Natural capital map creation: python code user guide

Alison Smith, Environmental Change Institute, 8 March 2020

# Introduction

These rough notes describe how to use the set of python scripts developed by Alison Smith and Martin Besnier at the Environmental Change Institute, University of Oxford, for creating natural capital maps based on ecosystem service scores. The scripts are research tools and have not been rigorously tested or documented.

The scripts are designed to work with OS Mastermap (OSMM) as the base map. They perform the following steps:

1. Merge in habitat data – either Phase 1 habitat data or National England Priority Habitats
2. Merge in CROME data to assign agricultural fields as either arable crops or improved grassland (this replaces the previous of CEH Land Cover Map to do this in the LCM/PHI option, as the CROME data is freely available and more up to date).
3. Merge in agricultural land class
4. Merge in site designations (e.g. nature reserves)
5. Merge in OS green space maps
6. Merge in footpath buffers and assign public accessibility information.
7. Join the base map to a table of ecosystem service scores
8. Analyse the impacts of different land use change scenarios by adding up scores within potential development zones.

These notes need to be merged with the ‘First steps’ document which is more up to date (includes using CROME instead of Land Cover Map) but only covers the first few steps (before merging ALC).

# Instructions for running the code

1. All the scripts should be in the same folder as the MyFunctions.py script, which supplies useful common functions.
2. Download the geodatabase version of OSMM topographic area from Edina Digimap or OS (assuming you have a license). Do not use a shapefile version, as field names get truncated and code does not work.
3. For each stage of the process, open the script (in IDLE or PyCharm), edit it to set the workspace directory and all the other parameters, and run the code. (In PyCharm, right click on the script tab and select ‘Run..script name’. Caution – if you just select ‘Run’ from the box in the top right, make sure the correct script is showing in the dropdown next to the box otherwise it will just run the last script that you ran). Check the outputs carefully. Depending on what stage you are at, the outputs may need to be copied to a new geodatabase for the next stage of processing. When you are happy with the final outputs, go back and delete the intermediate geoprocessing files to free up space. (One of the final scripts now does this, keeping just the important intermediate steps.)

# Creating maps for Oxfordshire

The code is set up to work with either the Phase 1 habitat and land use (HLU) map for Oxfordshire, provided under license from the Thames Valley Record Centre (TVERC), or the Natural England Priority Habitat data which is freely available. However, it should be possible to use any similar datasets by modifying the input parameters for the main merge script, and possibly modifying the pre- and post-processing steps to work with the habitat classifications in the alternative datasets (if they differ).

The most straightforward way to merge together two or more habitat datasets would be though a series of Intersect or Identity operations in ArcGIS. However, the habitat datasets we work with tend to have boundaries that do not exactly match OS Mastermap boundaries. Therefore this method, when performed at county scale, creates tens of thousands of tiny ‘slivers’ at polygon edges, which can cause problems when attempting to perform subsequent geoprocessing operations.

The novel aspect of the code (designed originally by Martin Besnier, a visiting researcher from the Université Paris Sud) is that it merges two polygon datasets (a base map and a set of new features), while avoiding the creation of an excessive number of slivers due to boundary mismatch. It does this through identifying polygons in the base map that need to be split to match the boundaries of the new features, while ignoring minor differences in the boundaries due to inaccurate mapping (which would generate slivers). The output is faithful to the base map boundaries as far as possible, though minor differences (just a few cm) may arise during one of the sliver elimination steps.

A sequence of scripts have been developed which:

1. pre-process the OSMM and habitat input maps (e.g. to remove duplicate and overlapping features);
2. merge the two polygon datasets together without generating too many slivers;
3. generate a unified ‘Interpreted habitat’ attribute field that uses a set of rules to identify the habitat based on information from both OSMM and the habitat map;
4. The code is set up to work for two main options:
   1. Phase 1 habitat data is available
   2. Phase 1 habitat data not available – use Natural England Priority Habitat data instead, together with CROME or CEH Land Cover Map to assign agricultural land into arable or improved grassland.

We currently run the scripts for Oxfordshire first, using Phase 1 data, and then run for the rest of the Arc using PHI / CROME / LCM data.

## 1a. Using Phase 1 habitat data

1. Check habitat data (e.g. Oxfordshire Phase 1 HLU) for ‘unidentified’ habitats, and use BAP or other info to determine what they are – otherwise delete them (OSMM definition will be used instead). Check for incorrect spellings in Phase1habitat field (duplicates with upper and lower case can be left in, as they should be resolved by the code). Check that habitat names still match those used in the code; if they have changed, modify the code accordingly.
2. Open the **Merge\_OSMM\_HLU\_ Preprocess.py** script:

* Set the workspace gdb path in the code (NB - it is faster to run this on your hard drive rather than a network server). The code currently expects a geodatabase called Merge\_OSMM\_HLU.gdb.
* Depending on what steps you want to run, change the flags in the parameter section to True or False. If the files have not been clipped to the exact boundaries, set ‘clip\_to\_boundary’ to True, name the input files HLU\_in and OSMM\_in, add boundary file to the gdb and set the name of the boundary feature class (e.g. “Oxfordshire”) as a parameter in the code. Otherwise the code expects the input files to be named HLU and OSMM (this can be changed in the parameter section).
* Run the script. This erases manmade features and water from HLU, removes overlaps and eliminates slivers, checks and repairs geometry; removes overlapping features (landforms and pylons) from OSMM; and removes unnecessary fields (enter the list of habitat fields to keep in the script: currently POLY\_ID, PHASE1HABI, BAP\_HABITA, "BAP\_HABI00", "SITEREF", "COPYRIGHT", "VERSION").

1. **Merge\_into\_Base\_MapV5a.py:** open the script, set the merge\_type parameter to “OSMM\_HLU”, check the workspace and all the other parameters and run the code. This can take about 14 hours for merging HLU habitat data into OSMM for the whole of Oxfordshire.
2. **OSMM\_HLU\_Interpret.py**: set ‘region’ to ‘Oxon’. This adds the habitat interpretations, by combining OSMM and Phase 1 appropriately. Note: If OSMM is "undefined" this usually means the area is under development or scheduled for development. The ‘undefined\_or\_original’ flag allows the user to choose whether to map these areas as "undefined" or as the current / original habitat pre-development.

## 1b. Using CEH Land cover map and NE PHI data

(These scripts not yet uploaded to GitHub)

1. All scripts run in a loop through Local Authority Districts (LADs) (leaving out Oxfordshire LADs as we do that separately).
2. **Prep\_OSMM.py** – copies tiles from download folders into a single folder, merges into a single feature class in a geodatabase, and clips to LAD boundaries.
3. **Setup\_LAD\_gdbs.py**. Has several stages – set any that are not needed to ‘False’. At present it is set to ignore Oxfordshire as we have a separate procedure for generating the Oxfordshire data, using Phase 1 habitat.
   1. Prepare PHI data. There are three separate datasets – the main PHI data, plus Wood Pasture and Parkland (WPP) and Open Mosaic Habitats on previously developed land (OMHD). For all three datasets, dissolve on habitat field (Main\_habit or PRIHABTXT), convert to single part, delete polygons <10m2, and copy habitat field to new field called ‘PHI’, ‘WPP’ or ‘OMHD’. Then union all three datasets. WPP and OHMD are not very accurate, e.g. WPP maps whole parkland areas with a mix of habitats including grass, fields, woods, buildings and plantations, and often overlaps with other PHIs. So we let OSMM woodland, water, manmade and other PHI habitats take priority.
   2. Set up individual geodatabases for each LAD:
      1. Create new gdb
      2. Copy in the OSMM data for this LAD
      3. Create boundary feature
      4. Clip Arc-wide LCM and PHI data to the boundaries of the LAD, and copy in.
4. **OSMM\_HLU\_Interpret.py** – set ‘region’ to ‘Arc’, and ‘simplify\_OSMM’ to True but other stages to False. This will interpret a simplified habitat from the OSMM Make, Descriptive Group and Descriptive Term.
5. **METHOD 1 (intersect PHI). Run Arc\_LCM\_PHI.py**  - combines with CEH land cover map and Natural England priority habitat data. Set ‘merge\_or\_intersect’ = “intersect”. There are two main stages:
   1. Process\_LCM: Add data from CEH Landcover Map 2015 by defining agricultural land or ‘general surface natural’ as either arable or improved grassland. Then create a joint ‘Interpreted habitat’ field that assigns either the OSMM habitat or the LCM habitat.
   2. Process\_PHI: delete landforms from OSMM data, intersect (using ‘Identity’) with PHI data, then interpret PHI by copying to Interpreted Habitat but only when OSMM is not manmade, garden or water. For WPP and OMHD, keep fine detail of other habitats and only copy these designations across for generic habitats (e.g. farmland). PHI boundaries match OSMM quite well, though with some genuine split polygons, so the intersect (Identity) doesn not create too many slivers, e.g. Aylesbury goes from 455 single part slivers (<1m2) for OSMM\_LCM to 2372 after intersect with PHI, which is not too bad out of approx. 300,000 polygons. However, if time, it is neater to use the alternative method (step 6).
6. **METHOD 2 (merge PHI) ALTERNATIVE to step 5:** Merge rather than intersecting PHI, to reduce slivers. For example, this only increases slivers from 455 to 457 (single part) for Aylesbury. There are relatively few PHI polygons so the Merge does not take too long – about 50 minutes per LAD, 12 hours for all LADs (except Oxfordshire). To do this,
   1. run **Arc\_LCM\_PHI.py** with ‘merge\_or\_intersect’ = “merge” and ‘step = 1’. This should set ‘intersect\_PHI’ and ‘interpret\_PHI’ to False but all other steps to True,
   2. run **Merge\_into\_Base\_Map.py** with merge-type set to ‘Arc\_LCM\_PHI’,
   3. **re-run Arc\_LCM\_PHI.py** with ‘step = 2’. This should set ‘process\_PHI’ and ‘interpret\_PHI’ to True and all other stages to False, to copy the correct habitat interpretation across from the PHI, WPP and OMHD fields to the Interpreted\_habitat field.
7. Check habitat data (e.g. Oxfordshire Phase 1 HLU) for ‘unidentified’ habitats, and use BAP or other info to determine what they are – otherwise delete them (OSMM definition will be used instead). Check for incorrect spellings in Phase1habitat field (duplicates with upper and lower case can be left in, as they should be resolved by the code).
8. Open the **Merge\_OSMM\_HLU\_ Preprocess.py** script:

