whenever opening a new terminal window by running this command:

```
conda activate acacia
```

Note: Remember to activate the *acacia* environment whenever you want to use *acacia*.

If you are SURE you won't use anything else than *acacia*, you can do without virtual environments. In this case, just type:

```
conda install -c conda-forge python=3.7 tqdm scikit-image pyshp gdal
```

Acacia is primarily a Graphical User Interface. to use the *acacia* GUI, you need an extra package called *PyQt5*. To install it, type:

```
pip install pyqt5
```

2.1.4 Installing acacia

acacia does not require installation apart from what you've already done. Congratulations, you are ready to use *acacia*!

2.2 Graphical User Interface

acacia comes in as a Graphical User Interface (GUI). This is the preferred option to use *acacia*, but of course if you are handy in Python feel free to extract the script from the file *Acacia.py* and make it your own.

2.2.1 Using the acacia Graphical User Interface (GUI)

The *acacia* Graphical User Interface (GUI) allows you to use *acacia* with minimal usage of the command line. In your terminal or the *Anaconda prompt* (if you're using Windows), navigate to the folder where you stored *acacia* by typing:

```
cd /full/path/to/acacia/
```

This folder should have a file called `Acacia.py`.

To open the GUI, type:

```
python Acacia.py
```

The GUI window should then open. Now you can use *acacia* several times with minimal typing.

