SYNTHESIS OF A COMPLETE LAND USE/LAND COVER DATA SET FOR THE CONTERMINOUS UNITED STATES EMPHASIZING ACCURACY IN AREA AND DISTRIBUTION OF AGRICULTURAL ACTIVITY

Draft of May 8, 2011 at 11 : 15

A Thesis Presented to
The Faculty of the Department of Geography & Environmental Studies
Northeastern Illinois University

In Partial Fulfillment
Of the Requirements for the Degree
Master of Arts
In Geography & Environmental Studies

By Neil A. Best May 2011

Abstract

This paper presents an effort to produce a new land cover data set for the conterminous United States of America (cUSA) that augments available agricultural land use data with other uses and natual covers to create a complete landscape characterization. Using the Agland2000 data set as a benchmark we formulate a hybridization of the MODIS Land Cover Type (MLCT) for 2001 and the 2001 National Land Cover Database (NLCD) that is particularly tailored to serve as an initialization data set for long-term economic land use change models. In order strike a balance between spatial precision and local diversity of use and cover the new data set has lower resolution than the MLCT ($5'$ vs. 500m) but represents consituent cover classes as sub-pixel fractions rather than discrete categories. After aggregating to the $5'$ grid we present a method for decomposing the natural vegetaion/cropland mosaic class found in MLCT into constituent classes as a function of the local landscape and quantify its contribution to aggregate acreages by class, particularly cropland. We compensate for the absence of certain fine-grained details from MLCT, such as rural transportation networks, small settlements, linear water features, and wetlands, mainly due to sensor resolution, by incorporating corresponding components of the NLCD, after similar reclassification and aggregation, as a set of offsets to the MLCT-derived fractions. The 175Crops2000 data set, valuable for its basis in per-crop agricultural production statistics, is used as a guide to further decompose the cropland areas into a set of crop-specific sub-categories designed to facilitate the economic modeling goals of the simulations that will be initialized by this data product. The final classification scheme, now conceptually equivalent to a stack of spectral bands with the additional quality that the components of each pixel sum to unity, is a mixture of a simplified version of the IGBP schema used in MLCT and a disaggregation of the monolithic cropland class that differentiates among the world's major commodity crops. At each step of refinement we show that overall spatial distribution of cropland across the study area improves relative to the Aglands2000 data set. We close with a discussion of how this method might be applied globally and to successive years in the MLCT time series.

It remains to be seen how well the disaggregation of commodities will work. We may have to leave it out.

Acknowledgements

This thesis is dedicated to my son, Leo. Son, I began working on this degree before you were born and my commitment to completing it was sustained by my desire to demonstrate to you that in life we finish what we have started.

I could not have completed this paper over the past year and, by extension, my degree over more years than I care to mention without the support of my loving wife, Laura.

I want to thank Dr. Nicholas Kouchoukos of Lanworth, Inc. for throwing me in the deep end of applying the open-source geospatial software tool chain to spatial analysis of agriculture.

This work was made possible through the support of my employer, the Computation Institute at the University of Chicago, and its director, Dr. Ian Foster under the Community Integrated Model of Economic and Resource Trajectories for Humankind project (CIM-EARTH, `http://www.cim-earth.org/`).

My thesis committee was comprised of Dr. Monika Mihir (chair), Dr. Erick Howenstine (department head), both of the Department of Geography & Environmental Studies at Northeastern Illinois University, and Dr. Joshua Elliott from the Computation Institute. I deeply appreciate their guidance and support through all stages of this project.

Table of Contents

List of Tables . . . . . **vi**

List of Figures . . . . . **vii**

List of Abbreviations . . . . . **viii**

List of Symbols . . . . . **ix**

Todo list **1**

Chapter 1 Introduction **2**
1.1 Background 2
1.2 Objective 4
1.3 Reproducible Research 7

Chapter 2 Data Sets **10**
2.1 MODIS Land Cover Type (MLCT) 11
2.1.1 Reclassification 11
2.1.2 Aggregation 15
2.1.3 Mosaic decomposition 22
2.2 Agricultural Lands in the Year 2000 (Agland2000) 25
2.3 National Land-cover Database 2001 (NLCD) 25
2.3.1 Reclassification 27
2.3.2 Aggregation 27
2.4 Harvested Area and Yields of 175 Crops (175crops2000) 27

Chapter 3 Analysis **32**
3.1 NLCD Offsets 36

Chapter 4 Conclusions **42**

References **43**

Source Code: Data Sets **45**

Source Code: Analysis **81**

List of Tables

1.1 Summary of global LULC data sets 3

3.1 Total Acreages by Map and Cover 32
3.2 RMSE, MLCT vs. Agland2000 crop 34
3.3 Balance of adjustment fractions and original MLCT aggregation 38
3.4 RMSE, MLCT vs. Agland2000 crop with NLCD offsets 38
3.5 Effect of NLCD offsets on total acreages, $A_{min} = 0.5$ 40

List of Figures

2.1 MLCT primary cover reclassified detail 12
2.2 MLCT secondary cover reclassified detail 12
2.3 MLCT primary cover classification confidence 13
2.4 MLCT primary covers shown separately, detail 14
2.5 MLCT secondary covers shown separately, detail 14
2.6 MLCT primary cover reclassified 15
2.7 MLCT secondary cover reclassified 15
2.8 MLCT primary cover classification confidence 16
2.9 MLCT primary covers shown separately 17
2.10 MLCT secondary covers shown separately 18
2.11 Sub-pixel fractions at original resolution for $A_{min} = 0.5$ 19
2.12 Sub-pixel fractions at original resolution for $A_{min} = 1$ 20
2.13 Aggregated sub-pixel fractions for $A_{min} = 0.5$ 20
2.14 Aggregated sub-pixel fractions for $A_{min} = 1$ 21
2.15 Difference of aggregated sub-pixel fractions 22
2.16 Aggregated cover fractions after mosaic decomposition, $A_{min} = 0.5$ 23
2.17 Aggregated cover fractions after mosaic decomposition, $A_{min} = 1.0$ 24
2.18 Differences of sub-pixel fractions after mosaic decomposition, positive when $f(A_{min} = 0.5)$ is greater 24
2.19 Agland2000 distribution in detail area 26
2.20 Agland2000 distribution in cUSA study area 26
2.21 NLCD reclassified 28
2.22 NLCD covers shown separately, detail 28
2.23 NLCD aggregated cover fractions, detail area 29
2.24 NLCD aggregated cover fractions 30
2.25 175Crops2000 category maps 31

3.1 Total Acreages by Map and Cover 33
3.2 Scatter plot of MLCT crop ($A_{min} = 1.0$, no mosaic) versus Agland2000 cropland 34
3.3 Scatter plot of MLCT crop ($A_{min} = 0.5$, no mosaic) versus Agland2000 cropland 35
3.4 Total offsets calculated from NLCD 39
3.5 Area totals after NLCD adjustment 39
3.6 Scatter plot of MLCT adjusted crop versus Agland2000 cropland 40
3.7 Agland Complete cover maps 41

List of Abbreviations

175Crops2000	Harvested Area and Yields of 175 crops (M3-Crops Data) (Monfreda et al., 2008)
Agland2000	Agricultural Lands in the Year 2000 (M3-Cropland and M3-Pasture Data) (Ramankutty et al., 2008)
arcmin	minute of arc, 1/60th of a degree
arcsec	second of arc, 1/60th of a minute, 1/3600th of a degree
AVHRR	Advanced Very High Resolution Radiometer
cUSA	conterminous (contiguous) Unites States of America, the "lower 48"
GLC2000	Global Land Cover 2000 (European Commission, 2003)
GRASS	Geographic Resources Analysis Support System, `http://grass.osgeo.org`
IGBP	International Geosphere-Biosphere Programme
LULC	land use / land cover
MODIS	Moderate Resolution Imaging Spectroradiometer
MLCT	MODIS Land Cover Type (DAAC, 2008)
NLCD	National Land-Cover Database, 2001 (Homer et al., 2004)
RMSE	root of the mean squared error
SPAM	Spatial Production Allocation Model
SPOT	Système pour l'Observation de la Terre