* Set the workspace gdb path in the code (NB - it is faster to run this on your hard drive rather than a network server). The code currently expects a geodatabase called Merge\_OSMM\_HLU.gdb.
* Depending on what steps you want to run, change the flags in the parameter section to True or False. If the files have not been clipped to the exact boundaries, set ‘clip\_to\_boundary’ to True, name the input files HLU\_in and OSMM\_in, add boundary file to the gdb and set the name of the boundary feature class (e.g. “Oxfordshire”) as a parameter in the code. Otherwise the code expects the input files to be named HLU and OSMM (this can be changed in the parameter section).
* Run the script. This erases manmade features and water from HLU, removes overlaps and eliminates slivers, checks and repairs geometry; removes overlapping features (landforms and pylons) from OSMM; and removes unnecessary fields (enter the list of habitat fields to keep in the script: currently POLY\_ID, PHASE1HABI, BAP\_HABITA, "BAP\_HABI00", "SITEREF", "COPYRIGHT", "VERSION").

1. **Merge\_into\_Base\_MapV5.py:** open the script, set the merge\_type parameter to “OSMM\_HLU”, check the workspace and all the other parameters and run the code. This can take about 14 hours for merging HLU habitat data into OSMM for the whole of Oxfordshire.
2. **OSMM\_HLU\_Interpret.py**: set ‘region’ to ‘Oxon’. This adds the habitat interpretations, by combining OSMM and Phase 1 appropriately. Note: If OSMM is "undefined" this usually means the area is under development or scheduled for development. The ‘undefined\_or\_original’ flag allows the user to choose whether to map these areas as "undefined" or as the current / original habitat pre-development.
3. **Merge\_ALC.py:**  set ‘region’ to ‘Oxon’. Adds ALC attributes to the input file, for farmland only.
4. **Union\_Designations.py:** set ‘region’ to ‘Oxon’. Combines multiple designation feature classes or shapefiles into a single layer – need to dissolve out superfluous internal boundaries (e.g. in green belt).
5. **Process\_Designations.py:** sets up the attributes for the combined designation file. Set the appropriate path for the file.
6. COPY TO NEW GEODATABASE if you want to avoid over-writing the intermediate files
7. **Merge\_into\_Base\_MapV5.py:** set merge\_type to “Designations”. This merges the designations data with the base map.
8. **Join\_Greenspace.py** - set ‘region’ to ‘Oxon’. Combines with OS green space and OS OpenGS (and maybe Orval parks later)
9. **Public\_access.py** – set ‘region’ to ‘Oxon’. Assigns public access
10. **SetUpScoreTable.py** set ‘region’ to ‘Oxon : Assigns natural capital scores and multipliers to the base map.
11. **Set up mxd**. I did not manage to get this automated, so use the previously created mxd (OxNatCap\_*date*.mxd) and link to the correct NatCap scores feature class. The individual layers for the 18 ecosystem services are already set up and can be turned on or off. They all link to the NatCap scores feature class, but with the symbology set to use the correct attribute for the ES.
    1. Note: to get the symbology for each ES working, copy NatCap\_Oxon to a new layer, right click on the layer and select Properties, go to the Symbology tab, click on the Import button, navigate to the Layer\_symbologies folder and select **‘NatCap.lyr’**, and set the field in the dropdown box to the appropriate ES attribute (see below). You can’t just change the field directly in the Properties tab dropdown, because the boundaries of the scores for each colour shade will change.
    2. Make sure you use the attribute for the ES after appropriate multipliers have been applied, not just the raw scores. So you need to display **Food\_ALC\_norm, Aesthetic\_norm, Nature\_desig, Education\_desig, Sense\_desig, Rec\_access**, not the raw scores Food, Aesthetic, Nature, Education, Recreation, SensePlace.
12. **ExportNatCapMaps.py** – does not work yet! Have to do manually. Intended to set up the .mxd to display the natural capital maps and then export jpegs for each ecosystem service map. To display each ES map manually, turn the extra layers for rivers, paths and hedges either on or off as well as the main ES layer. For the maximum and average total score maps, food can be displayed as a separate layer (in orange) to the other services (in green), as requested by stakeholders.

# Creating maps for the rest of the Arc – use CROME / CEH LCM and NE PHI data

1. All scripts run in a loop through LADs (leaving out Oxfordshire LADs as we do that separately). Note: need to use ListWorkspaces not ListFiles, otherwise it does not seem to re-set the variable feature classes properly between loops.
2. **Prep\_OSMM.py** – copies tiles from download folders into a single folder, merges into a single feature class in a geodatabase, and clips to LAD boundaries.
3. **Setup\_LAD\_gdbs.py**. Has several stages – set any that are not needed to ‘False’. At present it is set to ignore Oxfordshire as we have a separate procedure for generating the Oxfordshire data, using Phase 1 habitat.
   1. Prepare PHI data. There are three separate datasets – the main PHI data, plus Wood Pasture and Parkland (WPP) and Open Mosaic Habitats on previously developed land (OMHD). For all three datasets, dissolve on habitat field (Main\_habit or PRIHABTXT), convert to single part, delete polygons <10m2, and copy habitat field to new field called ‘PHI’, ‘WPP’ or ‘OMHD’. Then union all three datasets. WPP and OHMD are not very accurate, e.g. WPP maps whole parkland areas with a mix of haibtats including grass, fields, woods, buildings and plantations, and often overlaps with other PHIs. So let OSMM woodland, water, manmade and other PHI habitats take priority.
   2. Set up individual geodatabases for each LAD:
      1. Create new gdb
      2. Copy in the OSMM data for this LAD
      3. Create boundary feature
      4. Clip Arc-wide LCM and PHI data to the boundaries of the LAD, and copy in.
4. **OSMM\_HLU\_Interpret.py** – set ‘region’ to ‘Arc’, and ‘simplify\_OSMM’ to True but other stages to False. This will interpret a simplified habitat from the OSMM Make, Descriptive Group and Descriptive Term.
5. See the First Steps document for a more up to date version of the next steps, merging in CROME and PHI together
6. **METHOD 1 (intersect PHI). Run Arc\_LCM\_PHI.py**  - combines with CEH land cover map and Natural England priority habitat data. Set ‘merge\_or\_intersect’ = “intersect”. There are two main stages:
   1. Process\_LCM: Add data from CEH Landcover Map 2015 by defining agricultural land or ‘general surface natural’ as either arable or improved grassland. Then create a joint ‘Interpreted habitat’ field that assigns either the OSMM habitat or the LCM habitat.
   2. Process\_PHI: delete landforms from OSMM data, intersect (using ‘Identity’) with PHI data, then interpret PHI by copying to Interpreted Habitat but only when OSMM is not manmade, garden or water. For WPP and OMHD, keep fine detail of other habitats and only copy these designations across for generic habitats (e.g. farmland). PHI boundaries match OSMM quite well, though with some genuine split polygons, so the intersect (Identity) doesn not create too many slivers, e.g. Aylesbury goes from 455 single part slivers (<1m2) for OSMM\_LCM to 2372 after intersect with PHI, which is not too bad out of approx. 300,000 polygons. However, if time, it is neater to use the alternative method (step 6).
7. **METHOD 2 (merge PHI) BETTER ALTERNATIVE to step 5:** Merge rather than intersecting PHI, to reduce slivers. For example, this only increases slivers from 455 to 457 (single part) for Aylesbury. There are relatively few PHI polygons so the Merge does not take too long – about 50 minutes per LAD, 12 hours for all LADs (except Oxfordshire). To do this,
   1. run **Arc\_LCM\_PHI.py** with ‘merge\_or\_intersect’ = “merge” and ‘step = 1’. This should set ‘intersect\_PHI’ and ‘interpret\_PHI’ to False but all other steps to True,
   2. run **Merge\_into\_Base\_Map.py** with merge-type set to ‘Arc\_LCM\_PHI’,
   3. **re-run Arc\_LCM\_PHI.py** with ‘step = 2’. This should set ‘process\_PHI’ and ‘interpret\_PHI’ to True and all other stages to False, to copy the correct habitat interpretation across from the PHI, WPP and OMHD fields to the Interpreted\_habitat field.
8. **Merge in RPA CROME Crop map**, as this is better than CEH LCM for farmland, and is also freely available.
9. **Merge\_ALC.py**: Depending on whether merge or intersect was used in the previous step, **c**hange the name of the input file to either OSMM\_LCM\_PHI\_merge or OSMM\_LCM\_PHI\_intersect and set ‘region’ to ‘Arc’.
10. **UnionDesignations.py** and **ProcessDesignations.py** - set ‘region’ to ‘Arc’.
11. **Merge\_into\_Base\_Map.p**y – set merge\_type to “Arc\_Desig”.
12. **Join\_Greenspace.py** – as above – set ‘region’ to ‘Arc’. Adds OSGS and openGS.
13. **Public\_access.py** – as above – set ‘region’ to ‘Arc’. Adds access info from CROW, OSGS and openGS, Orval paths and parks.
14. **SetUpScores.py:** set ‘region’ to ‘Arc’
15. UDM scenarios from Ali Ford’s group at Newcastle – Convert Ascii to Raster, then Raster to Polygon.
16. SpatialStrategyAnalysis.py – can apply for UDM scenarios – set region to “Arc”. Compares natural capital scores within each scenario. Also additional metrics - areas of highest value assets, length of hedges, number of ancient trees. Writes output to a text file that can be imported to Excel for analysis / charts.
17. Design ‘green vision’. Draft NRN networks not yet available – Paul Leinster wants to keep them as ‘fuzzy’ areas, not precise. Can that number of houses be integrated without impinging on network integrity?

