Vernon, CR, et al. 2018 Demeter – A Land Use and Land Cover Change Disaggregation Model. *Journal of Open Research Software*, 6: 15. DOI: https://doi.org/10.5334/jors.208

SOFTWARE METAPAPER

Demeter – A Land Use and Land Cover Change Disaggregation Model

Chris R. Vernon[1], Yannick Le Page[2], Min Chen[2], Maoyi Huang[1], Katherine V. Calvin[2], Ian P. Kraucunas[1] and Caleb J. Braun[2]

[1] Pacific Northwest National Laboratory, Richland, WA 99352, US

[2] Joint Global Change Research Institute, Pacific Northwest National laboratory, College Park, Maryland, US

Corresponding author: Chris R. Vernon (chris.vernon@pnnl.gov)

Demeter is an open source Python package that was built to disaggregate projections of future land allocations generated by an integrated assessment model (IAM). Projected land allocation from IAMs is traditionally transferred to Earth System Models (ESMs) in a variety of gridded formats and spatial resolutions as inputs for simulating biophysical and biogeochemical fluxes. Existing tools for performing this translation generally require a number of manual steps which introduces error and is inefficient. Demeter makes this process seamless and repeatable by providing gridded and land cover change (LULCC) products derived directly from an IAM—in this case, the Global Change Assessment Model (GCAM)—in a variety of formats and resolutions commonly used by ESMs. Demeter is publicly available via GitHub and has an extensible output module allowing for future ESM needs to be easily accommodated.

Keywords: Global Change Assessment Model (GCAM); Land-use land-cover change; LULCC; disaggregation; downscaling; integrated assessment; model; Python; NumPy

Funding statement: This research was supported by the U.S. Department of Energy, Office of Science, as part of research in Multi-Sector Dynamics, Earth and Environmental System Modeling Program. It builds on previous work supported by the National Aeronautics and Space Administration Carbon Monitoring System and ACCESS programs under projects NNH12AU35I and NNH13AW58I, and by the Laboratory Directed Research and Development Program at Pacific Northwest National Laboratory (PNNL), a multi-program national laboratory operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL01830.

(1) Overview

Introduction

Understanding land use and land cover change (LULCC) is important for projecting the evolution of the Earth system, but tools for downscaling and translating LULCC changes across different modeling platforms are limited. For example, spatial disaggregation of LULCC projections from the Global Change Assessment Model (GCAM) [1–2], which predicts future land use change for each GCAM region and Agro-Ecological Zone (AEZ) or water basin, is generally conducted on a case-by-case basis and a number of manual post-processing steps are required to customize outputs before they can be used by Earth System Models (ESMs). Demeter was developed to refine this process to ensure ease-of-use, output format extensibility, and consistency in the disaggregation methodology among Earth system modelers who utilize global or national GCAM land allocation projections. This requires Demeter to produce gridded data in multiple formats and resolutions, allow users to provide pertinent constraints, evaluate time-series projection data in terms of LULCC, and much more. To accommodate the need for a flexible approach to land allocation disaggregation from zonal to gridded resolutions, Demeter employs a robust intensification and extensification (expansion) algorithm proposed by and demonstrated in West et al. [3] and Le Page et al. [4].

Demeter also enables users to evaluate model uncertainty by hosting the functionality to build random parameter ensembles. The ensemble function in Demeter creates an array of all possible permutations of key influencer parameters as identified by Le Page et al. [4]. The user may then choose the number of randomly selected permutations that he/she wish to evaluate. Permutation runs are conducted in parallel using Joblib in Python and spread over the number of CPUs defined by the user. Each run outputs a unique directory of products.

Demeter was developed to downscale LULCC information from GCAM but could be extended to other IAMs. It is written in Python (version 2.7) and utilizes NumPy to provide an efficient and effective N-dimensional array approach to quantifying global and national LULCC over time. Python's extensive package repository also provides the flexibility that Demeter requires to output