List of Symbols

A_{min}	Minimum sub-pixel fraction possible for primary cover given in MLCT base data
A_s	Sub-pixel fraction of secondary cover type, function of classification confidence level and A_{min}
A_p	Sub-pixel fraction of primary cover type, function of classification confidence level and A_{min}
$\hat{\theta}$	Predicted sub-pixel fraction
θ	Observed sub-pixel fraction
$'$	minute of arc, 1/60th of a degree
$''$	second of arc, 1/60th of a minute, 1/3600th of a degree

Todo list

It remains to be seen how well the disaggregation of commodities will work. We may have to leave it out. iii
Warning–page numbers missing in Fisher2005a 2
Does everyone agree that we can do without this passage? 5
Figure: MLCT reclassification table 11
Is this figure any better placed than others? 11
cite email from Friedl 16
consider including histograms showing confidence distribution 16
cite Friedl email 22
check whether/how urban, water, wetland are informed with priors in NLCD 25
Incorporate Joshua's suggestion to show further NLCD detail to better illustrate the discrepancy in developed areas 27
Figure: NLCD reclassification table 27
Figure: Table of crops and types reproduced from (Monfreda et al., 2008) 27
Figure: Summary table of crop aggregations for our model 29
Address issue of smaller land mask for 175crops2000 and Agland2000 . . 29
Figure: error map for "nomos" vs. Agland2000 crop 33
Figure: NLCD mask facet map 36
Figure: MLCT resampled and masked facet(?) map(s?) 36
Figure: Facet map of thumb offsets 37
Figure: Difference map, thumb adjusted vs. original after mosaic decomposition 37
Reference / hyperlink NLCD offset GRASS script in appendix 37
Figure: Facet map of cUSA NLCD offsets 37
hyperlink to section where MLCT was aggregated 38
Should the RMSE tables be rearranged? 38

Chapter 1

Introduction

1.1 Background

The continuing evolution and commoditization of high-performance computing infrastructure is constantly opening new horizons in spatial modeling of human/environment interactions. Increases in processing throughput, affordability of tera- and petabyte-scale storage resources, and ubiquity of parallelization tools and techniques create opportunities for formulating models of spatial processes of increasing extent, granularity, dimensionality, and complexity. The intersection of geography, economics, and computer science is a fertile frontier where researchers capable of harnessing the utility of available technology are presented with an unprecedented opportunity to contribute to resolving the urgent questions of our time regarding humankind's outlooks for survival, stewardship, and prosperity in coming decades and centuries. These issues generally revolve around characterizations of our manipulation of natural processes, notably food production; the side effects of those activities, being alterations of biogeochemical fluxes of matter and energy within and into the biosphere (Sellers et al., 1997); and the economic exchanges that mediate these activities as modulated by policy. Meaningful abstractions of these processes in the form of iterative, process-based models that we can formulate in order to derive descriptions of their dynamics and forecasts of their unfolding are not possible without some detailed, spatially explicit characterization of the ecological disposition of the earth's surface. This ecology is to be inclusive of human ecology, which is to say settlement, development, utilization, and transformation of natural resources. The general form of such a characterization is a land use/land cover (LULC) map which depicts landscapes according to categories of anthropogenic and natural phenomena (Fisher and Wadsworth, 2005). These maps are necessarily functions of history, climate, geology, hydrology and are formulated according to some design or convention with regard to their constituent types and their definitions, which make possible myriad representations of a given landscape regardless of scale. When conducting analysis in this space it is typically necessary to tailor the analysis to accommodate available data or create new data from raw physical measurements and observations, but a third option of fusing aspects of multiple available data sets is also available, as we will demonstrate

Warning–page numbers missing in Fisher2005a

here.

Arguably the most significant intersection of land use and land cover is agriculture. Agricultural activity has transformed all but the most inhospitable, impervious, and inaccessible corners of the globe and serves as a crucial underpinning of civilization, but is still an expression of variability in weather, soils, and biology, natural phenomena beyond humans' control, across the face of the earth. In the face of uncertainty regarding food security, availability of raw materials for industry and trade, impacts and dynamics of deforestation, desertification, and climate change, and sensitivity to these alarming trends due to a burgeoning global population, reliable forecasts of agricultural production and productivity over the long term are objects of much desire in the corridors of government, finance, and industry.

Recent years have seen a significant increase in the availability of global land cover data sets including the University of Maryland Global Land Cover Classification, Global Land Cover 2000 (GLC2000), and MODIS Land Cover Type (MLCT). At the regional level the National Land-cover Database (NLCD) provides high-resolution LULC data for the United States and Puerto Rico. These data sets are summarized in Table 1.1 with pertinent references and attributes of their collection. The proliferation of these data sets reflects the diversification and technological advances among space-borne sensors in recent years, resulting in improved resolution, both spatial and temporal, as well as innovation in post-processing and classification algorithms that transform raw sensor data into the thematic data that is readliy applicable to theoretical modeling.

Data set	Reference	Sensor	Resolution	Time S
UMD GLobal Land Cover 1998	Hansen et al. (2000)	AVHRR	1km	1981 –
Global Land Cover 2000	European Commission (2003); Bartholomé and Belward (2005)	SPOT	1km	Nov 19 posite)
National Landcover Database (NLCD)	Homer et al. (2004, 2007)	Landsat	30 m	2001
MODIS Land Cover Type v005	DAAC (2008); Friedl et al. (2010)	MODIS (Aqua & Terra)	500m	2001 nual ti

Table 1.1: Summary of global LULC data sets

Similarly there has also been a proliferation of data sets that describe the distribution and intensity of global agricultural activity. Some such as the Global Irrigated Areas Map (GIAM) (Thenkabail et al., 2008) and the Global Map of Rainfed Crop Areas (GMRCA) (Biradar et al., 2009) are the product of applying classification techniques to large collections of remote sensing and GIS data. Others such as Agricultural Land in the Year 2000 (Agland200) (Ramankutty et al., 2008), Harvested Area and Yields of 175 Crops (175Crops2000) (Monfreda et al., 2008), and the Spatial Production Allocation Model (SPAM) (You et al., 2006) are further informed by agricultural production data published at national and sub-national levels and disaggregated to grid cells within those boundaries according to an optimization method described by You and Wood

(2006). Data sets such as these have the potential to complement those of the general comprehensive LULC category by offering additional information on how to differentiate areas of cropland according to cultivars, and farming practices such as crop rotation, multiple cropping, and irrigation.