# Methodology notes and explanations

# Merging OSMM and HLU

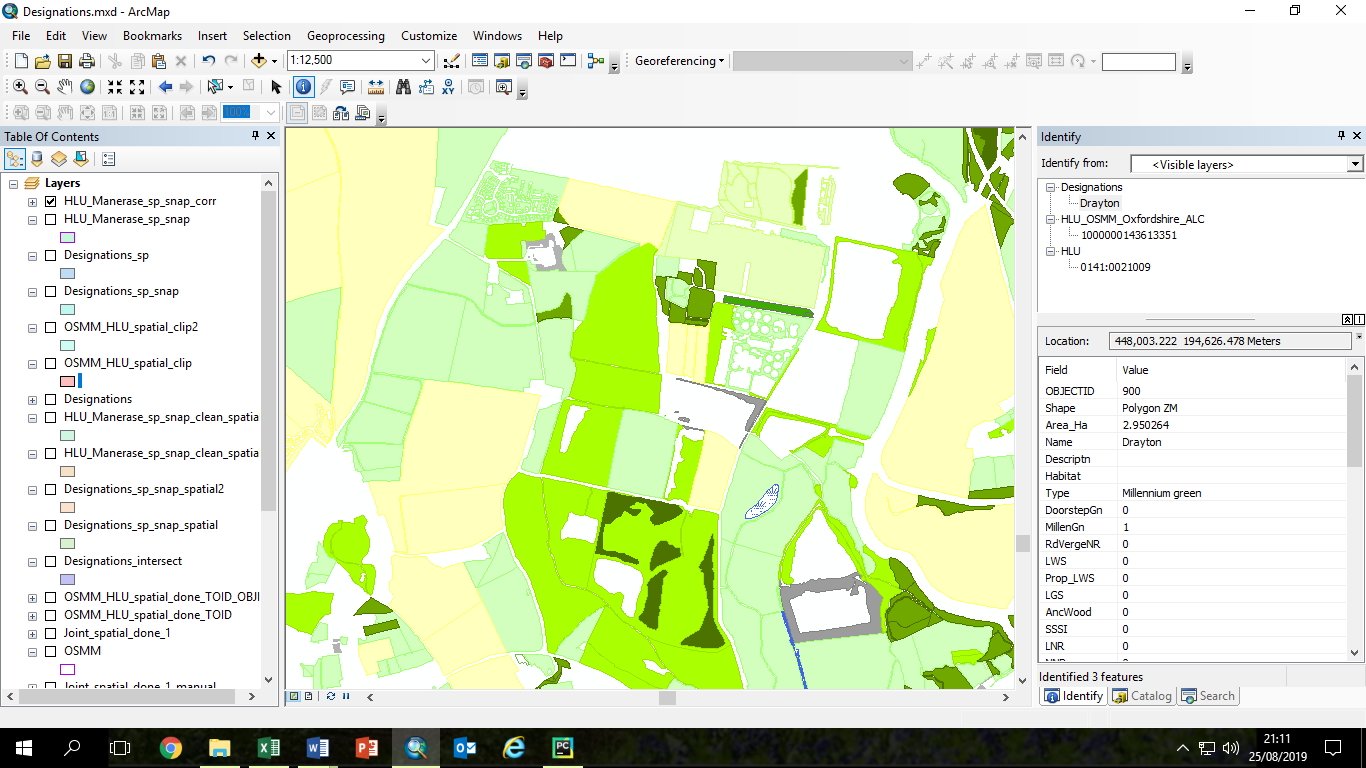
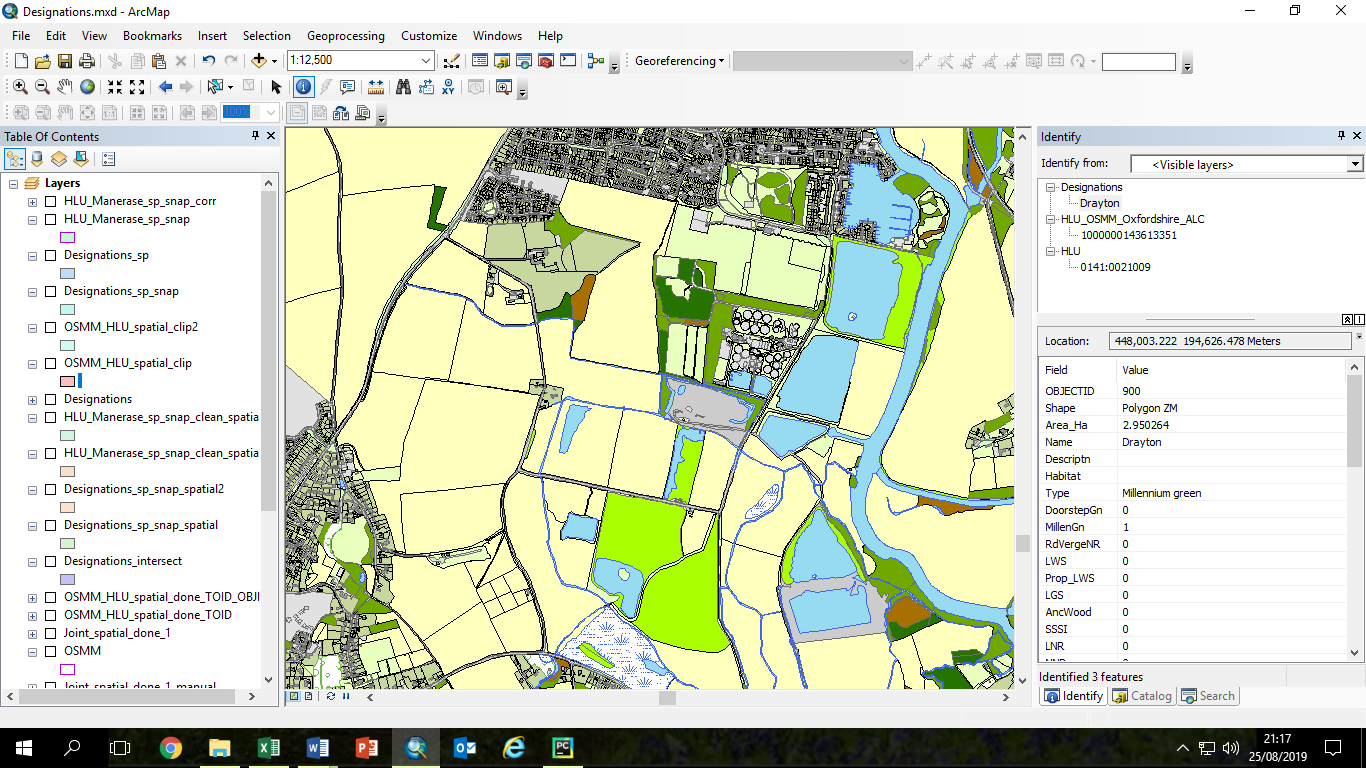
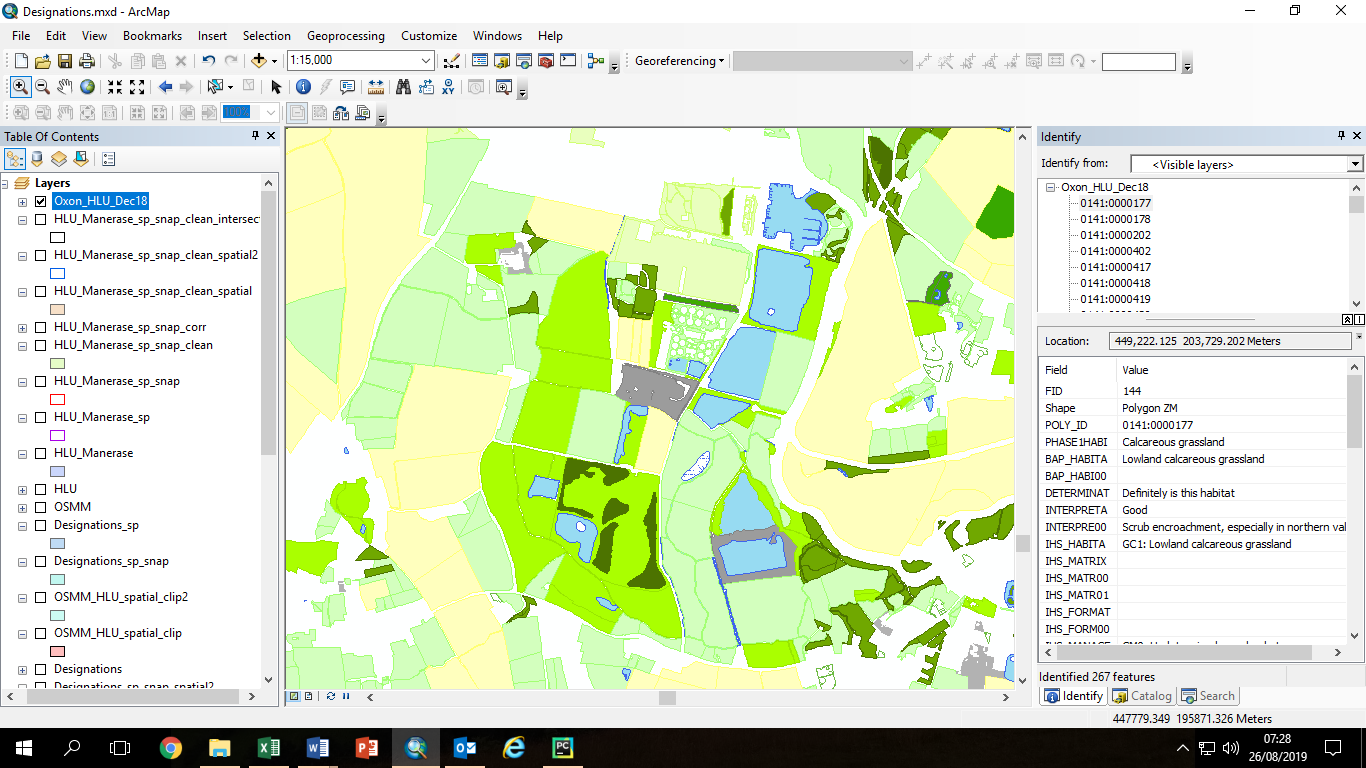
The method is based around a ‘tabulate intersection’ step to determine the percentage of new feature polygons within base map polygons.