2.2.2 The acacia window

2.3 How does acacia work?

2.3.1 Making polygons

acacia takes values from both input rasters to make polygons of change. In the figure below, the square on the left represents the initial data, and the square on the right represents the change data. If the value of cells in the initial data exceed 10 (Quantity threshold), the cells are selected. However, if less than 5 contiguous cells are selected (Area threshold), then the cells will be discarded. Likewise, if the value of the cells in the change data are high than -3 (Change threshold) and the cells were originally selected in the initial data, then the cells are selected. If less than 2 contiguous cells are selected (Area change), the remaining cell is discarded. The final selected cells are grouped to form polygons (3 polygons in the example below). Each polygon will be given a set of properties based on the boxes you ticked before running the script.

2.3.2 Making clusters

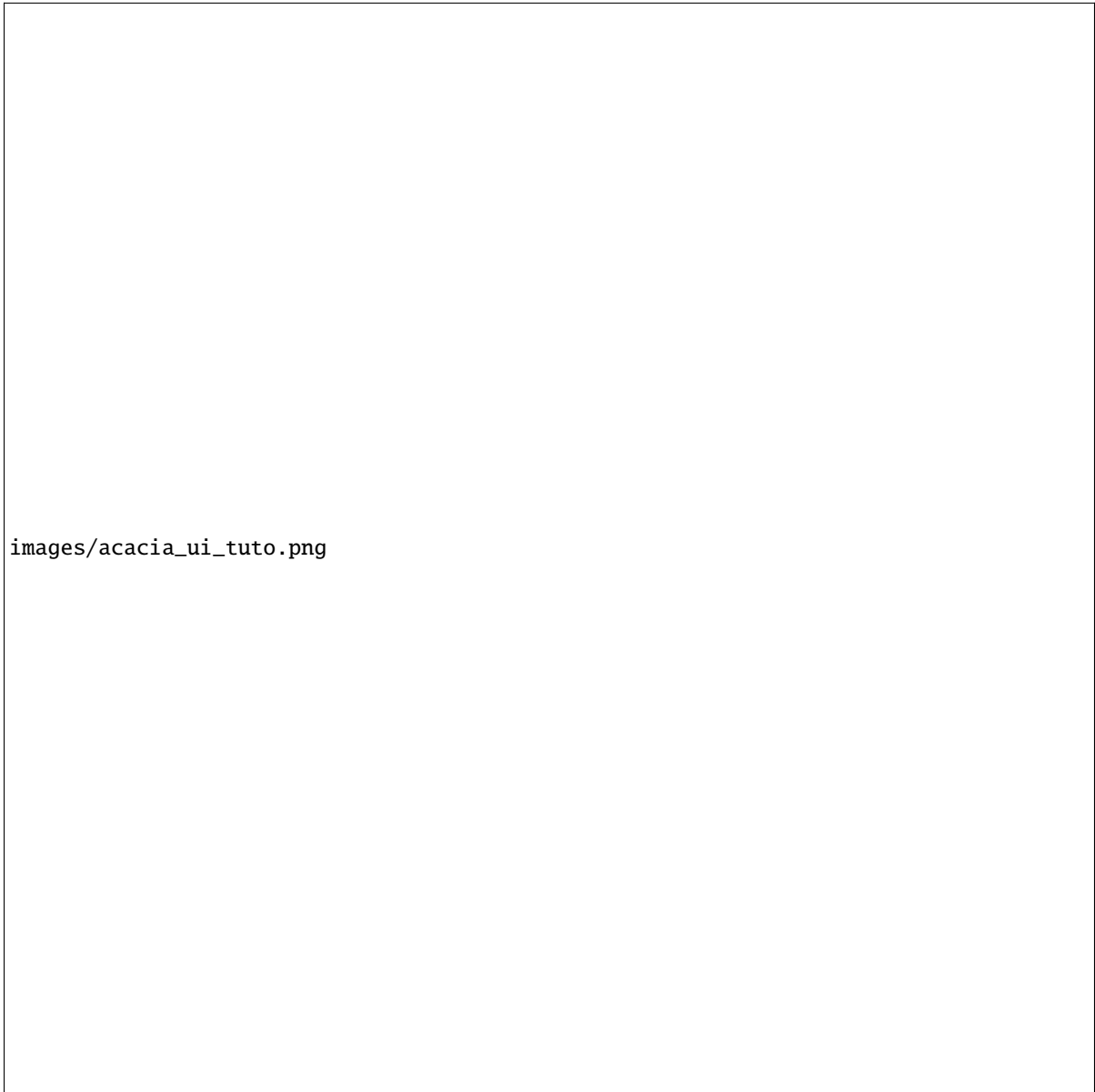
acacia uses a hierarchical clustering method. Based on the properties you selected, it will build a distance matrix between all the polygons. It will then select a separation distance to determine between 5 and 10 classes of polygon while ensuring maximum separation between the classes using the `Davies-Bouldin index` <https://en.wikipedia.org/wiki/Davies-Bouldin_index>`_`.