1.2 Objective

The Community Integrated Model of Economic and Resource Trajectories for Humankind (CIM-EARTH) project at the University of Chicago's Computation Institute, `http://www.cimearth.org/`, seeks to provide a framework in which to combine the best of modern computational and economic science to guide climate and energy policy. A major facet of this work involves forecasting of land use change over coming decades in the face of market pressures and hypothetical climate change scenarios. The supply side of this market analysis depends, among other industries, on agriculture. Prices of agricultural commodities are sure to change in years ahead in response to changes in technology, both of production itself and the products and materials that are derived from them, changes in aggregate demand for food and its attendant political ramifications, and changes to the environments where agricultural production occurs. Rents and prices of land will follow from the profitability, adaptiblity, and risks associated with the commodities that are possible to produce on it, as well as costs of energy and inputs needed to bring those goods to market. A spatially explicit model of not only agricultural production, but also the conversion of land into and out of active, profitable cultivation is needed in order to make statements about the magnitude, trend, volatility, and sustainability of agricultural output to guide decisions about investment and policy. We call this the Partial Equilibrium Economic Land-use (PEEL) model, which refers to the assumption of long-term demand trajectories as given inputs and calculates the likely distribution of production needed to meet that demand. The foundation of this modeling effort would have to be a LULC data set that is "complete" in the sense that it assigns all land plus coastal and inland water areas to one category or another, and that differentiates among crops to provide a modeling environment where shifts in production factor allocation can be driven by market and physical variables. None of the data sets considered so far exhibit these qualities; the LULC data sets treat cropland as a homogenous category and the agricultural maps do not depict other uses and covers. Hence the motivation to develop a hybrid data set that satisfies these criteria.

The mathematical properties of the PEEL model dictate a somewhat unconventional data model for representing the allocation of land area to the various LULC/crop categories. Rather than assigning individual grid cells to discrete categories as is typically done for LULC maps, PEEL is formulated in a subpixel analysis framework, such that for each cell a fraction is assigned to each category to represent the degree to which that LULC type is present across the

area of the grid cell. In a tabular representation the data would show cells in rows and the LULC types in columns with a constraint that the values in each row sum to unity. In terms of geospatial mapping this is equivalent to assigning a layer or band in a stacked image set to each category, as is done for spectral bands in radiometric data, and applying the same sum-to-one constraint to each pixel. ~~An advantage to this approach is that errors of false spatial precision in higher-resolution data sets from which our inputs are derived, meaning that pixels in the mother data set are aggregated into an expression of probability across the larger model grid cell versus the definite location implied by the thematic data. It also means that we will have an avenue for increasing model complexity and the raw size of the data set by a linear factor by increasing the depth of the data array rather than the quadratic increase that accompanies increases in resolution.~~ The primary purpose of this design choice is to strike a balance between locational specificity and a convenient accounting mechanism for land use conversion forecasts that would only confer false precision and impose additional computational burden if expressed spatially. In other words, the land area of a pixel is considered to be a single location whose internal arrangement is unspecified. The model can incorporate constraints governing the iterative transition of those fractions that are stated algebraically in order to exclude protected natural areas from conversion or require a degree of autocorrelation among neighbors to prevent unrealistic divergences in development patterns among grid cell neighborhoods, for example.

Does everyone agree that we can do without this passage?

A disadvantage of this data model is apparent when attempting to visualize the data. A thematic map can be viewed in a single pass given a well-designed palette that has a reasonable number of classes and the relative proportions and distributions of classes can be readily perceived by the viewer. For the sub-pixel data model a cognitive adjustment is necessary in order to consider multiple classes simultaneously. Although it is possible to employ the false-color approach typically used for viewing multispectral data, which is to map a subset of three bands to the red, green, and blue channels, this limits a given map to portraying three classes simultaneously, or else picking two of primary interest and lumping the remaining fractions into a catch-all category. This method is not quite as applicable to categorical data that we are discussing as it is to spectral data because a set of three spectral bands are typically left in long-to-short wavelength order and reassigned to red, green, and blue, which amounts to shifting their frequencies into the visible spectrum, in order to produce a false-color image. It would be difficult to interpret the mixing of thematic hues or the arbitrary assignment of categories to primary hues. The approach to visualization taken for this paper is to render maps in individual layers with a uniform palette to express the fractional expression of the classes and distinguish zero from null outside the set of pixels included by the analysis mask. Interpretation is aided by presenting these maps in collections called facets in Wilkinson's (2005) grammar of graphics to convey the full depth of

information in consideration.

Given that the CIM-EARTH modeling framework is in a prototype phase we are taking a conservative posture towards the degree of detail that we wish to capture in early applications. This is expressed by the choice of resolution of our model grid and the number of LULC categories, including crop sub-categories, to which each cell can be allocated. With an ultimate goal of running simulations at global extents we wanted to err on the side of prudence before measuring the computational requirements of processing time and storage of a working prototype. Early tests gauging the computational requirements for carrying out these simulations have indicated that operating on a $5'$ grid cell globally is not prohibitively costly in time, memory, or storage and that the design, implement, evaluate iterative development cycle can proceed at a satisfactory pace. This choice of resolution is not as arbitrary as it may seem given that it equates to roughly 10km at the equator and happens to be the same resolution as some of the base data employed in this exercise.

The algorithm described here will be peformed on the subset of the global 5-arc-minute grid that contain land area of the 48 contiguous states of the United States but is intended to be applied globally. As we will discuss in Chapter 2 when the base data sets are described in greater detail the MLCT is chosen as the foundation of this method because of its global coverage and greatest resolution among global data products. As the technique presented in Chapter 3 matures it will be applied globally and also extended in time to convert the proceeding years of the MLCT time series to a form useful in the PEEL model. This will be important for model validation to show that the model is capable of producing an evolution of the overall state of land use that corresponds to available observations. As we will see the necessary information needed to obtain a realistic distribution of areas for all classes is not currently available. We use the NLCD to complement MLCT for certain classes that are too small to resolve at 500m, hence the restricted extent for which this method is currently feasible. In Chapter 4 we wrap up with a discussion of the merits of this endeavor and propose future avenues of research based thereon.

At this time we are not aware of any other systematic attempt to incorporate the full depth of information offered by MLCT, which is a collection of three map layers: a primary cover class, a confidence level for that primary classification, and a secondary classification. Rather than interpret the secondary classification as the next most likely possibility we accept this triplet as an expression of the sub-pixel composition of that area. Aggregation of MLCT from 15 $''$ to $5''$ will blur the spatial precision implied by this formula and treat the local $20 \times 20 \times 3$ array as a probabilistic expression of the local landscape composition. We will show that this approach, given a principled assumption about the relationship between confidence level and sub-pixel area, that aggregate acreage estimates of the LULC classes, particularly cropland, are improved through this method. More on this in Section 2.1.

1.3 Reproducible Research

We maintain that the manner in which we execute this analysis is as significant, if not more so, to the practice of geospatial analysis as the product of the analysis itself. The second objective of this paper is to demonstrate the concept of reproducible resarch in geospatial analysis that has been made possible by a suite of open-source software tools. Previous to employing the suite of tools described below, our typical research experience with widely available GIS software, both free and commerical, is to conduct the analysis in a graphical user interface (GUI) environment and capture outputs for publication by manually exporting maps and charts as images and transcribing quantitative results from on-screen displays into the body of a document. Whenever an adjustment is made the maps, charts, tables, and quantities in the paper must be updated manually. The open-source GIS software package `GRASS` (GRASS Development Team, 2010) employs a command-line oriented interface as its basic mode of user interaction which makes recording of steps in an analysis in the form of a script a more approachable undertaking once the user develops familiarity with the necessary commands, but due to `GRASS`'s decades-long Unix heritage, this scripting is done using the Bash shell, a system that was designed primarily for system administration and suffers from a byzantine syntax and a dearth of native data structures, making succinct, expressive programming difficult.