## Prepare datasets

**Merge\_OSMM\_HLU\_ Preprocess.py**

**Delete OSMM landforms; delete HLU overlaps; erase manmade and water from HLU.**

1. Optional stage: clip to exact boundary of study area.
2. Optional stage: delete unnecessary attribute fields.
3. Delete landforms from OSMM -> OSMM\_noLandform NOTE: Pylons also overlap – originally these were not removed but I changed this as it was causing problems later in the code.
4. Delete HLU water features so we don’t get remnants left over later which obscure OSMM islands.
5. Erase OSMM manmade (buildings, roads, tracks) and water from HLU -> HLU\_Manerase Roadside is not erased, as this misses some semi-natural grassland and scrub etc.
6. Correct HLU overlaps with Union. **Note: this code may crash at the Union stage – if this happens, the union can be done manually in ArcGIS and the code can be re-started.**
7. HLU multipart to singlepart
8. Eliminate HLU overlap slivers (less than 1m2 – was originally also OR length/area >3 but I have removed this as I am not convinced it does not delete useful polygons.) This stage also deletes the small sliver gaps.
9. Delete remaining HLU stand-alone slivers
10. Delete remaining gaps (i.e. those that are genuine gaps, not slivers).
11. Delete identical shapes. If there are duplicate shapes with different habitats (there are a few of these in the Oxfordshire dataset) only one of these will be kept.



Examples from the South Abingdon / North Drayton area. Left: Original HLU layer omits urban areas and some water features. Middle: OSMM includes urban areas and woodland detail but not type of grassland. Right: pre-processed HLU (HLU\_Manerase) with manmade and water features erased, revealing shapes of lake and new housing estate.

## Main merge code – merge\_into\_base\_map\_V5a.py

First set up the input parameters in the first block of code. The steps in the main code are described below.

The code is designed to loop through a set of geodatabases, which we used for the different Local Authority Districts within the Oxford-Cambridge Arc. However it also works with only one geodatabase (which we used for Oxfordshire).

### Snap new features (e.g. HLU) to match base map features (e.g. OSMM) better

1. Only snap polygons that are not already identical (though only around 2000 out of 80000 HLU polygons are identical to OSMM after the pre-processing).
2. Snap new features to base map. Snap parameters are set up in the parameter input section. We use small snap distances (0.5 or 1m) to avoid distorting the edges of small linear features such as rides.
3. Correct overlaps created by the snapping process, by unioning and then deleting identical polygons.
4. Clean slivers from the snapped new features, by:
   1. Eliminating slivers less than 1m2
   2. Deleting stand-alone slivers <1m2

This is the most time-consuming part of the code – it can take 12 hours for a county the size of Oxfordshire. We experimented with densifying the two datasets before the snap, but this does not always work well and makes the snap operation even longer. Also it loses the original curves, replacing them with a series of short straight lines, and results in very large datasets (with far more points than previously).

### Decide which OSMM polygons should be split to include new HLU features

1. Save OBJECTID attribute and polygon areas to new attribute fields, for both the base map (Base\_ID) and the new features (New\_ID). These saved IDs will be used later, to transfer attributes across to the new merged shapes.
2. Use ArcGIS Tabulate intersection function to create a table with the percentage overlaps between the base map and the new features.
3. Create a new ‘Relationship’ attribute in the Tabluate Intersection table (“Base\_TI”) and use this to store the decision of whether or not to split polygons, based on a set of rules. Small overlaps are ignored, but larger overlaps mean that the base map (OSMM) polygon will be split (intersected with the new feature polygon outlines). However very large overlaps mean that the polygon will not be split, as the whole polygon will be assumed to match the new feature. The input spatial parameters ignore\_low and ignore\_high determine the threshold overlaps for ignoring or splitting polygons. There is also an overlap size threshold (significant\_size), above which polygons will be split even if the percentage overlap is small.
   1. Very small overlap, less than ignore\_low (default 5%) AND overlap\_area < significant size (default 200 m2): ignore the new feature (habitat will remain same as base map). Relationship attribute set to ‘Base’.
   2. Overlap between ignore\_low and ignore\_high (default 5-95%): split the polygon so that the new feature outline is included within the base map polygon. Relationship attribute set to ‘Split’.
   3. Very large overlap, greater than ignore\_high (default 95%) – set the whole polygon to the same habitat as the new feature. Relationship attribute set to ‘New’.

**Examples: OSMM outlines in black and HLU outlines in red (selected HLU polygons in Cyan).**

|  |
| --- |
| **Example 1:** The large HLU arable polygon (Cyan outline) contains 4 medium-sized OSMM fields plus a very small OSMM field. The four medium OSMM fields and the small field all fit exactly into the HLU polygon so there is no need to split them. |
| **Example 2:** The HLU polygon (Cyan outline) is between 5% and 95% of the OSMM polygon, so the OSMM polygon will be split. |
| **Example 3. HLU (main area); OSMM (smaller area)**  The HLU woodland polygon (Cyan outline) is smaller than the OSMM scrub / woodland polygon but this is due to inaccurate mapping of the boundary. The overlap is less than the 95% threshold, so the whole polygon is assigned to the HLU habitat, and it is not split unless any of the smaller parts are greater than the ‘significant size’ threshold (default 200m2). |

|  |
| --- |
| **Example 4:**  **Top part**: a HLU ‘improved grassland’ field is split into a number of smaller OSMM areas including an area of scattered trees. These are all completely within the HLU field so they are not split. The small block of woodland and the small field next to it are assigned the OSMM classification later.  **Lower part**: an OSMM field is split into two HLU fields and includes a patch of OSMM trees in the north half which is not in HLU, and a HLU marsh in the lower half that is not included in OSMM.  The top half of the field is interpreted as the HLU neutral grassland. The small woodland will become OSMM woodland. The lower part of the field and the marsh are also assigned the HLU habitat types. |

### Add new HLU features to the OSMM base map (combine geometry)

We start from a clean copy of OSMM and modify it to include the new feature (HLU) polygons where they are justifiably different from OSMM – in other words the rows marked ‘Split’ in the tabulated intersection tables.

We have to be very careful here, because each base map (OSMM) polygon can have multiple corresponding rows in the Tabulate Intersection table, if the overlaps with the new features are complex. So some of the table joins are ‘one to many’ joins, which we have to be very careful with.

1. Make a copy of the Tabulate\_Intersection table, selecting only the rows for the polygons to be split.
2. For the polygons that we do not want to split (because they are over 95% overlapped by a new feature polygon), we need to make sure that the only the habitat attributes from the largest overlapping new feature are included (not any smaller ones round the edges). So make another copy of the Tabulate Intersection table for the un-split polygons, sort it by size, and delete those with a duplicate Base\_ID (this is what we set up earlier, by copying the original ObjectID). This keeps only the details for the main new habitat within that base map polygon.
3. Make a clean copy of the base map (OSMM) and add a ‘Relationship’ attribute field.
4. Join this to the table of split polygons, based on the Base ID, and set ‘Relationship’ to ‘Split’ for all those rows. This identifies all the base map (OSMM) polygons that need to be split.
5. Make a clean copy of the new features (HLU) and join it to the table of split polygons, using the new feature ID (New\_ID). This identifies all the new feature polygons that will form part of the new outlines in the merged file.
6. Select the new feature polygons marked ‘split’, and clip them using the outline of the base map polygons to be split. This will produce a set of new shapes to be merged into the base map, which we call ‘Joint\_spatial\_clip’.
7. Union these new shapes (which may be only partial, e.g. half of an OSMM field) into the base map polygons that they are splitting. This produces the new shapes for the split polygons, with both the base map attributes and (for the parts that are split), the new feature attributes.
8. Caution: This is the brain-melting bit. There are now two copies of the Relationship field - one from the unioned base map, with all rows 'split', and one from the clipped new features, with some polygons marked 'split' and some either null or blank. Some of the split polygons may have a large part that was tagged 'new' (because the overlap percentage was very high but there were also smaller parts of the same polygon that were tagged for splitting because they exceeded the significant size). So we now want to match the non-split parts of these polygons (joined in via the Union) with the correct ID from the TI table, so that attributes can be transferred later. We do this by sorting both the TI tables and the unioned clip file by size, and then joining the tables so that the split polygon parts are matched with the intersections of the same size, and copying the ID across.
9. Clean the clipped shapes, by eliminating and deleting slivers. This is the step where eliminating slivers may lead to small deviations from the integrity of the base map boundaries, including small polygons (< 1m2) in the original base map.
10. Another complicated bit. Create a tag ‘Not new’ to identify the parts of the split polygons that are base map only, i.e. not part of the new features. This is needed when transferring the new feature attributes back into the merged shapes, in step 13.
11. Erase the base map with the new joint shapes, and then merge the new shapes back into the base map. This has created a version of the base map that has polygons split where necessary to match the new features.
12. Convert to single part, check and repair geometry.

### Transfer attributes from OSMM and HLU to OSMM\_HLU layer

1. The merged dataset contains the base map attributes for all polygons, and the polygons that were split also contain the new feature attributes. However we now need to add the new feature attributes for the polygons that were not split but were marked as ‘new’, via a table join. This is a one to many join - each base polygon could be overlapped by more than one new feature polygon, but as these polygons were not marked for splitting, we want to use the ID of the largest intersection (i.e. ignoring slivers and insignificant areas) so we sort by size. The table join is done in two stages, because Relationship <> 'Not new' excludes NULLs.
2. The next few steps are a bit complicated but the end result is that all the attributes for the new features have been copied across to the appropriate polygons in the merged dataset.