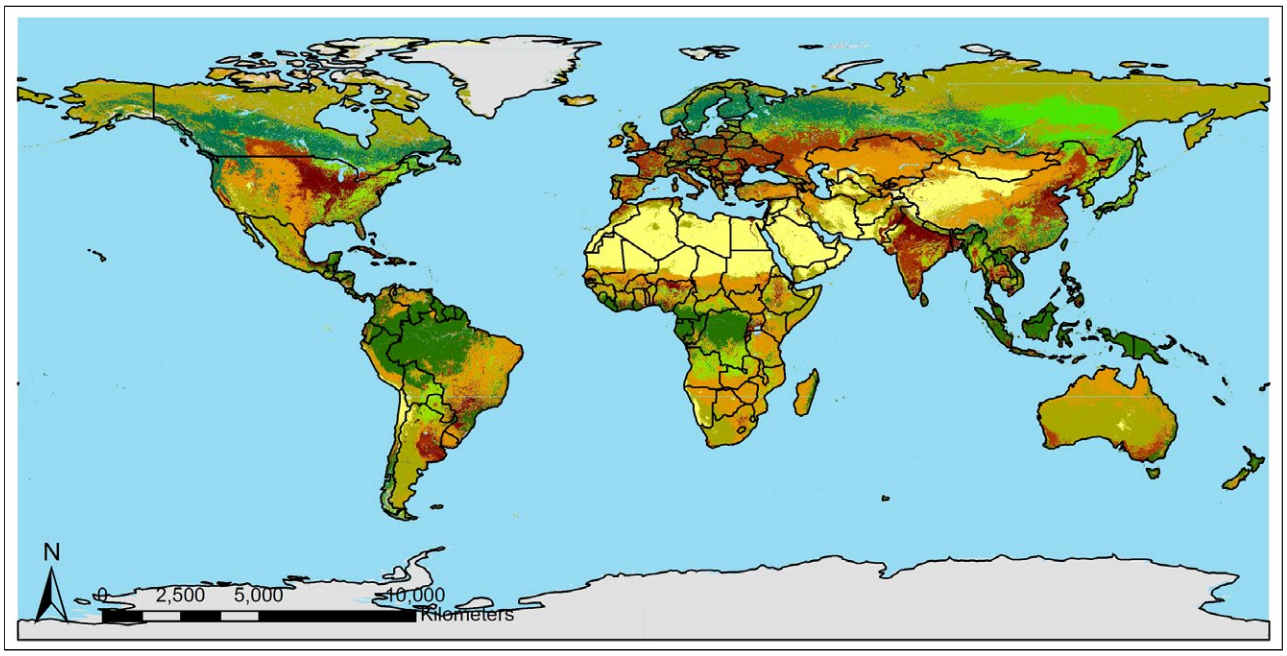

Figure 1: Gridded observational spatial data from MODIS MCD12Q1 version 5.1 [5] product for the year 2005, PFT Type 5 classification [6] at 500-meter resolution.

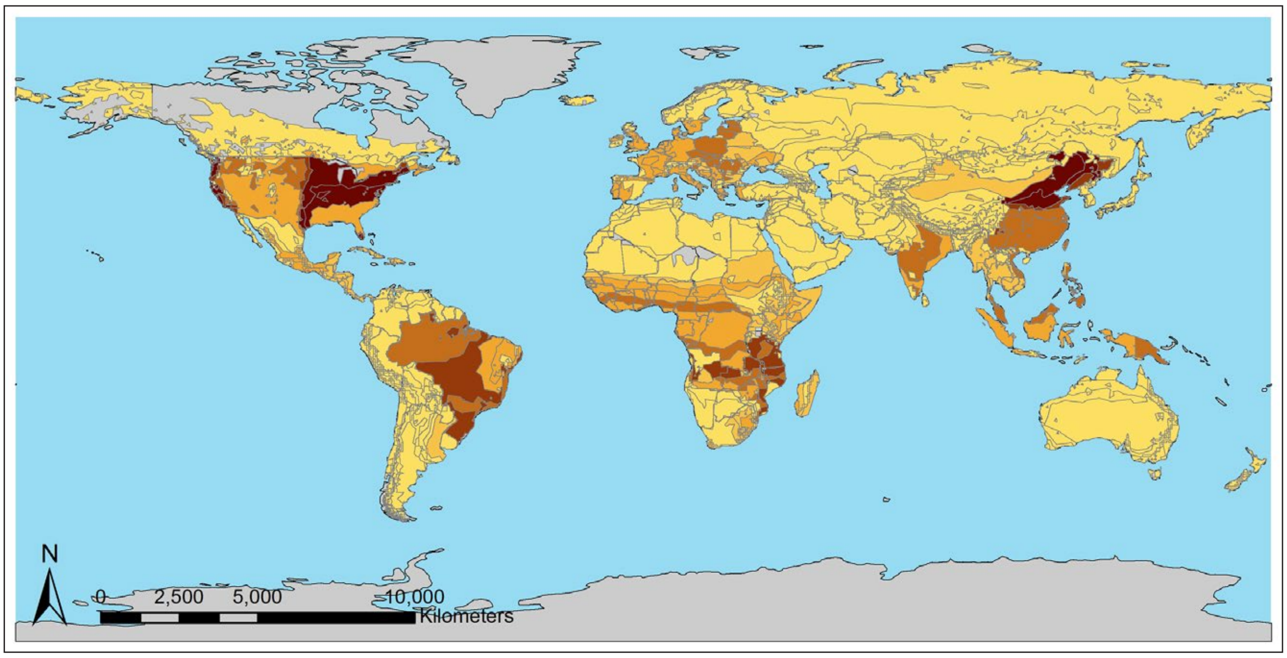

Figure 2: GCAM region/AEZ land allocation for crops for year 2005 from a reference scenario. Gray represents no allocation, yellow represents least allocation, and dark brown is greatest.

data in a variety of formats. Finally, Python was chosen to encourage community involvement in the extension of Demeter due to the language's broad use.

The objective of this paper is to describe Demeter's work-flow, capabilities, output module extensibility, and accessibility for community use.

Implementation and architecture

The scientific underpinnings of Demeter are described in West et al. [3] and Le Page et al. [4]. Demeter uses gridded observational spatial data (OSD) representing real-world land use and land cover observations as an on-the-ground reference to determine the spatially distributed land allocation projections for the base year in an IAM (**Figures 1** and **2**). The OSD serves as a starting point to disaggregate zonal land allocation projections from GCAM to a grid. The spatial resolution of the OSD controls the resolution to which GCAM land allocations will be disaggregated. Demeter's OSD can be derived from publically available LULC products such as MODIS (https://modis.gsfc.nasa.gov/data/dataprod/mod12.php), ESA CCI (https://www.esa-landcover-cci.org/), etc. Land class name and type harmonization between the OSD and GCAM is defined by the user and reclassified during runtime in

Demeter. The harmonization of land area between observed and projected data is achieved by adjusting GCAM's total projected area to the feasible area in the OSD.

Applying the projected future land allocation to the OSD is achieved systematically by intensification and expansion [3, 4]. Intensification is the process of increasing a particular land use or land cover in a grid cell where it already exists, while expansion creates new land use or land cover in grid cells where it does not yet exist but is in proximity to an existing allocation. Proximal relationships are defined by kernel density probabilities that are generated during runtime to determine the probability that a grid cell may contain a particular land use or land cover class (**Figure 3**). The user can define a probability threshold that must at least be present in the grid cell for expansion to occur.