The `R` statistical package addresses these shortcomings (R Development Core Team, 2010) by virtue of its design's orientation towards mathematical and statistical analysis. Using Robert Hijmans' (2011) `raster` package for `R` provides an interface for accessing and analyzing geospatial raster data sets without being forced to load the entire data set into memory, a constraint that has historically been the case with `R` data in general and made operations on large geospatial data sets difficult. Friedrich Leisch's (2002) `Sweave` package for `R` is a tool for embedding `R` code within a LaTeX (Lamport, 1994) document for inline code evaluation and dynamic injection of figures, tables, and text into a document prior to final typesetting. The utilization of these tools results in a software environment where the princples of reproducible research described and demonstrated by Gentleman and Temple Lang (2007) can be applied. An academic paper produced under this paradigm is analogous to a piece of open-source software where the majority of "users" will simply want the "compiled" version in the form of a PDF document, but the author also provides access to the the source code behind the production of that document for inspection, re-execution, and adaptation for follow-on research. This approach lowers the costs of reproduction and verification of scientific analyses, central tenets of the scientific method that have effectively fallen out of practice due to these costs. With the advent of software tools such as these this approach to documenting research has gained a foothold in numerous desciplines from statistics to medical imaging.

The tables, charts, and maps included in this document are generated by `R`

code which will be included as an appendix. The maps and charts are produced using Hadley Wickham's (2009) `ggplot2` package, employing the grammar of graphics mentioned above. David Dahl's `xtables` package is used to convert `R` data frames into tables marked-up for typesetting. `Sweave` itself provides a facility for injecting the results of evaluating arbitrary `R` expressions in the text body, making it possible to render pieces of data, such as total acreages, in a dynamic fashion within the body of the text.

The source code of this paper will be submitted on optical media to Northeastern Illinois University's Graduate college along with the final draft. It will also be available via GitHub at `https://github.com/nbest937/thesis`. The initial, intermediate, and final data products will be made available for download either through `http://www.ci.uchicago.edu/~nbest` and/or `http://www.cimearth.org/` by request to `mailto:nbest@ci.uchicago.edu` or `mailto:nbest@alum.mit.edu`.

s

Chapter 2

Data Sets

This chapter presents summary descriptions of the various data sets that are relevant to this analysis and further discussion on how they were manipulated in preparation for analysis. Operations where multiple data sets are used in conjunction are deferred to Chapter 3.

The general approach with the MLCT and NLCD data sets is to reclassify their categories, calculate per-pixel, per-class areas at the native resolutions, and aggregate the new classification to the $5'$ grid. The purpose of the reclassification is to reduce the number of classes and have a uniform set of classes across data sets. The challenge in this is that classification defintions are sometimes subtly different which makes direct comparison across data sets somewhat subjective, so we describe the mapping between original and simplified classifications. We apply and aggregation operation that calculates the relative proportion of each class in the new classification system present in each $5'$ grid cell according to the base data. In this process we convert classified maps whose pixels have discrete values to a stack of maps, one map per class, whose pixels have real number values on the interval $[0, 1]$ representing fractional areas and are constrained to sum to unity for each pixel through the stack. In the general case of the MLCT data product the process converts two discrete, thematic variables and one continuous variable, those being a primary covery type, a secondary cover type, and classification confidence level respectively, into a set of continuous variables representing fractional areas for the cover types in the siplified classification system. This general case is also compared to simpler cases of the NLCD and considering only the primary classification of MLCT. In these cases the process is simplified by considering only a primary thematic layer and performing the aggregation without a secondary cover type or confidence level by which to relate them but we are able to reuse the same functions for the raster calculations.

To illustrate the process of converting the these data sets from their original representation we are including maps of an area of southeastern Michigan to show greater detail through each step of the process. We chose this region for its diversity of land covers and uses, its relative diveristy of agricultural commodities across its significant cropland area, the significant presence of the mosaic class to illustrate our method for its decomposition and its familiarity

to our principal author, being his brthplace.

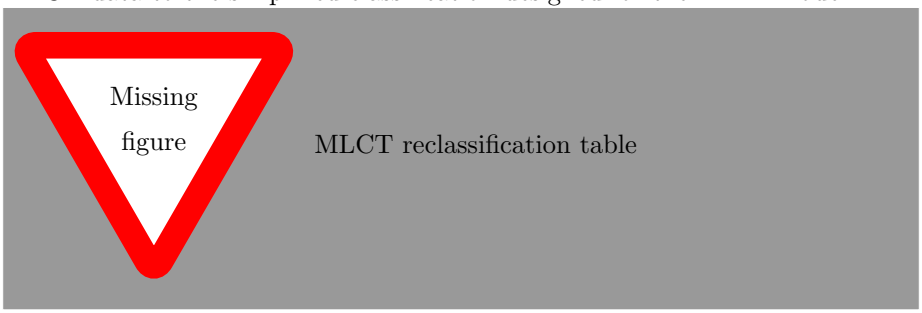
2.1 MODIS Land Cover Type (MLCT)

In preparation for this analysis we prepared the 2001 MLCT data by patching together the tiles as delivered in the equal-area sinusoidal projection, reprojecting that mosaic to geographic coordinates, and extracting a subset for the conterminous United States (cUSA). These preparation steps were carried out in a `GRASS` database prior to the adoption of the reproducible research framework for this paper, so those steps are not demonstrated here. The cUSA study area is defined as the set of $5'$ grid cells that intersect with the cUSA polygon in version 1 of the Global Administrative Areas (GADM) vector data set, which includes the water bodies on the American side of the international border across the Great Lakes, but does not extend to oceanic waters beyond the coastal grid cells that intersect with any land mass.

In this section we will demonstrate the process of converting the MLCT data from its native form, consisting of primary cover type, classification confidence for the primary cover, and secondary (alternate) cover type at $15''$ resolution, to a stack of cover fractions at $5'$ resolution using the simplified cover/use classification specified by the PEEL model.

2.1.1 Reclassification

The following table shows the mapping of the IGBP classes used in the original MLCT data to the simplified classification designed for the PEEL model.



Is this figure any better placed than others?

Figure 2.1 shows the result of reclassifying the MLCT data for our detailed study area. From this map we see that this area is dominated by the crop class in the north and the mosaic class to the south with scattered forests and pockets of development throughout. The urban complex of Port Huron, Michicagn and Sarnia, Ontario is visible in the southeast corner. along with the confidence level given for the primay classification.

In Figure 2.2 we notice that areas in the northern and central sections of the map that were classified as crop in the primary layer have null values in the