### Interpreting data: OSMM\_HLU.py

This script assigns an overall habitat classification to each polygon as follows.

1. The OS mastermap attributes Descriptive Group, Descriptive Term and Make are used to assign new attribute field ‘OSMM\_hab’.
2. The Phase 1 habitat data (from the HLU habitat dataset) is used to assign new attribute field ‘HLU\_hab’.
3. The habitat based on a combination of OSMM and HLU is stored in ‘Interpreted Habitat’
4. The BAP classification (also from the HLU habitat dataset) modifies this where applicable, resulting in the final habitat interpretation in ‘BAP\_interpretation’.

It uses a fairly complex set of rules. It would need to be adapted if there are habitats present that we have not included, e.g. heather and bog.

# Merge with CROME

Rural Payments Agency CROME crop map is better than Phase 1 for farmland, and is also freely available, unlike CEH LCM. For example, the north part of Blenheim Park is mapped as arable in Phase 1 habitat data (based on aerial photos from around 2005) but correctly as grassland in CROME. CROME is composed of small hexagons (40m sides), but it is relatively accurate to transfer the data to OSMM or OSMM-HLU merged polygons, by dissolving CROME on ‘field’ (simplified land use description), tabulating the intersections, sorting by descending order of size and then joining the tabulated table to the base OSMM map. The sort by size is to ensure that only the largest polygon is joined (as it is a one-to-many join). Non-agricultural polygons are excluded and there is a minimum 30% overlap threshold.

Examination for Oxon shows only two single CROME polygons for short roatation coppice, and both look like mis-classifications so ignore. Similarly, ignore ‘Perennial crops or isolated trees’ CROME as some areas are actually grassland. Also ‘Fallow’ looks more like arable than grassland in most cases (from Google Earth and Bing maps) even though it is classed as ‘grass’ in the CROME simplified categories.

Examination of the ten largest polygons (and then some further randomly sampled ones) classed as arable in CROME but improved grassland in Phase 1 shows that CROME is usually correct.

To insert CROME belatedly to the full natural capital map, I created an input dataset containing only the attributes up to where we want to insert the CROME data, by selecting the attributes we want to keep (from properties, fields) and then exporting. Then we will have to re-run all the subsequent stages.

Copy CROME across only for agricultural land. Do not include ‘Natural land’ as that is mainly roadside and urban amenity grass.

# Merge with ALC

Pre-process ALC if not already done, by:

1. clipping to area (Oxfordshire oval)
2. dissolving (without multipart polygons) to get rid of odd-looking large arc-shaped lines
3. union with gaps allowed and tolerance of 1m, to get rid of internal slivers.

Set up geodatabse containing:

1. OSMM\_HLU layer
2. ALC layer (pre-processed)

The Merge\_OSMM\_HLU\_ALC code does the following steps:

* 1. Extracts farmland from OSMM\_HLU by selecting Simplified\_Interpretation LIKE ‘Arable%’ OR Simplified \_Interpretation = ‘Agricultural land’ OR Simplified \_Interpretation LIKE ‘Improved%’. Note: could use BAP\_Interpretation but this will leave out coastal and floodplain grazing marsh and lowland meadows, some of which is actually intensive farmland? Could also add intensive orchards?
  2. Runs ‘Identity’ with the extracted farmland as input and ALC as identity layer. Output must be a geodatabase feature otherwise field names will get truncated. Do not specify a tolerance distance, as this will mean the layers no longer match when you append the farmland layer back in. Specify ‘no FID’.
  3. Exports the OSMM\_HLU layer to a new layer (in the geodatabase) called ‘OSMM\_HLU\_Oxfordshire\_noFarmland’. Selects all the farmland rows (as above) and deletes them.
  4. Appends the noFarmland layer to the Farmland\_ALC identity layer.

Note: Identity performs an intersect. This creates some slivers, but not many – number of slivers increases from 1645 in OSMM\_HLU merged file to 1819 after the Identity with ALC, so 174 extra slivers. (later note: but there are quite a few larger slivers along the edge of fields – would be good to get rid of them at some stage). Could snap before merging but probably not worth it. Some agricultural polygons are not assigned an ALC value but these are just small slivers outside the ALC dataset (e.g. in urban or other non-agricultural areas). These could disappear with a snap. Or could eliminate slivers – but OSMM boundaries then might change, leaving gaps when the agricultural areas are merged back in to the main OSMM file.

Then (in Set Up Scores code):

1. Import the ALC multiplier table and join to the HLU table.
2. Join the HLU table to the matrix of scores. It could be a good idea to now re-export the joined table to a new dataset so it does not rely on joins.
3. Add an extra column for the adjusted score and populate with the score multiplied by the ALC multiplier using ‘calculate field’. But **only for arable and improved grassland** (using select by attributes) – for the others, set to same as basic food score.
4. Normalise: add a new field ALC\_norm and calculate as (score x ALC multiplier) x 10/30.3

# Merge with designations

We currently use 17 habitat designation datasets. All have very inaccurate boundaries that do not follow OS Mastermap. We first Union all the designation layers into a single file (following a detailed procedure) and then merge this file with the base map. The unioned designation layer will only need to be updated if one of the input datasets changes.

## Creating a combined designations file

1. **Union\_Designations.py**
2. **Preprocess\_designations.py**

We combined the following habitat designations into a single layer. For Oxfordshire we include local designations from TVERC, but when mapping the whole Arc we use only the freely available national datasets. When mapping the Arc we also include RAMSAR sites (there are none in Oxfordshire).

**Freely available national datasets**

* 1. AONBs
  2. National Nature Reserves (and proposed NNRs?)
  3. Local Nature Reserves (and proposed LNRs?)
  4. SSSIs
  5. Special Areas of Conservation (SACs) (there are no SPAs in Oxfordshire)
  6. Ancient Woodland
  7. Country Parks
  8. Millennium Greens
  9. Doorstep Greens
  10. Green belt land (note: there was a projection error in the 2017-2018 dataset on data.gov.uk so we are using the 2016-2017 dataset instead)
  11. National Trust Open Access land
  12. Scheduled ancient monuments
  13. Historic parks and gardens
  14. World Heritage Sites

**Local Oxfordshire datasets from TVERC (not used for the Arc)**

* 1. Local Geological Sites
  2. Proposed Local Geological Sites
  3. Local Wildlife Sites
  4. Proposed Local Wildlife Sites
  5. Road verge nature reserves

**Designations present in the OxCam Arc but not in Oxfordshire**

* 1. RAMSAR sites and proposed RAMSAR sites
  2. SPAs and proposed SPAs

**In other areas such as Northern Powerhouse**

* 1. SCIs?
  2. Heritage Coasts

Note that the Oxfordshire Conservation Target Areas (CTAs), which will be an important component of future Nature Recovery Networks (NRNs), were not included as designations because they represent opportunities for habitat restoration rather than signifying existing high-value habitats. They are treated as an ‘opportunity layer’ instead, which can be superimposed on the natural capital map later to help see where opportunities for natural capital enhancements might align with the CTAs / NRNs.

### Combine the shapes.

First manually create a table (DesignationFiles.dbf, imported from DesignationFiles.xlsx) containing a list of the datasets to merge (see next page). The code reads through this table and processes each dataset in turn.

**Union\_Designations.p**y

There is an optional first step to copy all the original shapefiles into geodatabase feature classes.

The raw datasets contain a varied set of attributes. We want to extract only four attributes: the name of each area (if provided), description (if provided), habitat (if applicable) and type of designation. The DesignationFiles.dbf table maps the raw attribute names in each input dataset to these four attributes in the new combined dataset. The first part of the code sets up these fields in each of the input datasets, and deletes any fields that are not needed.

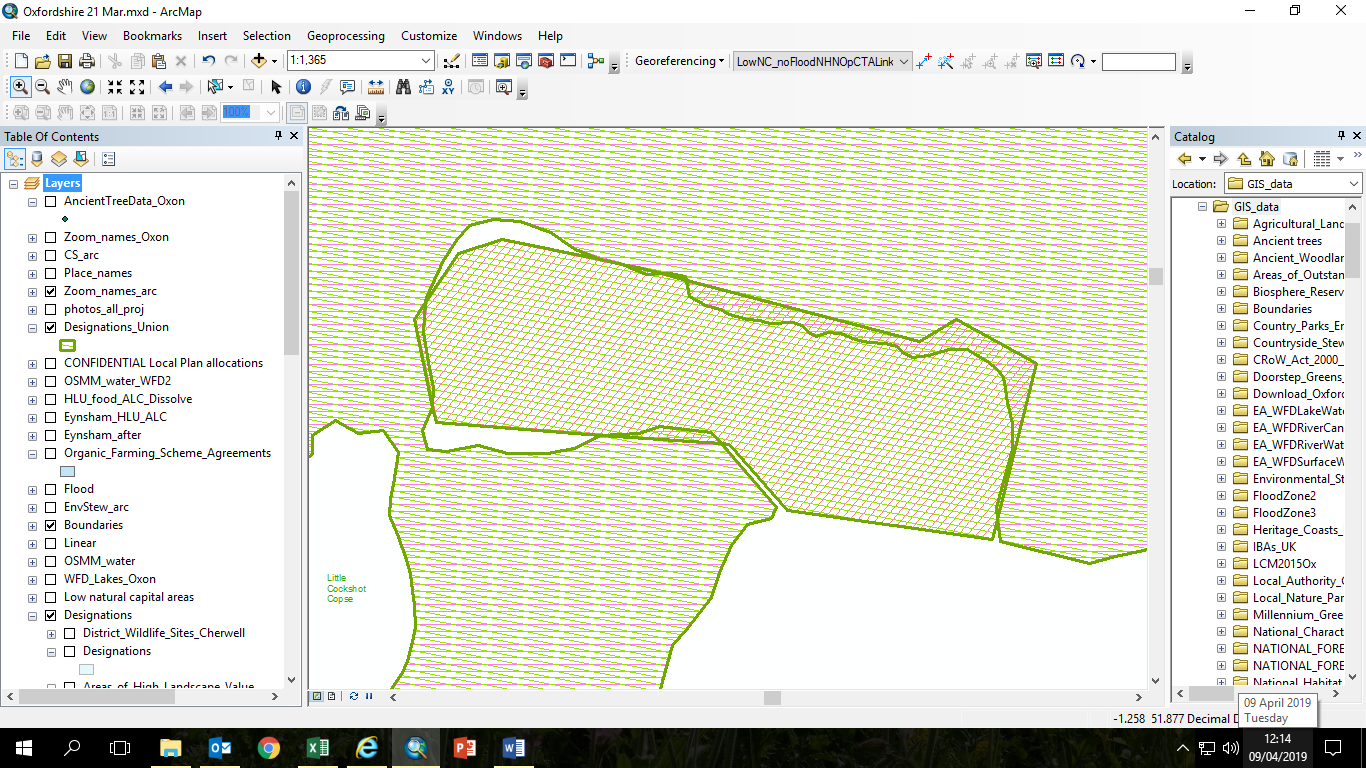
The combined designation layer is created with a Union operation. The order in which the datasets are unioned matters – it is best to start from the dataset with the best fit to OS Mastermap. In this case, this is the green belt dataset. The code pre-processes the green belt dataset by unioning it with itself with no gaps allowed and tolerance 10m, then deleting the gaps to remove internal slivers.

