

# Land-Based Layers Workflow

## Step 1: FAO

*Download and format FAO pesticide/fertilizer data.*

### Get raw FAO fertilizer data

- go to: <http://faostat.fao.org>
- choose classic version
- click 'Resources-Fertilizers' link
- the above steps should take you to: <http://faostat.fao.org/site/575/default.aspx>
- select data options:
  - country: [World > (List)]
  - year: select all (2002-2010)
  - item: select all items
  - element: [Consumption in nutrients]
  - selected parameters: click show
- click download (csv format)

### Get raw FAO pesticide data

- go to: <http://faostat.fao.org>
- choose classic version
- click 'Resources-Pesticides Use' link
- the above steps should take you to: <http://faostat.fao.org/site/424/default.aspx>
- select data options:
  - country: [World > (List)]
  - year: select all (1990-2010)
  - item: select all the "+ (Total)" items
  - element: [Use]
  - selected parameters: click show
- click download (csv format)

### Get FAO country codes table

- go to: <http://www.fao.org/countryprofiles/modulemaker/en/>
- download "Self-governing territories", XML format

### Get World Bank AG % of GDP table data

- go to: <http://data.worldbank.org/indicator>
- click on "Agriculture, value added (% of GDP)"
- this should take you to: <http://data.worldbank.org/indicator/NV.AGR.TOTL.ZS/countries>
- click 'Download Data->XML'

## Get World Bank GDP table data

- go to: <http://data.worldbank.org/indicator>
- click on “GDP (current US\$)”
- this should take you to: <http://data.worldbank.org/indicator/NY.GDP.MKTP.CD/countries>
- click ‘Download Data->XML’

## Run scripts

- change worldbank xml files to csv file for next step:

```
./worldbank_gdp_parse.py
```

- create table of fertilizer/pesticide consumption in tonnes per country code (run twice with different year ranges; e.g. 2003-2006 and 2007-2010):

```
R --vanilla <fao_update.R
```

- compare/plot results from previous step (run with correct input files from previous step):

```
R --vanilla <fao_compare_temporal.R
```

- the `fao_compare_2008.R` script was used to compare the new methodology to the one used in the original Halpern 2008 paper

## Notes

- Details behind FAO fertilizer methodology can be found here (note that there was a change in the methodology after 2002):  
[http://www.fao.org/fileadmin/templates/ess/ess\\_test\\_folder/Publications/Agrienviromental/Methodological\\_Notes\\_FAOSTAT2011.pdf](http://www.fao.org/fileadmin/templates/ess/ess_test_folder/Publications/Agrienviromental/Methodological_Notes_FAOSTAT2011.pdf)
- Also note the distinction between consumption by nutrients and consumption by weight for the fertilizer data. We have used consumption by nutrients for the final analysis, but also tried consumption by weight (see `fert_by_weight` folder for plots).

---

## Step 2: Landcover/Landuse (LCLU)

*Download and format LCLU data.*

### Dataset background

- desired dataset: MCD12Q1 V051: land cover type 1 (IGBP)
- relevant links:
  - <http://www.bu.edu/lcsc/data-documentation/>
  - <http://earthobservatory.nasa.gov/Newsroom/view.php?id=22585>
  - [https://lpdaac.usgs.gov/products/modis\\_products\\_table/mcd12q1](https://lpdaac.usgs.gov/products/modis_products_table/mcd12q1)

## Download dataset

- go to: <http://reverb.echo.nasa.gov/reverb/>
- follow these steps:
  - Search Terms: mcd12q1
  - Temporal Search:
    - start: 2005-01-01 00:00:00
    - end: 2005-12-31 23:59:59
  - [press enter]
  - Select Datasets: MCD12Q1 version 51
  - [Press: Search for Granules]
  - [Press: Accept]
  - [Press: All Shopping Cart button]
  - [Press: View Items in Cart]
  - [Press: Download]
  - Select URLs to Download: Data
  - Select Download Option: Text File
  - Press: Save
  - [Press: Download]
  - Select URLs to Download: Metadata
  - Select Download Option: Text File
  - Press: Save
- after downloading, empty basket from last step and then repeat above steps with:
  - Temporal Search:
    - start: 2009-01-01 00:00:00
    - end: 2009-12-31 23:59:59

## Run scripts

- download hdf files:
 

```
./go_download_files.sh
```
- merge hdfs into single tif with igbp landcover classes, for both years:
 

```
./hdf_to_tif.sh
```

## Notes

- We also tried to use the UMD classification (see `global_landcover/umd`), but used the IGBP classification for the final analysis.
- Some classification statistics (number of 500 m<sup>2</sup> pixels):

```
umd classification:
  landclass 12 (cropland), 2005:      76716687
  landclass 12 (cropland), 2009:      76463693
  landclass 13 (urban),    2005:      3063063
  landclass 13 (urban),    2009:      3062706

igbp classification:
  landclass 12 (cropland), 2005:      56632531
```

landclass 12 (cropland),	2009:	56090829
landclass 14 (cropland/natural veg),	2005:	42801682
landclass 14 (cropland/natural veg),	2009:	41760296
landclass 13 (urban),	2005:	3058152
landclass 13 (urban),	2009:	3056223