During the intensification and expansion process, user-defined transition priorities for each land class determine the order of transitions among land classes. The order in which land classes are processed is also defined by the user. Additionally, Demeter permits the user to provide custom constraints that may impede siting (e.g., soil quality) and weight them by how the constraint may apply to a specific land class. The intensification and expansion process is completed for each GCAM time step starting from the spatial distribution of LULC area from the previous time step. **Figure 4** demonstrates the OSD and the product of applying the GCAM allocation for 2005

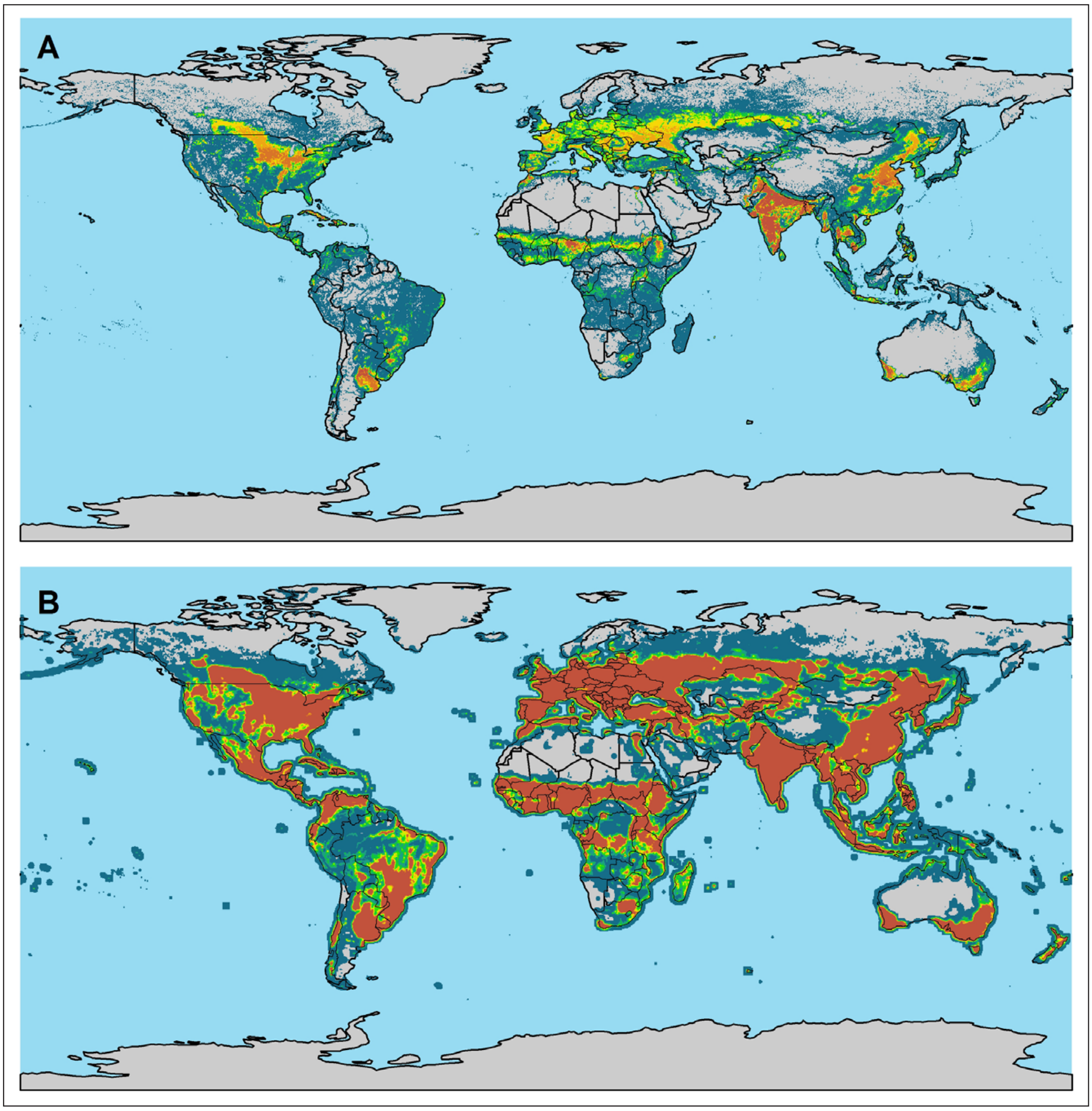

Figure 3: A represents the intensity (warmer colors are higher intensity) of crops per 0.25-degree grid cell for year 2005 after intensification has occurred. **B** represents kernel density smoothing applied to the top panel for use during expansion (warmer colors represent a higher probability of a proximal grid cell being crops).