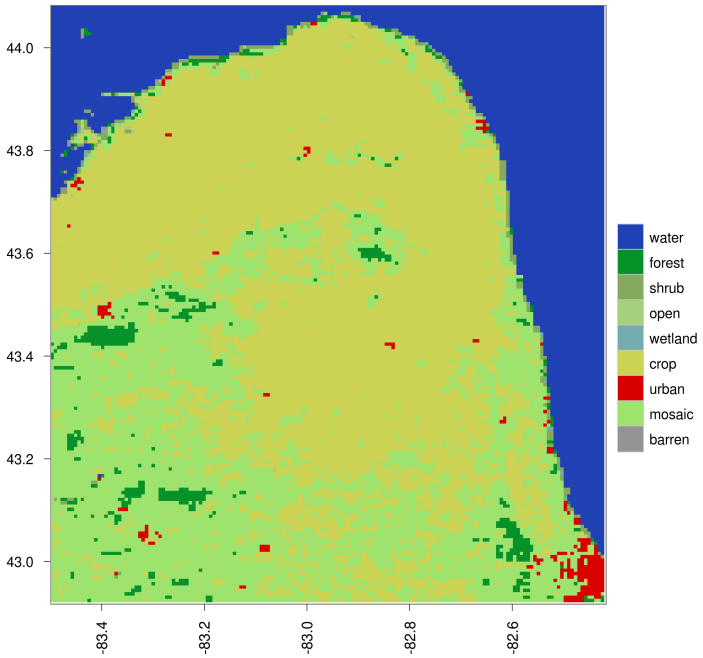


Figure 2.1: MLCT primary cover reclassified detail

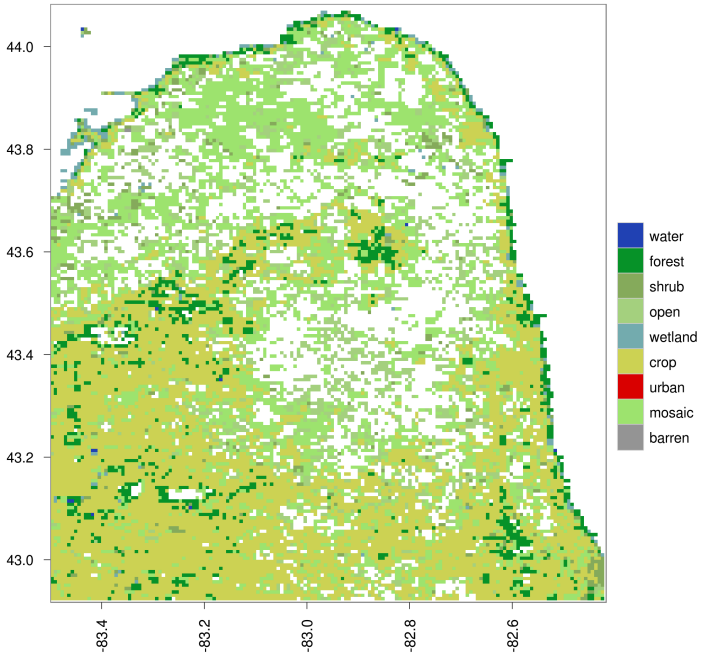


Figure 2.2: MLCT secondary cover reclassified detail

secondary class. It is apparent that where a secondary class is given that the mosaic class is often indicated where the primary class indicates cropland and vice versa. It is possible for primary and secondary classes to be assigned to the same category because of the reclassification step. When one of our pixels indiactes the forest class for both its primary and secondary classifications it simply reflects a distinction between sub-types of forest in the original data, for example evergreen and deciduous.

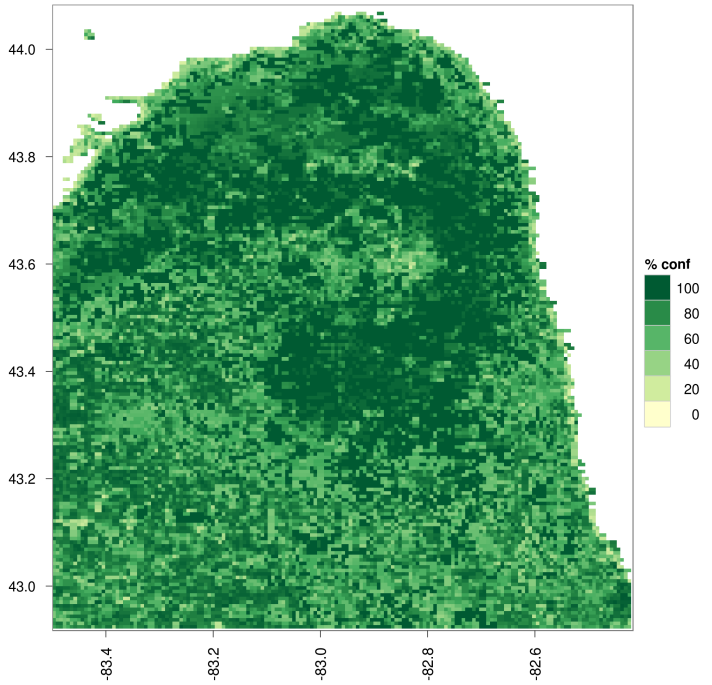


Figure 2.3: MLCT primary cover classification confidence

Figure 2.3 shows the confidence level as a percentage. We see that the areas where no secondary class is given are areas where confidence is 100% and the primary classification is cropland and therefore would be accounted as 100% cropland by area by any method of adding up these areas. In light of this observation it is clear that MLCT will generally over-estimate cropland because it is certain that these areas are not completely under cultivation but rather are interspersed with homesteads, fence lines, small wood lots, roads, and such cultural features. In areas such as this that were made available for settlement in the 19th century according to the Public Land Survey System (PLSS) we expect to find roads delineating every square mile in general.

The relationships described among the three layers of the MLCT are perhaps more easily appreciated visually by mapping the individual classes separately. Figure 2.4 does this for the primary class in our example detail area and Figure 2.5 for the secondary class.

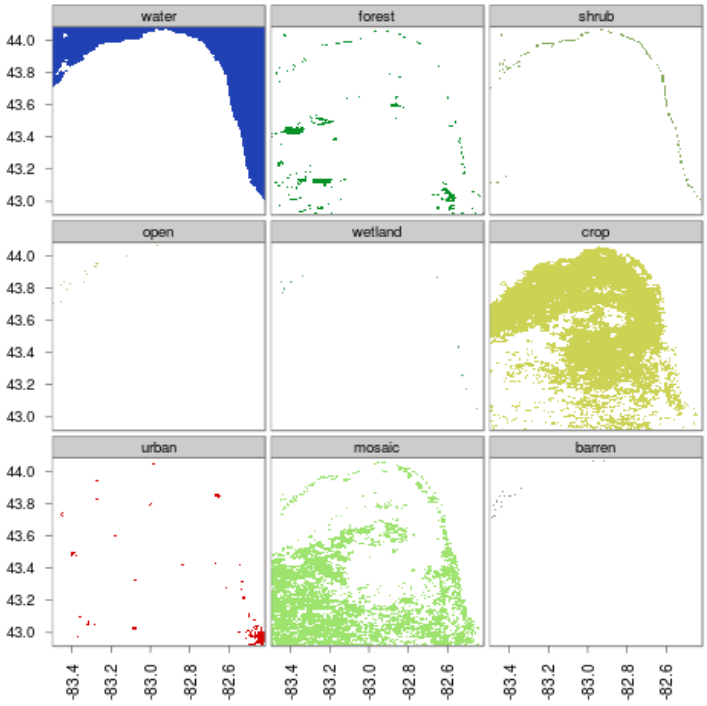


Figure 2.4: MLCT primary covers shown separately, detail

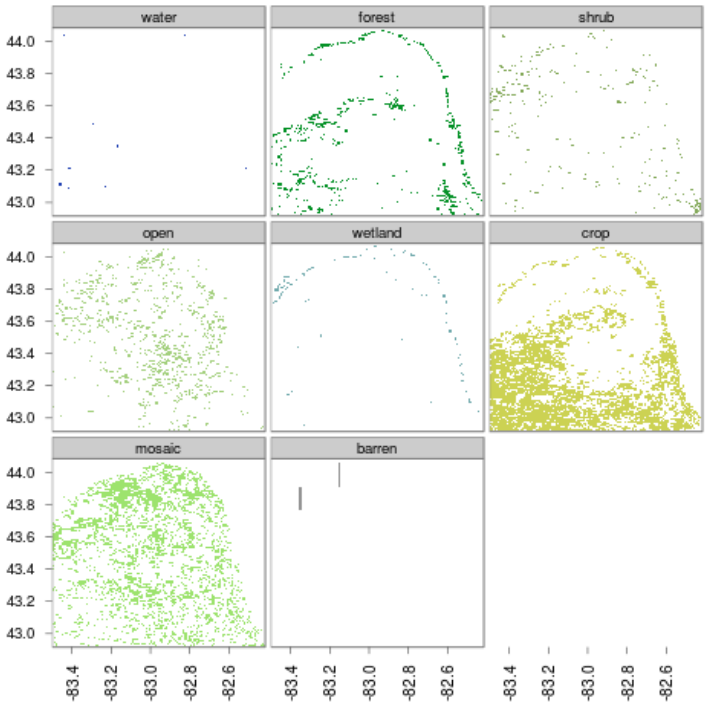


Figure 2.5: MLCT secondary covers shown separately, detail

Conveniently we are able to reuse the same functions for reclassification and mapping of the data that we have prepared for the larger study area. Figure 2.6 shows the map of the primary classification across the cUSA, and likewise Figure 2.7 for the secondary layer and Figure 2.8 for the confidence level. Because the maps are showing a greater extent in relatively the same amount of page space it is even more useful to create the facet maps for the individual classes as Figure 2.9 and Figure 2.10 have done. From these maps familiar generalities of the cUSA's geography are more apparent, such as the prevalence of forests in the east and northwest, cropland in the midwest, shrub lands in the southwest and open lands across the west. It is interesting to note that the mosaic class is primarily concentrated in the eastern portion of the study area which we can attribute to greater population density, topography, and historical patterns of settlement resulting in characteristically smaller parcels and a greater degree of mixing among agricultural uses and natural covers.