Designation Files table

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Filename | Type | NameField | Name  Length | HabField | Hab  Length | DescField | Desc  Length | NewName | DesType | ShortName | Union  Order | Table  Order |
| Greenbelt\_Oxford\_2016\_17\_bng | Green belt | GB\_Name | 254 |  |  |  |  | GreenBelt | Culture | GB | 1 | 1 |
| Doorstep\_Greens\_Oxon | Doorstep green | NAME | 80 |  |  |  |  | DoorstepGn | Culture | DGn | 2 | 16 |
| Millennium\_Greens\_Oxon | Millennium green | NAME | 80 |  |  |  |  | MillenGn | Culture | MGn | 3 | 15 |
| Road\_Verge\_Nature\_Reserves\_spectrum | Road verge NR | Name | 30 | Habitat\_type | 20 | Site\_description | 200 | RdVergeNR | Nature | RVNR | 4 | 14 |
| Oxfordshire\_Local\_Wildlife\_Sites\_spectrum | LWS | Name | 70 | Habitat | 60 | Site\_Description | 150 | LWS | Nature | LWS | 5 | 13 |
| Proposed\_Oxfordshire\_Local\_Wildlife\_Sites  \_and\_Extensions\_spectrum | Prop LWS | Name | 50 | Habitat | 30 | Description | 130 | Prop\_LWS | Nature | PLWS | 6 | 12 |
| Oxfordshire\_Local\_Geological\_Sites  \_March\_2017 | LGS | Site\_Name | 60 | Sitetype | 60 | Description | 254 | LGS | Education | LGS | 7 | 11 |
| Ancient\_Woodlands\_Oxon | Ancient woodland | NAME | 104 | THEMNAME | 80 |  |  | AncientWood | Nature | AW | 8 | 10 |
| SSSIs\_Oxon | SSSI | SSSI\_NAME | 80 |  |  | CONDITION | 80 | SSSI | Nature | SSSI | 9 | 9 |
| LNR\_Ox | LNR | LNR\_NAME | 120 |  |  |  |  | LNR | Nature | LNR | 10 | 8 |
| NNR\_Ox | NNR | NNR\_NAME | 120 |  |  |  |  | NNR | Nature | NNR | 11 | 7 |
| SAC\_Oxon | SAC | SAC\_NAME | 80 |  |  |  |  | SAC | Nature | SAC | 12 | 6 |
| RSPB\_Reserves\_Oxon | RSPB | Name | 80 |  |  |  |  | RSPB | Nature | RSPB | 13 | 5 |
| AONB\_Oxon | AONB | NAME | 80 |  |  |  |  | AONB | Culture | AONB | 14 | 2 |
| Country\_Parks\_Oxon | Country Park | NAME | 80 |  |  |  |  | CountryPk | Culture | CP | 15 | 3 |
| National\_Trust\_Oxon | National Trust | Name | 80 |  |  | Description | 50 | NT | Culture | NT | 16 | 4 |

Starting from the pre-processed green belt layer, we add the following columns: Name (string 80), Description (250), Habitat (80), Type (20), and short integer fields (which will be set to 1 or zero depending on which designations apply to each polygon) for all the designation types (DoorstepGn, MillenGn, RdVergeNR, LWS, Prop\_LWS, LGS, Anc\_wood, LNR, NNR, SAC, CountryPk, AONB, GreenBelt). Finally there is an integer attribute NumDesig for the number of designations that apply to any one polygon.

We then union all the datasets, entering the starting dataset first (Green belt) and giving it a priority of 1. We set snapping tolerance to10m and set ‘no gaps allowed’. This enables us to manually inspect the ‘gaps’ in the next stage of processing (**Process\_Designations.py**) and remove any that are genuine gaps in the datasets (such as when there is a house in the middle of a nature reserve that is not part of the reserve itself), while retaining any that are simply caused by mis-matched boundaries (see example below of Kitlington Lakes).



Kirtlington Lakes – example of poor fit and gaps. These gaps can be ‘filled in’ by unioning with no gaps allowed.

The Union can leave the fields all jumbled up, so the code sorts them all out into the right order.

### Allocate attributes in the new dataset

Automated with **Process\_Designations.py**

**(Note: this section came from my working notes and has not been checked carefully so it may differ slightly from the latest version of the code. It describes what the python code does and also the manual checking that went on as I developed the code)**

* Check geom revealed >500 self-intersections, so run repair geom.
* Check for duplicate polygons. Easier to do this before converting to single part, but need to check if that creates more duplicates. Duplicates arise if the original datasets contained overlapping or duplicate polygons. If doing manually, use Find Identical first with Shape as the input field, and then check for any that have different names / descriptions and delete manually, so you can decide which one to keep. For any where it does not matter which one is kept (e.g. there were many duplicate road verge nature reserves in the original data), follow with Delete Identical (which keeps only the first instance of each set of identical polygons). Find Identical revealed 36 polygons with duplicate shapes. Inspected most, to see if there are any that have different attributes. Should really inspect each individual dataset for identical polygons, before the union step. Duplicate polygons are either:
  + Edge overlap slivers, e.g. in Green Belt
  + Duplicates in original datasets (LGS, AW)
  + **Two duplicates in SSSI that had different details:** 
    - **I) need to keep Stonesfield Slate Mine**s as that is a smaller area overlapped by Reed Hill. Fortunately that has a lower FID so should be kept anyway.
    - Overlap between Aston Rowant and Chilterns SAC – need to keep Chilterns – again, that has lower FID

For Arc, found a few overlapping corners of SSSIs etc. Only significant overlaps were LNRs Barnwell II and Coldham’s common. Barnwell II is smaller and should be kept – and is also on top so that’s OK.

* Select rows with all FIDs = -1, and set type to “Gap”.
* Convert multipart to single part. This increased number of polygons from 3495 to 6975. Check to see if this has created any more identical shapes – it hadn’t.
* Eliminate small gaps and delete large gaps. There are 387 gaps (out of 3559 polygons, before multipart to singlepart and delete identical) of which 173 are less than 10m2 and 289 are less than 100m2. Detailed inspection showed:
  + all gaps > 2500m2 are genuine.
  + Of the 9 between 1000 and 2500m2, 5 are slivers and 4 are genuine.
  + Of the 14 between 500m2 and 1000m2, 2 are genuine (fields / housing areas).
  + Of the 12 between 250m2 and 500m2, two are genuine and they are both water features in AW.
  + Of the 36 between 100m2 and 250m2, 5 are genuine: 1 building, 3 water features in AW / Lye valley PLWS, I gap around car park in woodland / SSSI.
  + Almost all of the 289 gaps <100m2 are edge slivers but some are buildings. Many have len/area close to 1 so we can’t use that to distinguish. 126 have area <100 and len/area<3, of those I sampled I found 2 buildings in AW, and one stretch of stream in a proposed LWS at Lye Valley.

It would be possible to manually assign the gaps between 250 and 2500m2 as there are not many. So use an **automatic cutoff of 500m**2 and can manually edit if desired. This is very easy – simply take the output from the first elimination stage, select Type = ‘Gap’ AND ShapeArea <2500 (there are only 23 rows), use ‘Calculate Field’ to set Type to ‘Sliver’ for all these, go into edit mode, highlight and ‘Zoom to highlighted’ one by one, and if it is a genuine gap rather than a sliver, ‘Delete highlighted’. Restart the code, which will eliminate the remaining slivers and delete the remaining gaps. It would be possible to separately erase buildings and water features from AW and WS if desired, to re-instate the genuine gaps that were lost for shapes <500m2.

* Go through the designation types in order of the priority, starting with the one that takes lowest precendence in the ‘Type’ field – then if other designations also apply to that polygon they will overwrite that one. Small areas (e.g. road verge nature reserves) should take precedence over larger areas, e.g. Green belt, AONB. Suggested order of precendence (starting with least important): Green belt, AONB, Country Park, LGS, Proposed LWS, LWS, LNR, SSSI, NNR, SAC, Ancient woodland, Doorstep and Millenium greens. For each designation type, select rows with FID for that type >=0. Populate the designation field with ‘1’ for the selected rows.
* Use ‘calculate fields’ to fill in name, habitat and description from the equivalent dataset fields. This will overwrite any information from designations that are taking lower priority.
* Delete any un-needed fields from the new dataset (e.g. FIDs, status) but keep names and descriptions as they are useful in cases with more than one designation.
* In the NumDesig field, add up all the columns that contain 0 or 1 to indicate whether the designation applies. The number ranges between 1 and 5. Those scoring 5 are ancient woodlands in an SSSI, NNR and SAC and AONB (Aston Rowant).