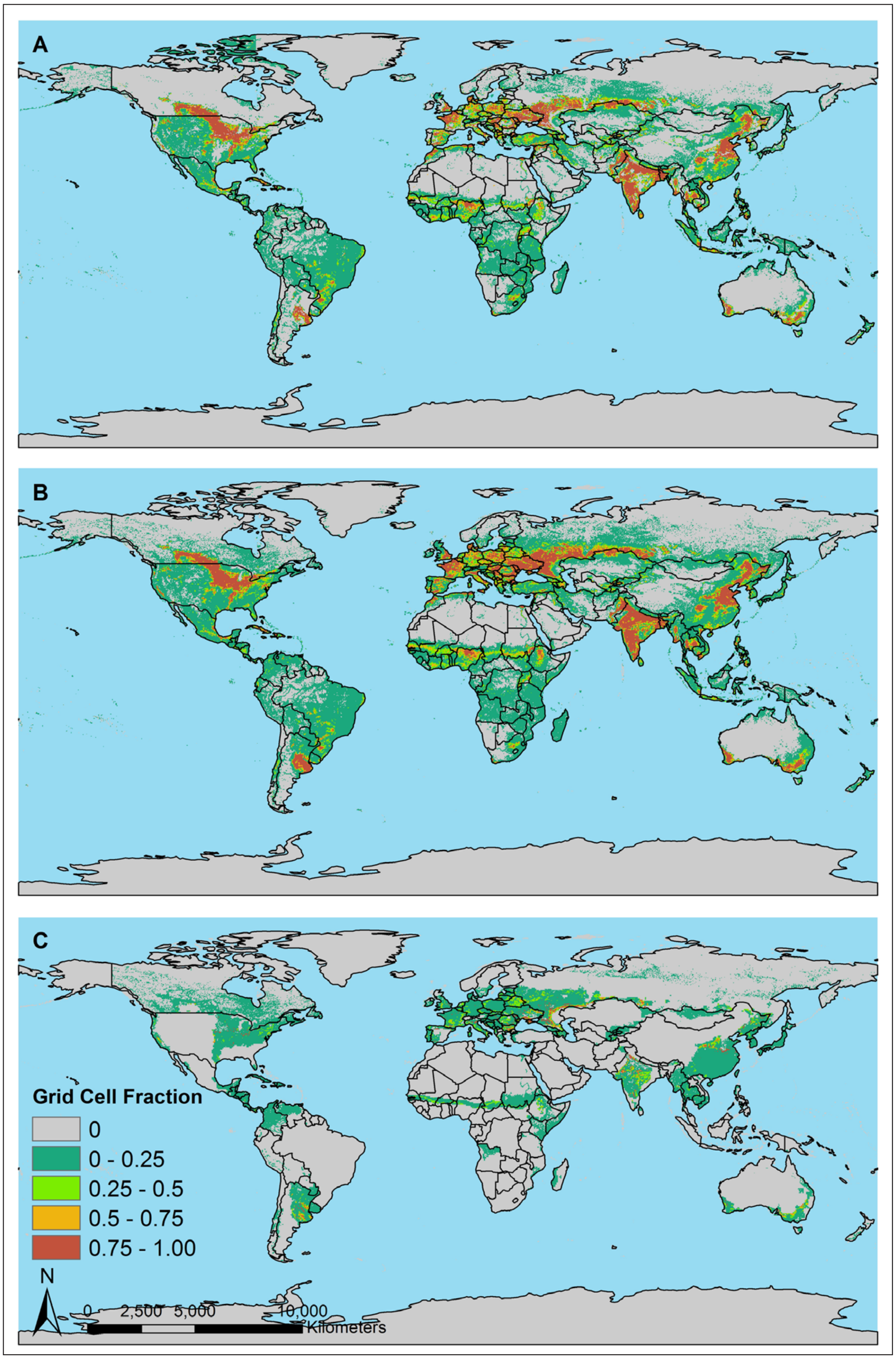

Figure 4: A shows grid cell fraction of crops from the OSD for 2005, **B** shows grid cell fraction of crops after GCAM projections have been applied through intensification and expansion, and **C** shows the change in grid cell fraction from OSD to after GCAM projections were applied. Units for each map are in grid cell fraction.

through intensification and expansion and the resulting change.

Demeter finalizes its run by outputting a suite of tabular, spatial, and mapped data. These outputs can be used to evaluate the spatial distribution of land data and which transitions have occurred where. The user can choose to output their data in a format specifically required by their model or research objectives.

Currently Demeter output products include:

- kernel density probability maps as PNG images per land class,
- land change maps as PNG images per time step per land class,
- land change maps as PNG images showing intensification and expansion per time step per land class,
- land transition maps as PNG images per time step per transition,
- a tabular LULC file per time step in either percent or square kilometres per grid location,
- a spatial LULC file per time step in either percent or square kilometres per grid location as an ESRI Shapefile,
- a tabular LULC transition file per time step per land class in square kilometres per grid location, and
- a NetCDF file per time step containing each land class and the resulting percent land cover per grid cell.

Configuration

Demeter requires the setup of several input files to begin a run. Examples of all input files can be found in the 'example' directory (https://github.com/IMMM-SFA/demeter/tree/master/example) and the expected file structure is outlined in the following:

- Example directory
  - Inputs directory
    - Allocation directory
      - Constraint weighting file
      - GCAM landclass allocation file
      - Kernel density weighting file
      - Spatial landclass allocation file
      - Transition priority file
      - Treatment order file
    - Observed spatial data directory
      - Observed spatial data file
    - Constraint data directory
      - Constraint files
    - Projected GCAM land allocation directory
      - GCAM land allocation file
    - Reference data directory
      - Reference files.

The following describes the requirements and format of each input.

Observed spatial data

This file represents the area in square degrees of each land class existing within a grid cell. The grid cell size is defined by the user. This file must be presented as a comma-separated values (CSV) file having a header in the first row and must contain the field names and fields described in **Table 1**.

Projected land allocation data

This file is the formatted GCAM run output for land allocation projections. Since the format of this file can vary based on GCAM user preference, the file must be formatted to meet Demeter input requirements as described in **Table 2**. The file must be a CSV file having the header in the first row.

Table 1: Observed spatial data required fields and their descriptions.