The figures below show examples of clusters and their properties

2.3.3 Using your field data

acacia offers the possibility to compare your field data to the data from remote sensing. If you have entered a file for validation data, then *acacia* will extract properties from your data and compare it to the polygons. It will then fit your data in the most likely class.

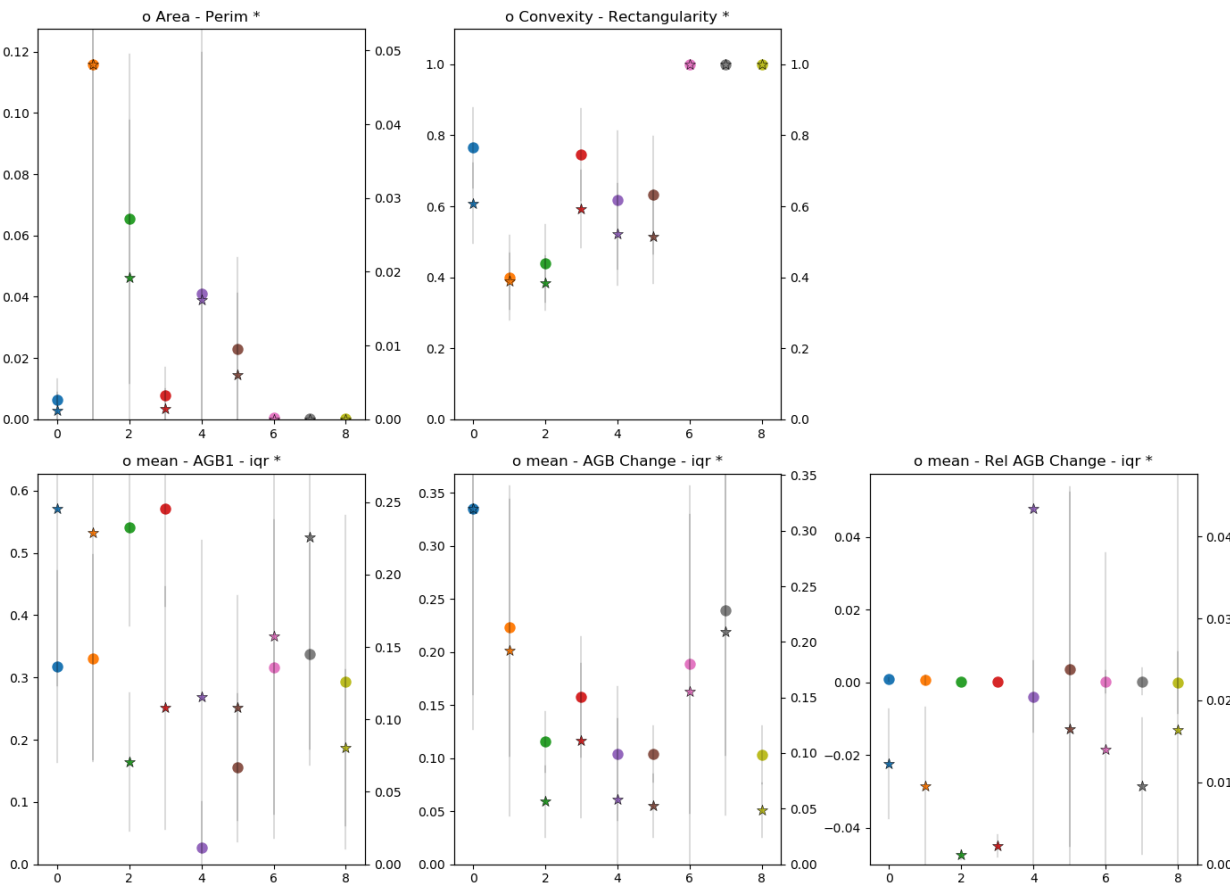
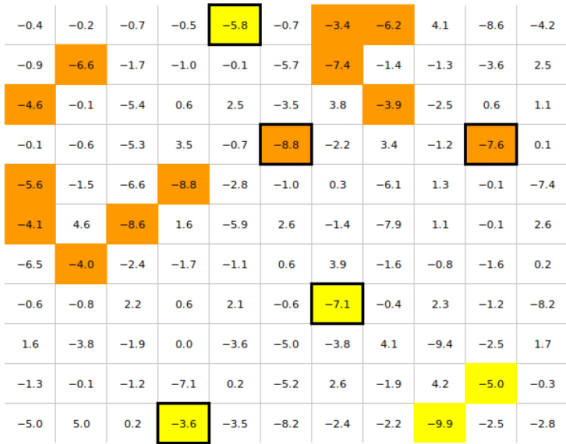
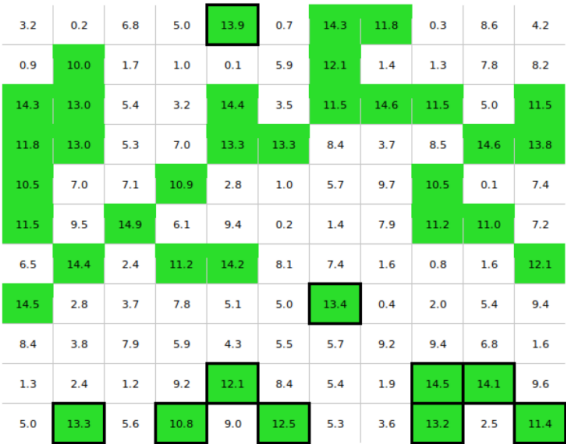
The figures below show examples of clusters and their properties compared to field data.