The output has only 25 slivers <1m2.

## Merge designations into base map

Re-run Merge\_into\_base\_map.py but set the input parameters to choose merge\_type = Designations.

# Add in Green Space layers.

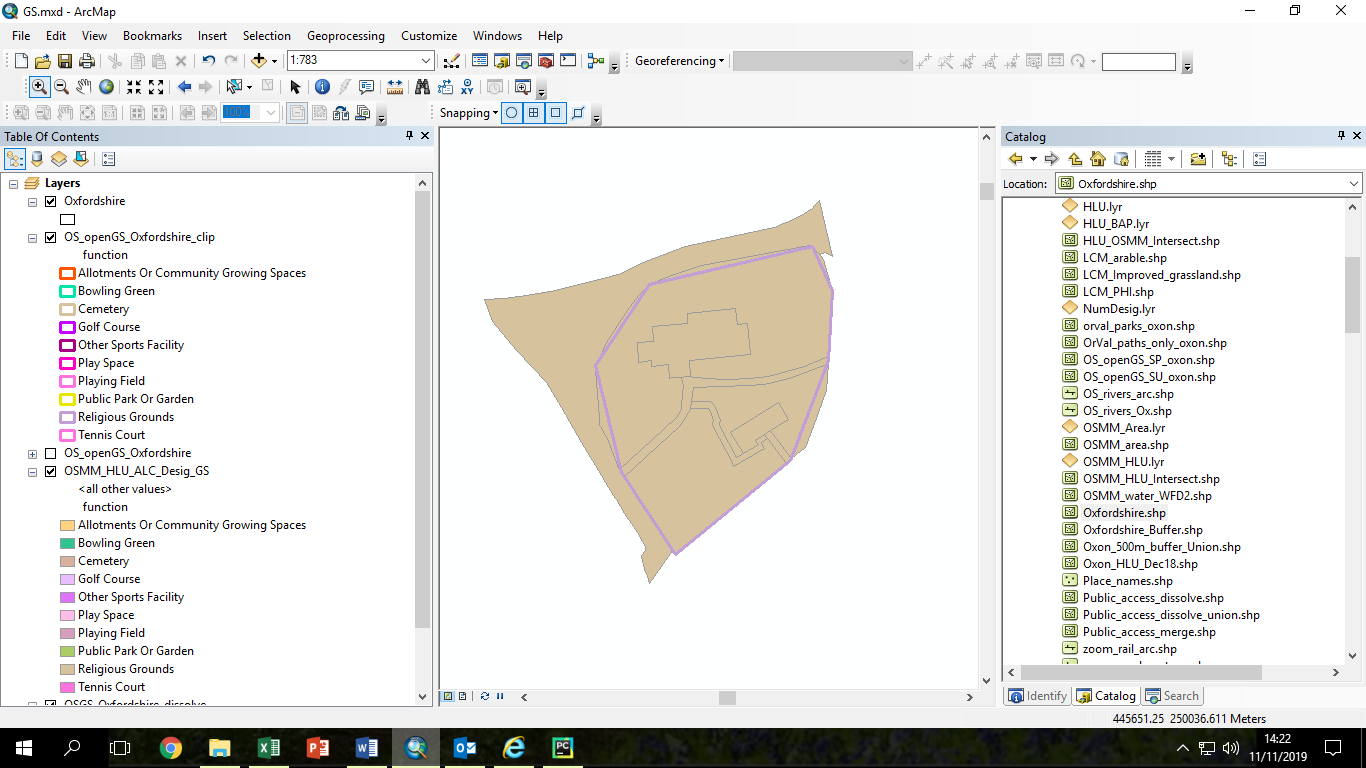
See **join\_greenspace.py**. Three sources:

1. OS open greenspace has all allotments, playing fields, etc. Open GS does not have TOID so will have to be a spatial join – we need this for the rural GS. Has access points and names. Boundaries do not match OSMM. Can be multiple layers, e.g. a playground on top of a park. Data includes name and function.
2. OSMM greenspace (not open data) has all green space in urban areas, including gardens and verges (which we don’t need), but not in rural areas. OSMM GS has TOID so can just be linked directly (excluding amenity and gardens as we have those already). Slightly different classifications to open GS. Uses OpenGS Function. Where there are multiple layers of Open GS, these are shown as Primary and Secondary function. We don’t really need Form (woodland, water etc) because we have this from the habitat / land use data.
3. Orval Parks – have asked and been given permission from Brett Day. Much comes from open street map – but poor fit to OSMM. Used for public access (deleting data types that we already have from designations), but not currently for GS.
4. Prepare and join to OSGS

* Copy all files to a single directory on C drive
* Loop through all files
* Select all rows except gardens (previously also amenity but now we use that).
* Append to first file (could stop if gets too big, or split into SU, SP etc)
* Clip to Arc boundaries
* Join priFunc and secFunc to base map on TOID.

### Steps to link to OS Open GS

* Remove GS already covered by OSGS - clip out by dissolved area of OSGS.
* Tried selecting base map polygons that ‘have their centre in’ OpenGS polygons – selecting each type of GS polygon in turn - and assigning same function. Followed this with spatial join (again ‘have their centre in’) to transfer names. However, this picks up polygons that curve around the outside of GS polygons, like this.



* So tabulate intersection and pick ones that are mainly (>50%) within OpenGS. Seems to work oK. However – **There is a lookup table that links the OSGS and OpenGS datasets, so it would be better to use that**. However this does not appear to contain all the Open GS polygons. Only about 25% have an entry, last itme I checked.
* Delete un-needed fields
* Join to multiplier table
* Export to new file
* Interpret habitats including new GS info – decide when this takes precedence (e.g. for allotments). Do not interpret all habitats as cemeteries etc as this masks woodland, shrub etc. Only use the GS designation for grassland or culltivated areas. Need to use the Greenspace field for public\_access.py and for multipliers in SetUpScores.py.

# Public access

**Public\_access.py.**

The scores for recreation reflect probable ‘usability’ for different activities. The accessibility layer modifies these scores based on accessibility.

**These are my working notes – sorry!**

## Paths

Based on South and Vale GI strategy ANGSt approach: PROW paths count as ANG if buffer of 50m non-urban around paths, and within 300m of settlement. n.b. how big is a ‘settlement’? They added in other accessible areas defined by working with council officers.

* Oxon: we have PROW, Orval paths (from open street map, tagged as accessible) and Sustrans off-road routes (usually built paths – but give access to surrounding areas. Raises the question of whether we should also buffer roads / car parks!)
* Paths that are only in Orval, not PROW, are typically permissive paths (e.g. in Bagley wood) or alongside roads (e.g. cyclepath from Abingdon to Culham). They were derived from Open Street Map and tags were used to determine whether they are accessible or private.
* For the Arc outside Oxon we do not have PROW. Would be good to get this from individual LAs. We have Orval paths, but these omit a lot of paths. So I updated this by downloading the **latest Open Street Map** from Geofabrik. This now shows a lot more paths, because the Orval version was derived from OSM in 2016. The problem is that the downloadable shapefiles from geofabrik do not include the accessibility tags. The Orval team got round this by downloading the full dataset from the pbf version using an Osmosis query to get it into a PostGIS dataset, but it would take time for me to find out how to do this myself. So for now, I will simply use the latest version of OSM instead of Orval paths. This will include some paths that are actually private. Some of these are in military facilities so these can be clipped out (using osm land use). Note however that there can be errors in OSM paths, e.g. near Kempston – county boundary marked as path!
* Orval uses a 25m buffer: “we expand each path by 25 metres to the left and right and then create a land cover grid over that buffered area. Accordingly, the land cover for a path is described by the nature of the greenspace along the 50 metre wide strip through which it passes”. “For access points on the paths network… we calculate a distance-weighted sum of the grid points where the distance weights decline from a value of one for grid points on the path adjacent to the access point to a value of zero at distances of 10 km or more.
* We apply a buffer of 50m but this is problematic especially in urban areas – can pick up green space not attached to path – houses in the way! Ideally we need to make sure the GS is contiguous with the path but this will not work if, for example, there is a narrow strip of easily crossable habitat (e.g. verge or small stream) between the path and a larger area of green space. Only want to eliminate green space if it is cut off from path by buildings, motorway / major road, railway, private gardens? So could extract these ‘barrier’ land cover types within the buffer zone, dissolve the remaining habitats (which will include small areas of sealed suface, small roads etc), select the dissolved areas that intersect the path and then use these to clip the green space in the buffer zone?
* Lots of problems handling the updated OSM paths – erase, union and delete\_identical crash (maybe OSM buffer is just too complex). Tried manually clipping out all OSM paths that were within Orval path buffer, then erasing with Orval path buffer before restarting code with buffer\_paths set to False. After merging OSM and Orval buffers, convert merged path buffers to single part and check and repair geometry. Only 1 geometry problem found so that may not have been the issue. However it still crashes at Delete identical – probably because the polygon path buffers are too large and complex (too many vertices to match polygons). This stage would probably have worked better without the dissolve stage, but I thought the dissolve was needed to make earlier steps work? Trying again without dissolving individual path segments – now the erase and union work but join field was taking a very long time as there are over 8 million polygons somehow, including 2.5M slivers! Really not sure why. Rewrote with Add Join and Calc Field instead. Still fails on Delete Identical. Go back to dissolving paths before merge. Rewrite merge method so we do one layer at a time, erasing before merging, so we don’t have to delete identical polygons. There are a few identical polygons in the input datasets though (two in CROW, two in Orval) that need removing separately. Need some bespoke code to combine CROW and Orval to make sure access is always set to ‘OPEN’ for CROW areas?
* Could include urban paths only if green? Orval excludes urban paths unless alongside a river etc. or natural area. But as we select only the natural areas alongside paths, maybe this is not necessary. The surface of the path could be less important than the fact that it allows access to green space / recreation in green space.