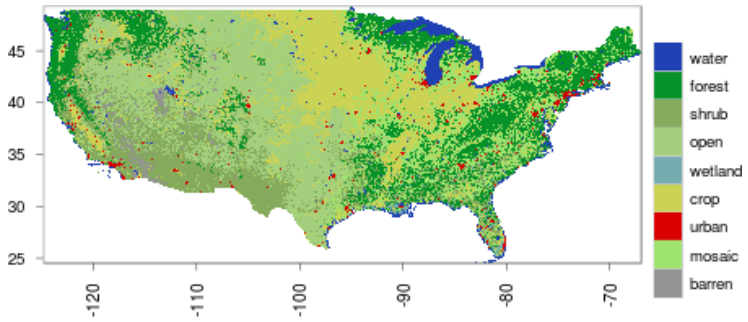


Figure 2.6: MLCT primary cover reclassified

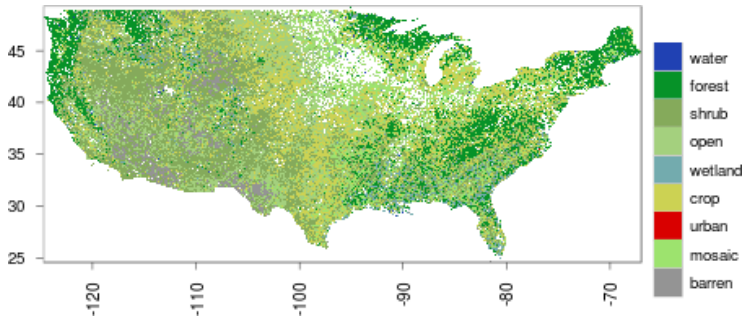


Figure 2.7: MLCT secondary cover reclassified

2.1.2 Aggregation

MLCT has a nominal resolution of 500m which roughly equates to 15″ at the eqautor and so is conveniently an even division of the 5′ grid to which we wish to aggregate it, the two related by a factor of 20. Therefore each cell in the output

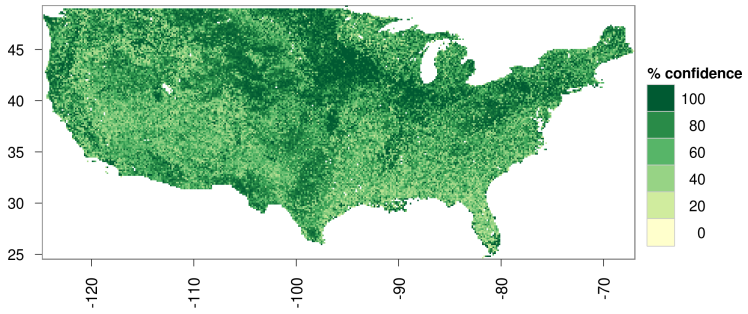


Figure 2.8: MLCT primary cover classification confidence

of this aggregation will be a function of the 400 orginal MLCT pixels within its footprint. The dataset consists of a primary classification, along with a measure of confidence up to 100%, and a secondary classification. The secondary cover type is given as the most likely alternative to the primary type (Friedl et al., 2010), but for purposes of our analysis we are taking a more probabilistic view and incorporating all available information from the base data. Because we are aggregating the data up to 5-arcmin resolution there is no expectation that the sub-pixel fractions at full resolution are spaitally specific, but in the aggregate our characterization of each grid cell's composition will be nuanced by this additional information. The primary class covers at least roughly 50-60% of a given pixel x, and this percent is almost certainly a monotonically increasing function of the confidence measure c. . For the purposes of this analysis we assume that this dependence is linear. Thus, for the primary and secondary cover types in a pixel:

cite email from Friedl

$$A_p(x) = A_{min} + (1 - A_{min})c(x)$$

$$A_s(x) = 1 - A_p(x)$$

consider including histograms showing confidence distribution

where $0.50 \leq A_{min} \leq 0.60$ is primarily chosen based on an interpretation of c. Given that there are only a handful of examples of $c < 0.20$, setting $A_{min} = 0.50$ is appropriate. Certainly for a classification to be considered the primary it must represent a bare majority of the area covered by that pixel at minimum, and the distributions of confidences indicate that the vast majority of pixels contain greater than 60% of their area in the primary under the rubric described above. The equations are simplified as follows by assuming this value for A_{min}.