Field	Description
fid	Unique integer ID for each grid cell latitude and longitude.
landclass	Each land class field name (e.g., shrub, grass, etc.). Field names must not include commas.
region_id	The integer ID of the GCAM region that the grid cell is contained in. Exact field name spelling required.
metric_id	The integer ID of the GCAM AEZ or basin that the grid cell is contained in. Exact field name spelling required.
latitude	The geographic latitude value of the grid cell centroid as a signed float. Exact field name spelling required.
longitude	The geographic longitude value of the grid cell centroid as a signed float. Exact field name spelling required.

Table 2: Projected land allocation required fields from GCAM.

Field	Description
region	The text name of the GCAM region. Exact field name spelling required.
landclass	Each land class field name (e.g., shrub, grass, etc.). Field names must not include commas.
year	Each year of the GCAM run as an integer (e.g., 2005, 2010, etc.).
metric_id	The integer ID of the GCAM AEZ or basin. Exact field name spelling required.
Units	The text name of the units of the data.

Allocation files

Constraint weighting

A weight for each constraint, with a value ranging from –1.0 to 1.0, can be applied to each land class. If no constraints are desired, a user should simply provide a header-only file. For example, for a given land class, the weight for the soil quality constraint with a value of –1 indicates that one land class is fully constrained inversely (e.g., grasslands are opportunistic and grow readily in areas with a low soil quality); a weight of 0 indicates that soil quality exerts no constraint to the land class (e.g., forest, etc.); a weight of 1 for soil quality indicates that high soil quality will highly influence where the type will be spatially allocated (e.g., cropland). These constraints are developed in separate files as described in the following *Constraints* section. See the constraint weighting file in the example inputs for reference.

Kernel density weighting

Weights the degree to which land classes subjected to a kernel density filter will be utilized during expansion to each class. Value from 0.0 to 1.0. See the kernel density weighting file in the example inputs for reference.

Transition priority

This ordering (ascending integers starting from 0) defines the preferential order of final land allocation (e.g., crops expanding into grasslands rather than forests). See the priority allocation file in the example inputs for reference.

Treatment order

Defines the order in which final land classes are downscaled. This will influence the results (e.g., if crops are downscaled first and overtake grassland, grassland will not be available for shrubs to overtake when processing shrub land). See the treatment order file in the example inputs for reference.

Observational spatial data class allocation

This file defines how the land-use and land-cover classes in the OSD will be binned into final land classes for output, which can be defined by the user and serve to place projected land allocation data from GCAM on a common scale with the on-the-ground representation of land-use and land-cover represented in the OSD. Values are to be from 0 or 1, where 1 is used to designate the chosen bin for the land class. Each land class should be binned to only one bin. See the Observed spatial data class allocation file in the example inputs for reference.

Projected GCAM land class allocation

This file defines how the land-use and land-cover classes in the GCAM projected land allocation data will be binned into final classes. Values are to be from 0 or 1, where 1 is used to designate the chosen bin for the land class. Each land class may be binned to multiple bins and the resulting allocation value will be the result of dividing the land evenly among each bin with a 1 designation. See the projected GCAM land class allocation file in the example inputs for reference.

Constraints (not required)

As discussed earlier, constraints such as soil quality may be desirable to the user and can be prepared by assigning a weighted value from –1.0 to 1.0 for each grid cell in the OSD. Spatial maps of constraints should be provided by the user for application during the downscaling process. Users should note that constraining a grid cell to 0.0 may impede the ability to be able to achieve a projected land allocation from GCAM since land area is being excluded that GCAM expects. Each constraint file must have two fields: fid and weight. The fid field should correspond to the fid field in the OSD input and the weight field should be the weight of the constraint per the cell corresponding to the OSD input. Each file should be a CSV with no header.

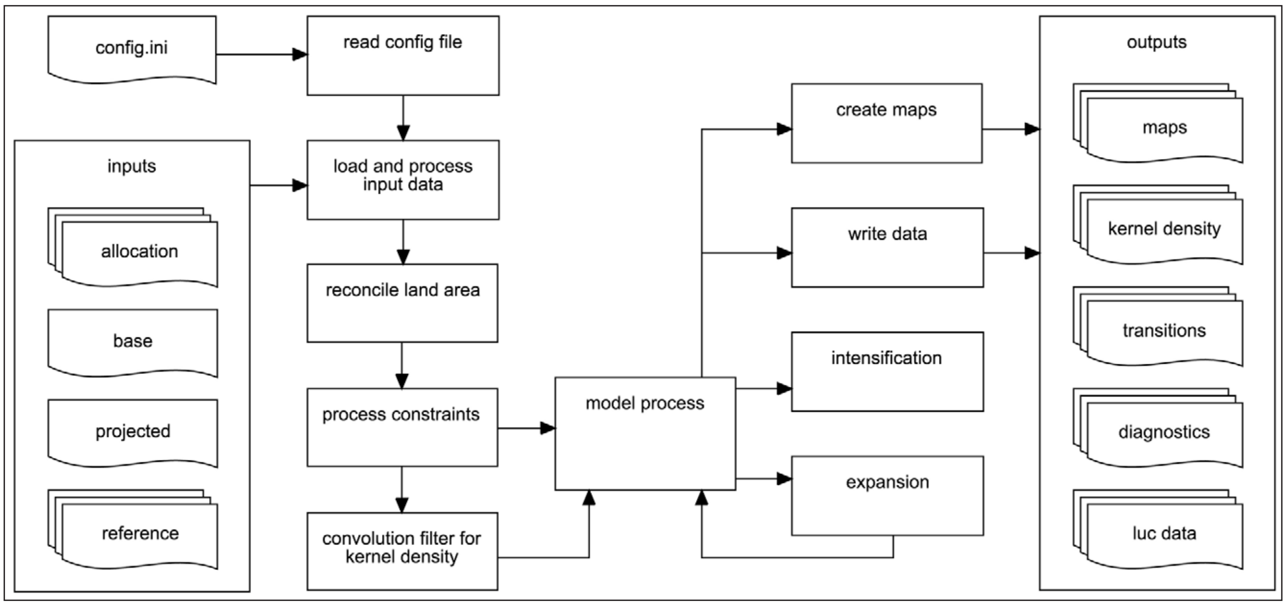

Figure 5: Demeter workflow diagram.