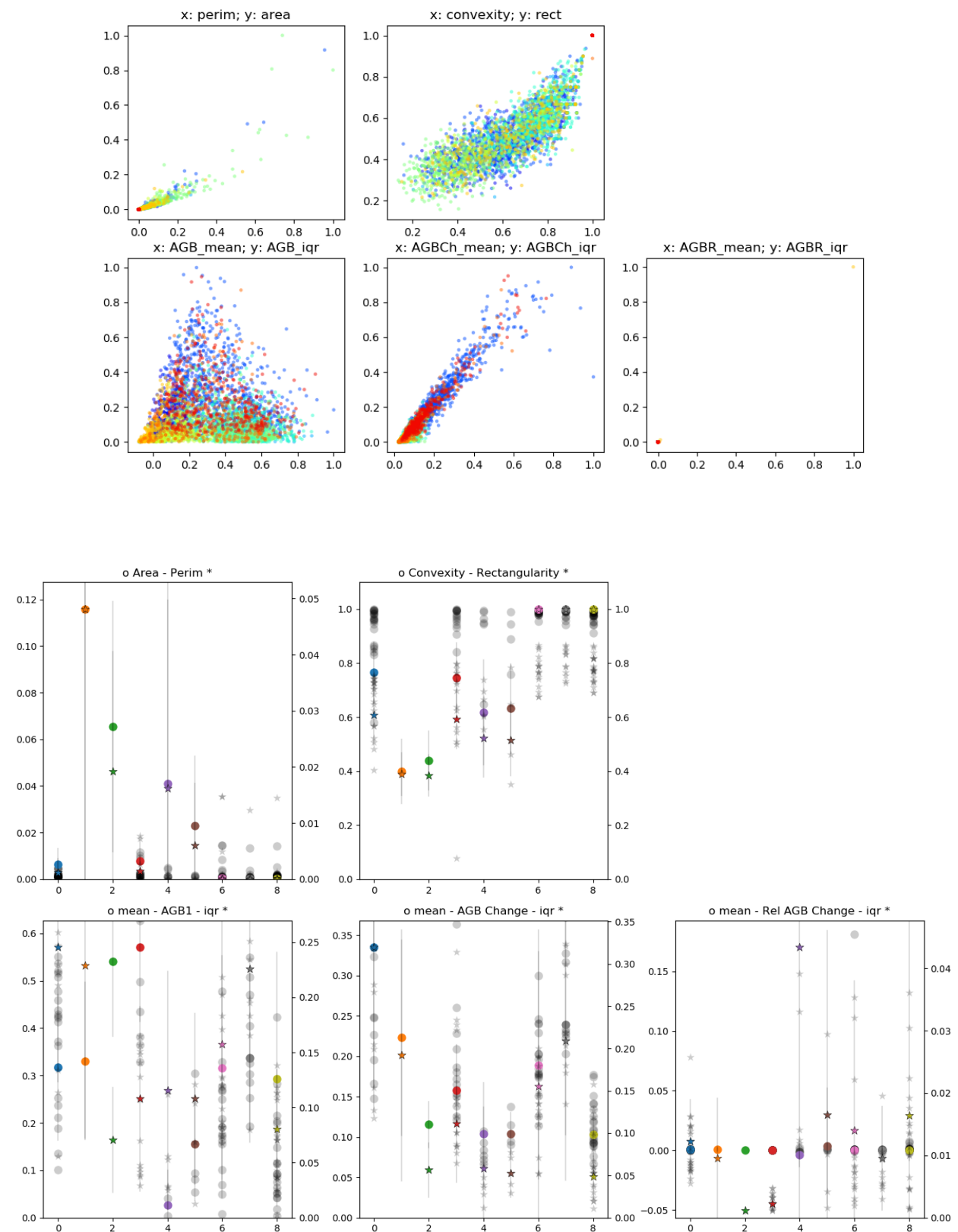


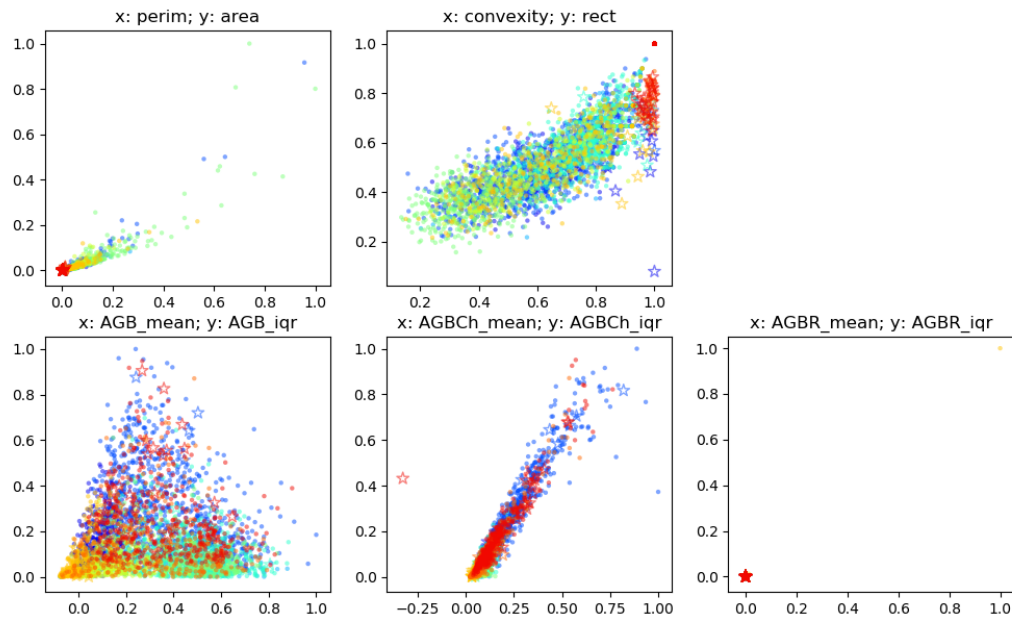
images/acacia_ui_tuto.png

Selected cells (Quantity threshold = 10)
Discarded cells (Area threshold = 5)

Unselected cells (Area threshold = 5)
Selected cells (negative Quantity change = 3)
Discarded cells (Area change = 1)







2.4 Worked example

In this worked example we will try out *acacia* on merged tiles of