$$A_p(x) = \frac{1 + c}{2}$$

$$A_s(x) = 1 - A_p(x) = \frac{1 - c}{2}$$

Applying these formulae results in a map for each cover type where the pixel

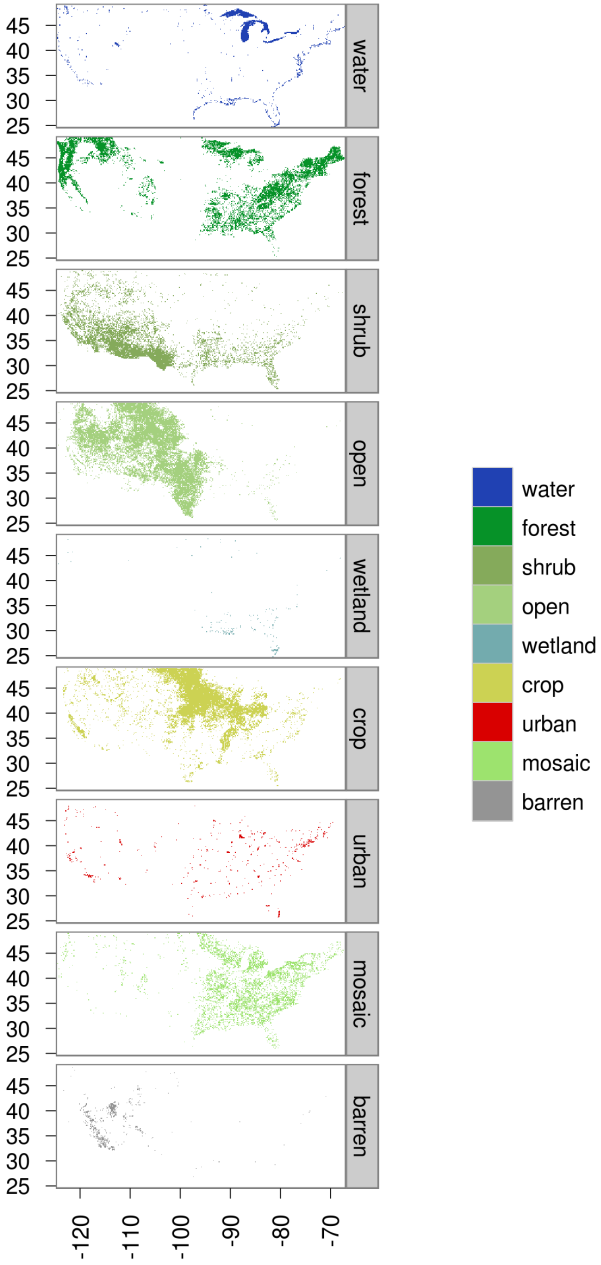


Figure 2.9: MLCT primary covers shown separately

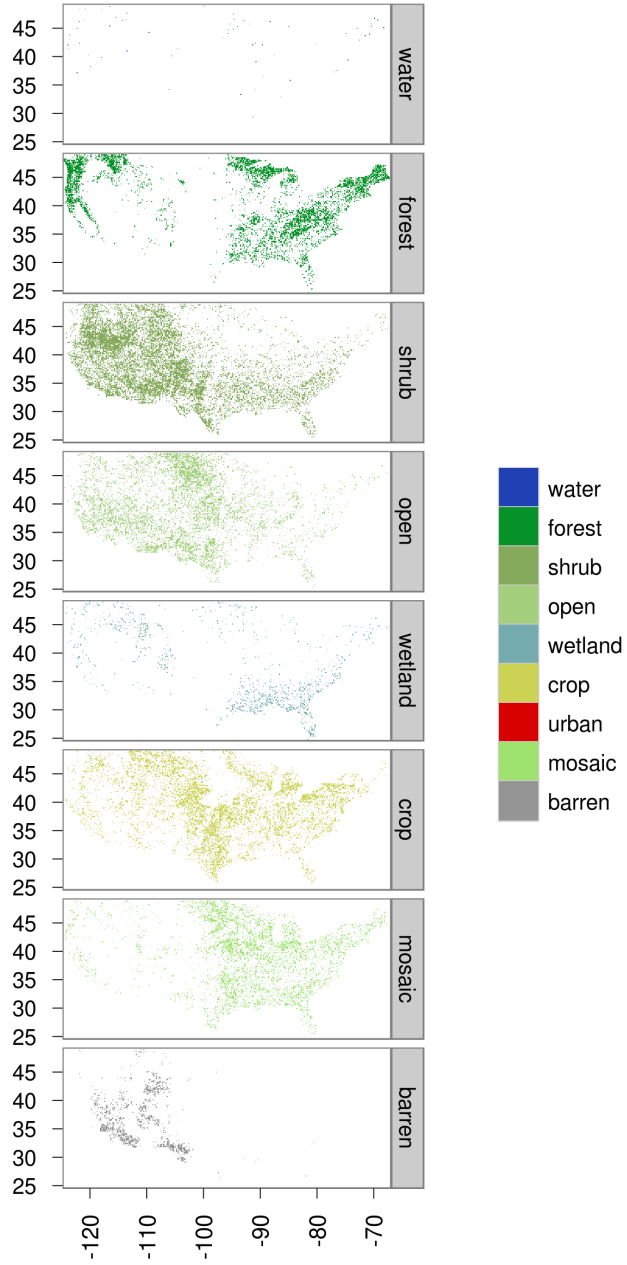


Figure 2.10: MLCT secondary covers shown separately

values are the sub-pixel areas on the interval $[0, 1]$. The map of the fraction of the primary cover type is visually equivalent to that of the classification confidence level because the former is simply a linear scaling and offset of the latter. Figure 2.11 shows the result of calculating $A_p + A_s$ for each individual class.

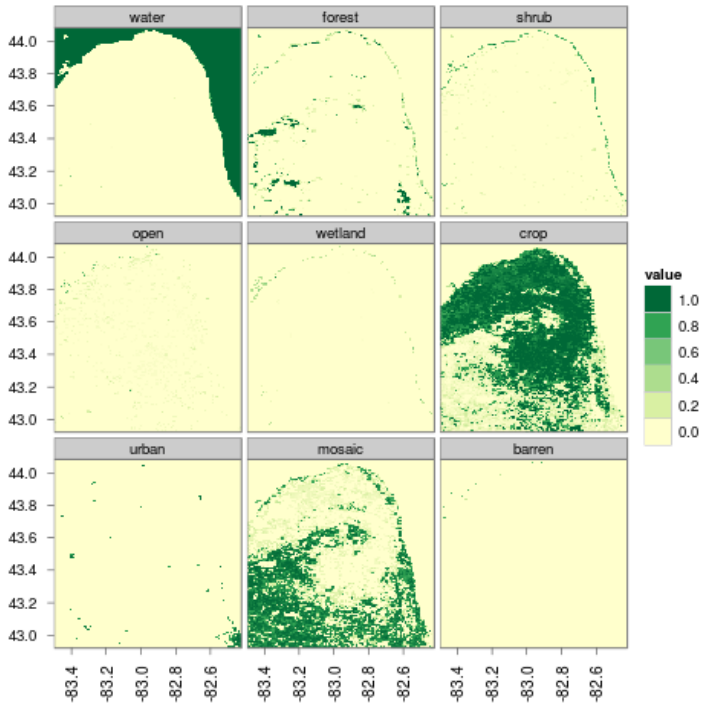


Figure 2.11: Sub-pixel fractions at original resolution for $A_{min} = 0.5$

By way of comparison we also consider the trivial case of settimg $A_{min} = 1$ which indicates that the secondary cover is ignored altogether and the primary cover is taken to represent 100% of the pixel area. Figure 2.12 shows these difference. The effect of adjusting A_{min} is subtle; we will examine it more closely after aggregating to the $5'$ grid.

Computationally the process of converting the reclassified maps to sub-pixel fractions at the desired 5-arcmin resolution is a three-step process. First we calculate the fraction of the primary cover type as a function of the classification confidence as described above. Next, a sub-pixel fraction for each cover type is calculated at full resolution, recognizing that the primary and secondary classes may be identical after the reclassification, such as cases where the original data indicated two different type of forests. Aggregating to a coarser resolution is a simple matter of calculating the mean of these values over the intersecting pixels at the original resoution. Because the desired $5'$ resolution is a multiple of the original $15''$ resolution the pixels are perfectly nested, which is convenient for properly computing this mean.