Table 3: Configuration file hierarchy, parameters, and descriptions.

Level	Parameter	Description
[STRUCTURE]	root_dir	The full path of the root directory where the inputs and outputs directory are stored
[STRUCTURE]	in_dir	The name of the input directory
[STRUCTURE]	out_dir	The name of the output directory
[INPUTS]	allocation_dir	The name of the directory that holds the allocation files
[INPUTS]	observed_dir	The name of the directory that holds the observed spatial data file
[INPUTS]	constraints_dir	The name of the directory that holds the constraints files
[INPUTS]	projected_dir	The name of the directory that holds the GCAM projected land allocation file
[INPUTS]	ref_dir	The name of the directory that holds the reference files
[INPUTS] [ALLOCATION]	spatial_allocation	The file name with extension of the observed spatial data class allocation
[INPUTS] [ALLOCATION]	gcam_allocation	The file name with extension of the projected land class allocation
[INPUTS] [ALLOCATION]	kernel_allocation	The file name with extension of the kernel density weighting
[INPUTS] [ALLOCATION]	transition_priority	The file name with extension of the priority allocation
[INPUTS] [ALLOCATION]	treatment_order	The file name with extension of the treatment order
[INPUTS] [ALLOCATION]	constraints	The file name with extension of the constraint weighting
[INPUTS] [OBSERVED]	observed_lu_data	The file name with extension of the observational spatial data
[INPUTS] [PROJECTED]	projected_lu_data	The file name with extension of the projected land allocation data from GCAM
[INPUTS] [REFERENCE]	gcam_regnamefile	The file name with extension of the GCAM region name to region id lookup
[INPUTS] [REFERENCE]	region_coords	A CSV file of GCAM region coordinates for each grid cell
[INPUTS] [REFERENCE]	country_coords	A CSV file of GCAM country coordinates for each grid cell
[OUTPUTS]	diag_dir	The name of the directory that diagnostics outputs will be kept
[OUTPUTS]	log_dir	The name of the directory that the log file outputs will be kept
[OUTPUTS]	kernel_map_dir	The name of the directory that kernel density map outputs will be kept
[OUTPUTS]	transition_tabular	The name of the directory that tabular land transition outputs will be kept
[OUTPUTS]	transition_maps	The name of the directory that land transition map outputs will be kept
[OUTPUTS]	luc_intense_p1_dir	The name of the directory that the land intensification first pass map outputs will be kept
[OUTPUTS]	luc_intense_p2_dir	The name of the directory that the land intensification second pass map outputs will be kept
[OUTPUTS]	luc_expand_dir	The name of the directory that the land expansion map outputs will be kept
[OUTPUTS]	luc_timestep	The name of the directory that the land use change per time step map outputs will be kept
[OUTPUTS]	lc_per_step_csv	The name of the directory that the tabular land change per time step outputs will be kept
[OUTPUTS]	lc_per_step_nc	The name of the directory that the NetCDF land change per time step outputs will be kept
[OUTPUTS]	lc_per_step_shp	The name of the directory that the Shapefile land change per time step outputs will be kept
[OUTPUTS] [DIAGNOSTICS]	harm_coeff	The file name with extension of the NumPy array that will hold the harmonization coefficient data
[OUTPUTS] [DIAGNOSTICS]	intense_pass1_diag	The file name with extension of the CSV that will hold the land allocation per time step per functional type for the first pass of intensification
[OUTPUTS] [DIAGNOSTICS]	intense_pass2_diag	The file name with extension of the CSV that will hold the land allocation per time step per functional type for the second pass of intensification

(contd.)