---

### Step 3: Dasym Prepare

*Prepare data for dasymetric mapping. This step splits global landcover rasters by continents.*

- run continent/countries raster prep script:

```
C:\Python27\ArcGIS10.1\python.exe  
[3]_dasym_prepare\landcover_countries_continents_split.py
```

- (Note: when re-running above script, use fresh/original shp file for countries\_units.shp in case .prj disappears during last run)
- 

### Step 4: Dasym

*Run dasymetric mapping.*

#### Run scripts

- use run\_dasym.sh to run dasym\_map.py multiple times for each continent, year, pest/fert combination

#### Notes

- relevant links:  
<http://portal.nceas.ucsb.edu/Members/perry/methods/dasymetric-mapping/>  
[http://portal.nceas.ucsb.edu/Members/perry/methods/dasymetric\\_flowchart2.png/image\\_view\\_fullscreen](http://portal.nceas.ucsb.edu/Members/perry/methods/dasymetric_flowchart2.png/image_view_fullscreen)  
<https://github.com/scw/global-threats-model/tree/master/Dasymetric>
- 

### Step 5: Zonal Stats

*Calculate zonal statistics for each continent – sums of fertilizer/pesticide values for each basin.*

- run zonal stats script:

```
C:\Python27\ArcGIS10.1\python.exe [5]_zonal_stats\terra_stats.py
```

- (Note: you'll need to create blank af, as, etc dirs first in [5]\_zonal\_stats\output)
-

## Step 6: Global Plume Prepare

*Prepare global plume data. This step updates global plume shapefile fertilizer/pesticide sum values to the new basin zonal statistics.*

- first, create blank `global_plume*.shp` files by resetting the DBF fields to zero via `zero_dbf.py` (resulting shapefiles can be found in `output/og_blank`)
- run `update_global_plume.py` twice, with different years
- run `clean_global_plume.py` twice for both time periods to remove zero fert/pest values (note: this script will not update FIDs)
- split up global plume shapefile into parts (for easier processing in next step; sizes are arbitrary):

```
ogr2ogr -f "ESRI Shapefile" global_plume_2003_2006_1.shp \
  -where "FID<35000" global_plume_2003_2006.shp
ogr2ogr -f "ESRI Shapefile" global_plume_2003_2006_2.shp \
  -where "FID>=35000 AND FID<70000" global_plume_2003_2006.shp
ogr2ogr -f "ESRI Shapefile" global_plume_2003_2006_3.shp \
  -where "FID>=70000 AND FID<105000" global_plume_2003_2006.shp
ogr2ogr -f "ESRI Shapefile" global_plume_2003_2006_4.shp \
  -where "FID>=105000" global_plume_2003_2006.shp

ogr2ogr -f "ESRI Shapefile" global_plume_2007_2010_1.shp \
  -where "FID<35000" global_plume_2007_2010.shp
ogr2ogr -f "ESRI Shapefile" global_plume_2007_2010_2.shp \
  -where "FID>=35000 AND FID<70000" global_plume_2007_2010.shp
ogr2ogr -f "ESRI Shapefile" global_plume_2007_2010_3.shp \
  -where "FID>=70000 AND FID<105000" global_plume_2007_2010.shp
ogr2ogr -f "ESRI Shapefile" global_plume_2007_2010_4.shp \
  -where "FID>=105000" global_plume_2007_2010.shp
```

## Step 7: Plume Model

*Run global plume model.*

- run `plume_distributions.R` script (will need to install `maptools` package in R: `install.packages('maptools')`). Use the "0.05%" values in log output to manually update `plume_buffer.py` limits values for pesticides and fertilizer.

```
$ R --vanilla < plume_distributions.R
```

- launch `grass64`:

```
# define new location: get projection from global_plume*.shp
# import global plume vector piece
v.in.ogr dsn="impact_layers_redo/land_based/ \
  [6]_plume_prepare/output/global_plume_2003_2006_pieces/global_plume_2003_2006_1.shp" \
  output="pours"
# import ocean mask
r.in.gdal -o input="impact_layers_redo/land_based/[0]_og_input/ocean_mask/ocean_mask.tif" \
  output=ocean
```

```
# run plume buffer model
python plume_buffer.py pours
# export grass plume rasters as .tif files
./export_plumes.sh
# repeat above for other plume pieces
# tip 1: run 8 pieces simultaneously during plume_buffer.py for efficiency
# tip 2: if needed, use 'clean_plumes.sh' to erase/re-do plume rasters
```

- combine plumes into single global raster:

```
./gdal_add.py -o global_plumes_pest_2003_2006_raw.tif -ot Float32 plume_pest*.tif
./gdal_add.py -o global_plumes_fert_2003_2006_raw.tif -ot Float32 plume_fert*.tif
# note: may need to add plumes in stages
```

- repeat above steps for other time period
- 

## Step 8: Plume Finalize

*Produce final products.*

- run `difference_rasters.py` to create change detection rasters
  - run `log_normalize_rasters.py` to log-normalize yearly and difference plume rasters
  - run `plume_plots.m` in matlab to produce histogram figures of final rasters
- 

John Potapenko, January-August 2013 (john@scigeo.org)