biomass and biomass change in North-West Zambia. These files are located in the `data/` folder by default.

2.4.1 Preparation

First ensure that you've followed the setup instructions successfully.

Open a terminal, and use `cd` to navigate to the location where you've stored *acacia*. Then open the GUI as shown below:

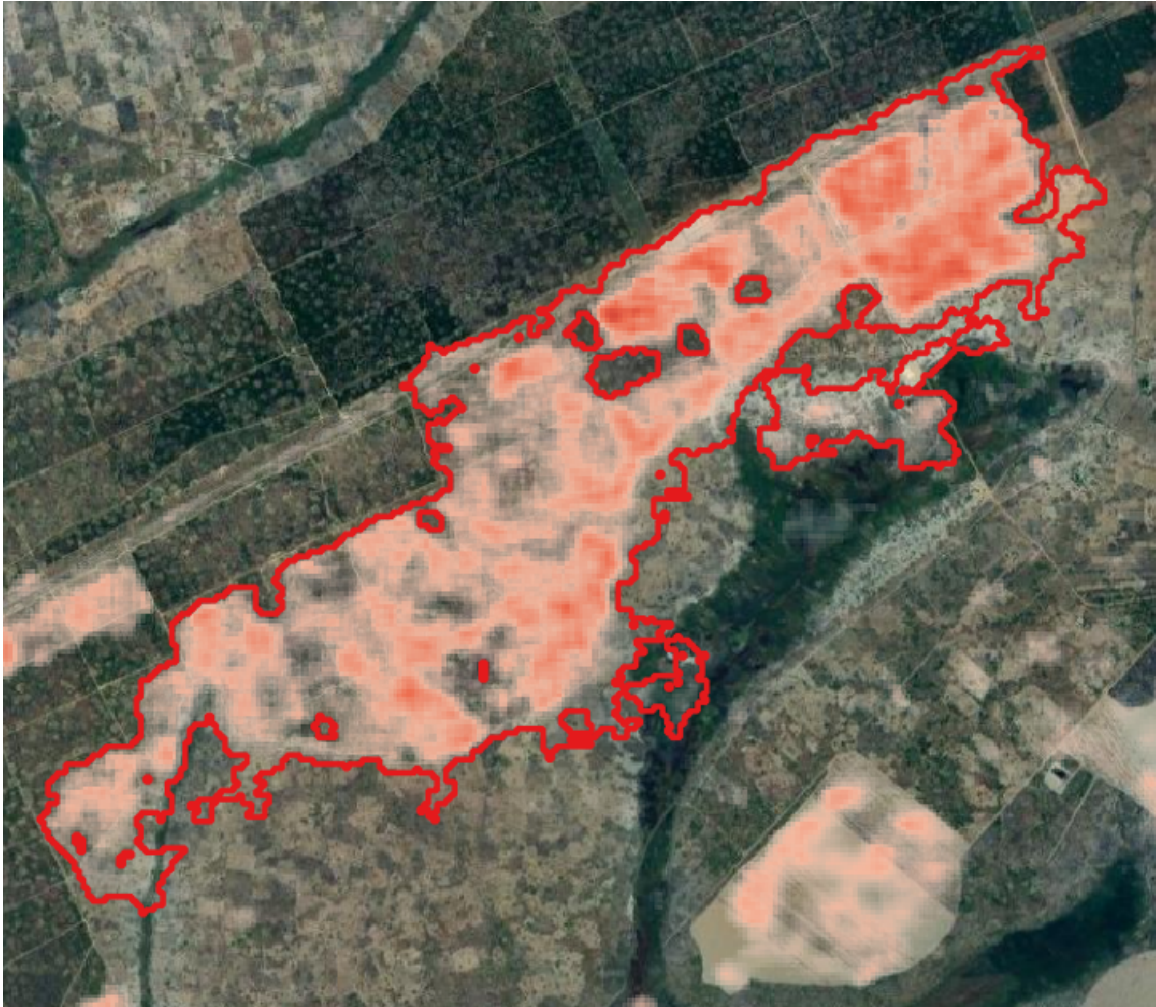
```
cd /path/to/acacia/
python Acacia.py
```