## Rivers

Included as a separate layer. Could use navigable rivers – the definition seems to cover only selected larger rivers and canals. Many smaller waterways could be used by canoes, kayaks and swimmers, but access may be disputed by landowners and anglers. See <http://access.canoedaysout.com/map>. Seems like a contentious issue – don’t want to imply that only navigable rivers are accessible so leave this for now (also, I can’t find navigable rivers in GIS form).

## Areas

* CROW. Orval is supposed to use CROW but actually many CROW areas are missing from Orval parks.
* RSPB reserves – the ones in the Arc are all open to vistors.
* Orval parks – these are heavily based on Open Street Map. The Orval team attempted to retain only publically accessible areas by removing areas tagged as ‘private’ access, and only retaining features with Access key null or tagged as ‘public’ ‘yes’ ‘permissive’or ‘destination’:
  + OSM ‘Parks’ - keys Landuse or Leisure tagged as ‘park’ ‘recreation\_ground’ ‘village\_green’ ‘common’. Orval team removed small areas (<0.4ha) and. They also removed school grounds and areas tagged as ‘FC’‘sports club’‘sports centre’‘leisure centre’‘club’
  + OSM nature reserves (‘nature’), public gardens, ‘cemeteries’ (incl. churchyards and graveyards), allotments, Playgrounds, Parking and Picnic Sites
  + OSM ‘golf courses’, but removing areas tagged as ‘nets’‘driving range’ ‘putting’ ‘crazy’ ‘adventure’ ‘mini’
  + OSM features in which the keys Natural had an entry that was not ‘water’, ‘beach’ or ‘sand’ for which access was specifically labelled as 'public', 'yes', or 'permissive' or had a name that included one of the following: recreation,common,park,heath,open access, community, play area, play space were assumed to be publicly accessible natural areas. OSM Natural key of ‘wood’ or ‘forest’ = type ‘wood’ and the rest = type ‘nature’.
  + Country Parks, NNR, LNR, Doorstep and Millennium Greens – now we strip those out as we have already included them as designations.
  + FC National Forest Estate England Recreation Routes – used to define areas enclosed by recreational path networks (concave hull technique)
  + Woodland Trust / Forestry Commission Woods for People – open access areas (2011 dataset). Cut into blocks separated by trunk roads, and areas <0.4 ha removed.
* Combine path buffers, Orval parks, CROW. Note: Orval parks are supposed to include CROW but do not include all of them.
* Extract only natural areas from base map – exclude gardens and manmade. We already have OS GS and OS openGS, so strip these out as well. Then intersect public access with this layer, and merge back into the base map. See python script for full procedure. Need to remove manmade and gardens first otherwise it doesn’t work (files too large / complex – runs for several days then fails). Still produces over 9000 slivers – probably partly because OS Open GS does not match OSMM boundaries. Switched to incorporate OSGS and OS OpenGS first, so that all boundaries will match OSMM better. Remove OSGS and OS openGS before intersecting. Now get 6739 slivers. Singlepart – get 10,909 slivers. Down to 9983 with improved method – leaving out polygons with existing designation or greenspace info. Ideally would use proper merge method, not just intersect. Or could just eliminate slivers.
* Add in amenity grass, and OSGS ‘Amenity – Residential OR Business’ as most of this is accessible (even on business premises – for those who work there.) Note – seems to be an error in OSGS in that habitats where ‘Rail’ in in DescriptiveGroup along with other categories are incorrectly identified as Amenity – Residential Or Business. Assume they excluded DescriptiveGroup = Rail but forgot about e.g. DescriptiveGroup = Natural Environment,Rail. Seems to be OK for roads. We do not currently include Amenity – Transport as it includes many roundabouts and inaccessible verges e.g. along motorways and other major roads. Later – can we work out which verges are accessible and which are not? Though they don’t have much value for recreation.
* Need to get other accessible areas from LAs, e.g. any LWS, LGS, SSSIs, ancient woods, roadside nature reserves. thatare accessible. Also need: Earth Trust, Ox Pres Trust. Rivers. Not all natural areas with public access are included – e.g. woods and paths around our local playing fields and by the river. But those might be included in OSGS Amenity?
* Also apply to **education – but need to include extra sites e.g. Youlbury, higher multiplier for school playing fields** and designated sites get ‘special educational use’ multiplier??
* Schools – have set a separate multiplier of 0.9 as they are restricted to the general public but still very valuable for recreation. Can use existing OSGS for schools in urban areas but for rural schools we would need to use OSM.
* Interaction with nature – think about how this differs to biodiversity (though have different scores)

# Set up score table

Set up scores in a spreadsheet Matrix.xlsx and import excel to table.

**Set\_up\_scores.py**

Code for setting up score table. Joins to score matrix, sets up and applies multipliers and then calculates average and max scores. Final output NatCap\_(…area), currently one file for Oxon and one for rest of Arc.

Sets up multipliers for food (using ALC), Aesthetic value (using AONB), Education, Interaction with Nature and Sense of Place (using designations) and Recreation (using public access).

Could also use green space categories to inform Education, Interaction with Nature and Sense of Place. Currently cemeteries and allotments get downgraded for Sense of Place if they are not in a designated area. Could either make them exempt from the multipliers, or could use ‘cemetery’ and ‘allotment’ as supplementary designations (would have to make them equivalent to three designations to avoid any downgrading). Considered playing fields - a playing field on a Millenium or Doorstep Green should probably get a sense of place multiplier but probably not in other cases – but then why should an arable field or pasture get a higher score in an AONB but not a playing field? Manmade habitats, landfill, quarries etc should not get a multiplier for being in an AONB etc, but they score zero anyway. However, perhaps it is right that designated sites score more – for example this gives incentive to protect designated sites from development even if development includes allotments and cemeteries. So leave it for now. If we want to change it later, the solution is to add ‘GreenSpace’ as a parameter to the function in SetUp Scores and use it to alter the multipliers where GreenSpace is cemetery or allotment.

Should school grounds get a higher score for education? Not sure for low value habitats – playing field does not have any value for education about nature. Maybe only for high value habitats? Too complicated – leave for now. Though multipliers will uplift high value habitats by more than low value habitats.

# Set up maps

Tried to write code for this but it didn’t work. Currently done manually by merging LADs for each county and/or for whole Arc, or Arc and Oxon separately. Then set up natural capital layers, or use existing template (OxCamArc.mxd) with layers, place names, boundaries and linear features already set up.

Can be slow to display - maybe partly because quite a few of the designation layers have dense vertices – e.g. looks as if all LWS may have been densified. Sort by ‘Shape’ to improve this – this is now within the python code. But can also set up one or more layers as a ‘Base Map’ which improves the speed a bit. However this seems to stop the mxd from opening from ArcMap – you have to click on it with ArcMap closed to open it (and this opens ArcMap at the same time).

# TO DO

My cryptic to do notes – can be ignored unless you are particularly keen to do some further improvements!

## Oxon

* Discuss with TVERC: Grassland in HLU that is trees or scrub in OSMM – ok to class as ‘scrub on grassland’ or ‘woodland on grassland’ or ‘with scattered trees / scrub?’ Can any semi-natural or OSMM rough grassland with scattered trees be classed as parkland?
* Check with Nick Groome - Does the order of the descriptive terms signify precedence? Simon says yes.
* We got most allotments etc from OSGS – but there are a few more in Orval parks.. Not urgent to add these.
* Later - add Earth Trust, Ox Pres Trust. Council data, e.g. which LWS are accessible. Oxford City GS data. Could try merging instead of intersecting public access layer to reduce slivers.
* Bicester – some incorrectly mapped OSMM fields marked as Natural Surface instead of Agricultural Land end up flagged as amenity grassland and therefore open access. How can we fix this? Correct the polygons? Or is there a new rule that can sort this out?

## Arc

* Try Merging access layer instead of intersect? And / or dissolve without PA name? Aylesbury 577 slivers in OSMM\_LCM\_PHI\_ALC\_Desig\_GS and 2825 in OSMM\_LCM\_PHI\_ALC\_Desig\_GS\_Access. So would be good to try Merge if time – but not a priority now.
* Not all school playing fields are identified as ‘school’ for access multipliers because we exclude amenity grass and areas with an existing green space classification from the intersect. Used OSGS to correct this for urban schools but use OSM for rural schools? (schools are not included in OS open GS or Orval).
* Check whether we need to use Greenspace field for any multipliers in SetUp Scores.py, for non-generic l habitats (woods etc) within cemeteries, playing fields etc that are not interpreted primarily as cemeteries etc. e.g. extra sense of place multiplier for woods in cemeteries? Extra recreation score for woods around palying fields?
* Finish documenting code.

## Check with stakeholders / further questions and areas for improvement

* Need complete PROW for Arc – Orval paths very incomplete for Oxfordshire. Although I have integrated latest OSM paths.
* Consider validity of 50m path buffer. Orval uses 25m.
* Path buffer – could use only intersecting OSMM polygons (to avoid selecting small patches of GS on the other side of houses, for urban paths)? This could also cut slivers. However, this leads to oddities when the path has a small strip of another land cover type alongside. So don’t do this for now.
* Could reduce number of slivers from intersect with public access layer (Public\_access.py. ) if we dissolve just on multipliers / access classes instead of including names as well. Could cut new slivers from 9000 (for one LAD) to around 3000. Would lose Orval names – OSGS names should be in already.
* Public access categories / multipliers for schools, clubs, allotments.
* Check assumptions re amenity GS (public access for residential but not transport)
* Soil carbon – need to bring in extra data – CEH NC indicators?