Level	Parameter	Description
[OUTPUTS] [DIAGNOSTICS]	expansion_diag	The file name with extension of the CSV that will hold the land allocation per time step per functional type for the expansion pass
[PARAMS]	model	The model name providing the projected land allocation data (e.g., GCAM)
[PARAMS]	metric	Subregion type (either AEZ or BASIN)
[PARAMS]	scenario	Scenario name
[PARAMS]	run_desc	The description of the current run
[PARAMS]	agg_level	1 if only by metric, 2 if by region and metric; AEZ is the default
[PARAMS]	observed_id_field	Observed spatial data unique field name (e.g., fid)
[PARAMS]	start_year	First time step to process (e.g., 2005)
[PARAMS]	end_year	Last time step to process (e.g., 2100)
[PARAMS]	use_constraints	1 to use constraints, 0 to ignore constraints
[PARAMS]	spatial_resolution	Spatial resolution of the observed spatial data in decimal degrees (e.g., 0.25)
[PARAMS]	errortol	Allowable error tolerance in square kilometres for non-accomplished change
[PARAMS]	timestep	Time step interval (e.g., 5 years) for the output data. This time step is the increment that Demeter will process when starting with the start year.
[PARAMS]	proj_factor	Factor to multiply the projected land allocation by
[PARAMS]	diagnostic	0 to not output diagnostics, 1 to output
[PARAMS]	intensification_ratio	Ideal fraction of land change that will occur during intensification. The remainder will be through expansion. Value from 0.0 to 1.0.
[PARAMS]	stochastic_expansion	0 to not conduct stochastic expansion of grid cells, 1 to conduct
[PARAMS]	selection_threshold	Threshold above which grid cells are selected to receive expansion for a target functional type from the kernel density filter. Value from 0.0 to 1.0; where 0 lets all land cells receive expansion and 1 only lets only the grid cells with the maximum likelihood expand.
[PARAMS]	kernel_distance	Radius in grid cells used to build the kernel density convolution filter used during expansion
[PARAMS]	map_kernels	0 to not map kernel density, 1 to map
[PARAMS]	map_luc_pft	0 to not map land change per land class per time step, 1 to map
[PARAMS]	map_luc_steps	0 to not map land change per time step per land class for intensification and expansion, 1 to map
[PARAMS]	map_transitions	0 to not map land transitions, 1 to map
[PARAMS]	target_years_output	Years to save data for; default is 'all'; otherwise a semicolon delimited string (e.g., 2005; 2020)
[PARAMS]	save_tabular	Save tabular spatial land cover as a CSV; define tabular units in tabular_units param
[PARAMS]	tabular_units	Units to output the spatial land cover data in; either 'sqkm' or 'fraction'
[PARAMS]	save_transitions	0 to not write CSV files for each land transitions per land type, 1 to write
[PARAMS]	save_shapefile	0 to not write a Shapefile for each time step containing for all functional types, 1 to write; output units will be same as tabular data
[PARAMS]	save_netcdf_yr	0 to not write a NetCDF file of land cover percent for each year by grid cell containing each class; 1 to write
[PARAMS]	save_ascii_max	0 to not create an ASCII raster of the land class with the maximum area for each grid cell per 1 to write
[ENSEMBLE]	permutations	If running an ensemble of configurations, this is the number of permutations to process
[ENSEMBLE]	limits_file	If running an ensemble of configurations, this is the full path to a CSV file containing limits to generate ensembles of certain parameters.
[ENSEMBLE]	n_jobs	If running an ensemble of configurations, this is the number of CPU's to spread the parallel processing over. –1 is all, –2 is all but one, 4 is four, etc.

Configuration file

Demeter's configuration file allows the user to customize each run and define where file inputs are, and outputs will be. The name of the configuration file can be arbitrary since it can be specified as an argument when running Demeter. The configuration options and hierarchical level are described in **Table 3**.

Figure 5 details the Demeter's workflow once the input files have been prepared. The process can be outlined as follows:

1. Configuration file read and input parameters are validated.
2. Input data is read and validated.
3. Grid area discrepancies between the OSD and the projected land allocation from GCAM are harmonized by adjusting the GCAM allocation areas using a correction factor (ratio of the OSD land use data per region and metric to the projected area from GCAM).
4. Constraints are processed if provided and prepared for integration.
5. The convolution filter that will be used to calculate kernel density is prepared. This is applied during each time step.
6. Initial and time step specific arrays are prepared.
7. An initial pass to intensify existing land is conducted.
8. An expansion pass is conducted to apply land allocation projections where the kernel density probabilities meet the user-specified selection threshold.
9. A final intensification pass is conducted to allocate any land projections not yet met.
10. Output products are created.

Execution

Demeter has two main model level functions available to the user: **`execute()`** and **`ensemble()`**. The execute function allows the user to run Demeter based on parameters defined in the configuration file and inputs provided. This is the most common use of Demeter. The ensemble function was built to allow the user to test Demeter using random combinations of the transition priorities, treatment order, intensification ratio, selection threshold, and kernel distance. The ensemble section of the configuration file as seen in the example directory gives the user the ability to define the number of permutations that he/she wish to evaluate as well as a limits file that gives the user the ability to set limits and the interval for sampling values of intensification ratio, selection threshold, and kernel distance. The initial state of the configuration file and input files are used to create a template by which every unique combination of the aforementioned parameters can be generated. Random samples are then drawn based on these configuration options. Demeter runs, each of which represent a unique sample of parameter values, are then executed in parallel based on the number of jobs the user has set.

Demeter can be installed as follows:

1. This repository uses the Git Large File Storage (LFS) extension (see https://git-lfs.github.com/ for details). First install Git LFS, then initialize it using the `git lfs install` command.
2. Clone Demeter into your desired location git clone https://github.com/IMMM-SFA/demeter.git.
3. Install the Python package setuptools (https://pypi.python.org/pypi/setuptools) if the package is not already on your machine.
4. From the directory you cloned Demeter into run `python setup.py install` which will install Demeter as a Python package on your machine and will install the required dependencies. Note: you need admin privileges to run this install.

The procedure to conduct a Demeter run is as follows:

1. Setup your configuration file (.ini). Examples are located in the "example" directory. Be sure to change the root directory to the directory that holds your data (use the 'demeter/example' directory as an example).
2. If running Demeter from an Integrated Development Environment (IDE), please include the path to your config file. A sample script can be found at "demeter/example/example.py".
3. If running Demeter from terminal, please run in the script at demeter/demeter/model.py by passing the full path using forward slashes with no spaces to the configuration file as the first argument and either "standard" or "ensemble" as the second argument (e.g., `python model.py/users/<uname>/github/demeter/example/config.ini standard`).

Both the output directory and log file generated during runtime are named per the run scenario and the timestamp of when the run began. An additional suffix of the permutation number will also be added if using the ensemble function in Demeter. All other outputs directories are named according to what the user specifies in the configuration file.