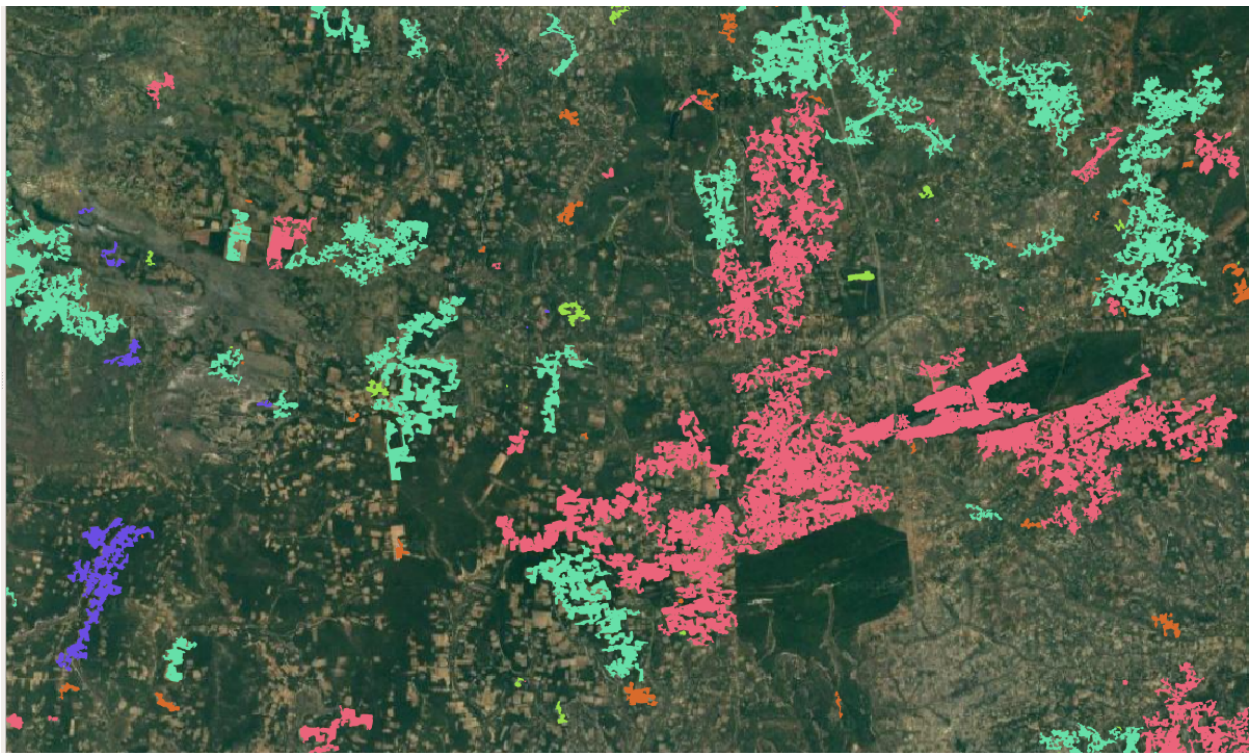
You should now see the *acacia* window on your screen.' By leaving all your parameters by default, you will obtain about 100 classified polygons. You can then load these polygons into QGIS and check the values of change within them, as shown below:

You can also visualise the clusters by choosing a classified colour scheme for the polygons according to the field 'cluster'.

You should see something like this (but not exactly this, the subsampling is random).

Note: *acacia* subsamples the polygons of change by default to avoid crashing your computer. If you are confident your machine can take it, increase the number of elements into the thousands!





INDICES AND TABLES

- `genindex`
- `modindex`
- `search`