Quality control

All possible combinations of the configuration file and input files have been tested. Strict requirements for input files are documented. Configuration options have set limits, values, and types and are validated during runtime. An example setup for Demeter is included in the package. The example has a configuration file and an input directory containing all other necessary files. To setup the provided example, the user should simply change the file paths in the configuration file and the example.py file to represent their local directories. The configuration file contains a root path and a full path with file name to the limits file that must be set when using the ensemble functionality.

Demeter outputs the following diagnostic files that help the user evaluate the outputs based on how they were processed during runtime:

- Harmonization coefficient NumPy array
  - This file contains the correction factor that was applied to each grid cell of the projected land allocation data to ensure the projected area is the same as the area actually available as represented by the OSD for the target base year
- Files documenting the land change per year per grid cell from one landclass to its transition class by region and metric
- A detailed log file
- A suite of tabular, spatial, and mapped data

(2) Availability

Operating system
Mac OS X; Linux; Windows 7

Programming language
Python 2.7.13

Additional system requirements
It is recommended that a minimum of 8 GB of memory be available for runs where the spatial resolution of the data is coarser than or equal to 0.05 degrees. Model runs facilitating data with a resolution finer than 0.05 degrees may require additional memory.

Dependencies
- numpy (>= version 1.12.0rc2)
- scipy (>= version 0.18.1)
- pandas (>= version 0.19.2)
- matplotlib (>= version 1.3.1)
- pyshp (>= version 1.2.12)
- joblib (>= version 0.11)
- setuptools (>= version 18.5)

Software location
Archive

Name: GitHub
Persistent identifier: https://doi.org/10.5281/zenodo.1214342
Licence: BSD 2-Clause
Publisher: Chris R Vernon
Version published: v1.0.0
Date published: April 7, 2018

Code repository

Name: GitHub
Identifier: https://github.com/immm-sfa/demeter
Licence: BSD 2-Clause (https://github.com/IMMM-SFA/demeter/blob/master/LICENSE)
Date published: August 30, 2017

Language
English

(3) Reuse potential
Demeter methods and functions contain docstrings that can be called in through Python's built-in help function. Demeter's code base is also verbosely commented to encourage community involvement for future development activities. Other integrated assessment model outputs for projected land allocation could also be utilized provided that they can meet the input requirements. Demeter can also be used in other multi-model, multi-scale integrated assessment studies since it can be called from other models.

An example of future research that could be conducted using Demeter could be to provide disaggregated land cover data as input to a potential evapotranspiration component of a global hydrology model. These types of land-water science applications can easily be evaluated using Demeter.

A tutorial on how to extend Demeter to output different formats is available here: https://github.com/IMMM-SFA/demeter/blob/master/docs/extend_outputs.md.

Acknowledgements
This research was supported by the U.S. Department of Energy, Office of Science, as part of research in Multi-Sector Dynamics, Earth and Environmental System Modeling Program. It builds on previous work supported by the National Aeronautics and Space Administration Carbon Monitoring System and ACCESS programs under projects NNH12AU35I and NNH13AW58I, and by the Laboratory Directed Research and Development Program at Pacific Northwest National Laboratory (PNNL), a multi-program national laboratory operated by Battelle for the U.S. Department of Energy under Contract DE-AC05- 76RL01830.

Competing Interests
The authors have no competing interests to declare.

References
1. **Edmonds, J** and **Reilly, J M** 1985 *Global Energy: Assessing the Future*, 317. Oxford University Press, New York.
2. **Edmonds, J, Wise, M, Pitcher, H, Richels, R, Wigley, T** and **Maccracken, C** 1997 An integrated assessment of climate change and the accelerated introduction of advanced energy technologies-an application of MiniCAM 1.0. *Mitigation and adaptation strategies for global change*, 1(4): 311–339. DOI: https://doi.org/10.1023/B:MITI.0000027386.34214.60
3. **West, T O, Le Page, Y, Huang, M, Wolf, J** and **Thomson, A M** 2014 Downscaling global land cover projections from an integrated assessment model for use in regional analyses: results and evaluation for the US from 2005 to 2095. *Environ. Res. Lett.*, 9. DOI: https://doi.org/10.1088/1748-9326/9/6/064004
4. **Le Page, Y, West, T, Link, R** and **Patel, P** 2016 Downscaling land use and land cover from the Global

Change Assessment Model for coupling with Earth system models. *Geosci. Model Dev.*, 9: 3055–3069. DOI: https://doi.org/10.5194/gmd-9-3055-2016
5. **NASA LP DAAC** 2005 MODIS MCD12Q1 version 5.1 land cover. NASA EOSDIS Land Processes DAAC, USGS Earth Resources Observation and Science (EROS) Center, Sioux Falls, South Dakota (https://lpdaac.usgs.gov), accessed June 29, 2017, at: ftp://ftp.glcf.umd.edu/glcf/Global_LNDCVR/UMD_TILES/Version_5.1/2005.01.01/.
6. **Friedl, M A, Sulla-Menashe, D, Tan, B, Schneider, A, Ramankutty, N, Sibley, A** and **Huang, X** 2010 Algorithm refinements and characterization of new datasets. *Remote Sens. Environ.*, 114: 168–182. DOI: https://doi.org/10.1016/j.rse.2009.08.016

How to cite this article: Vernon, CR, Le Page, Y, Chen, M, Huang, M, Calvin, KV, Kraucunas, IP and Braun, CJ 2018 Demeter – A Land Use and Land Cover Change Disaggregation Model. *Journal of Open Research Software*, 6: 15. DOI: https://doi.org/10.5334/jors.208

Submitted: 08 December 2017 **Accepted:** 08 April 2018 **Published:** 20 April 2018