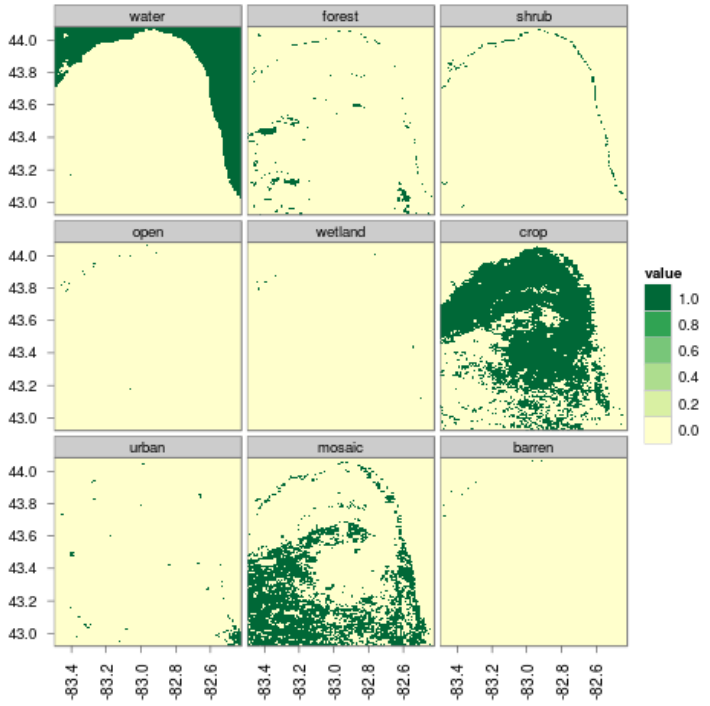


Figure 2.12: Sub-pixel fractions at original resolution for $A_{min} = 1$

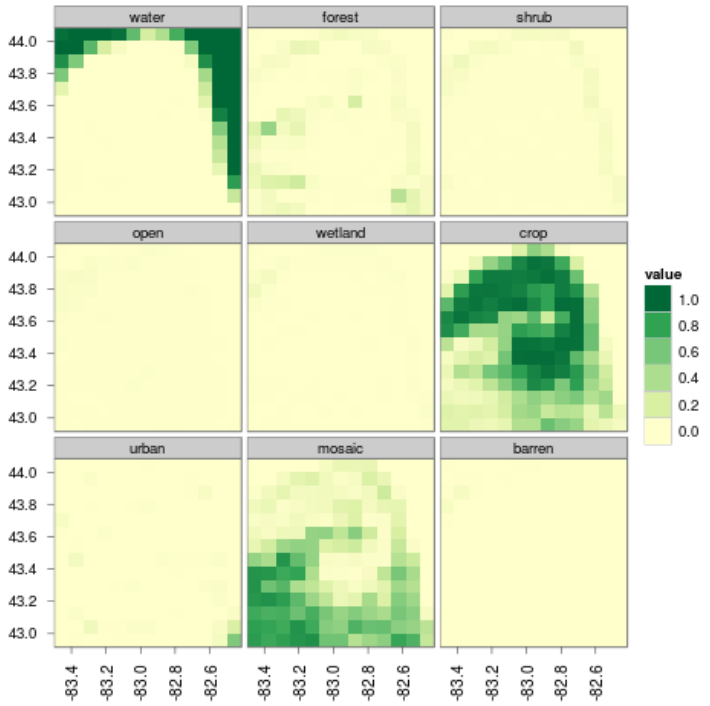


Figure 2.13: Aggregated sub-pixel fractions for $A_{min} = 0.5$

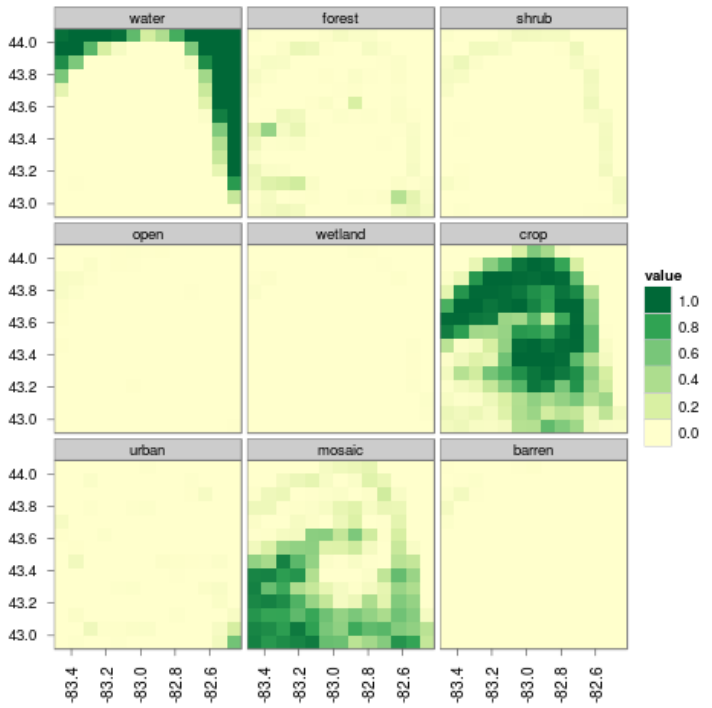


Figure 2.14: Aggregated sub-pixel fractions for $A_{min} = 1$

Before proceeding further it is interesting to inspect the differences between the aggregated maps for the chosen values of A_{min} as shown in Figure 2.15. Positive values indicate that $A_{min} = 0.5$ resulted in a greater fraction. The main message from this chart is that considering the secondary cover class results in greater mixture between the crop and mosaic classes because cropland is reduced in the north of the detail area where it was dominant in the primary land cover type, and simlarly for mosaic in the south. The relative suitability of these choices for A_{min} is discussed in chapter 3.

Figure 2.15 emphasizes the difference between the choice of $A_{min} = 0.5$ and $A_{min} = 1.0$ for the calculation of the sub-pixel fractions and their aggregation to $5'$ with a difference map. Positive values in the map indicate areas where $A_{min} = 0.5$ produced a greater value. We see more clearly from this set of maps that the effect of considering the secondary class results in a shift of up to 10% of total cell area from crop to mosaic in the north of our detail area and vice versa for the southern portion. This decrease in the relative dominance of the primary class is expected as we saw from the earlier maps (Figure 2.4 and Figure 2.2) of the MLCT data which classes were indicated by the secondary classes in those areas.

We apply the same functions for calculating the $15''$-resolution map of the primary cover class as a function of the confidence level c for the entire cUSA study area, converting those to per-class fractions at the same extent and scale,

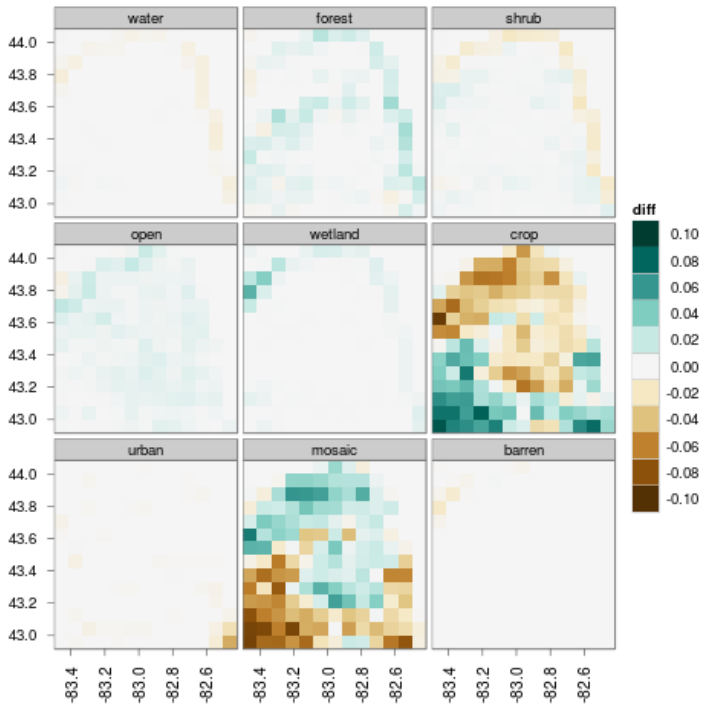


Figure 2.15: Difference of aggregated sub-pixel fractions

and aggregating those values to the 5′ grid. The corresponding figures are not shown because the decrease in relative resolution makes interpretation difficult. Based on the behavior that these functions exhibited over the detail area we can be confident that they will perform correctly over the greater extent.

2.1.3 Mosaic decomposition

The MLCT classification includes a type that is problematic for the economic models for which this data set is intended, the "cropland / natural vegetation mosaic" class. This class is defined as a hybrid of cropland and some mixture of natural covers (forest, shrub, or open) with no single component exceeding 60% (Friedl, 2002) and croplands generally comprising 40–60% of pixel area . Being a hybrid of developed land use and natural land cover we wish to differentiate the cropland from the natural vegetation in order to calculate a more meaningful total for cropland area and thereby eliminate the mosaic class from the final tabulation. In the present implementation of the reclassification and aggregation process we are making three very simple assumptions about the composition of area delineated as mosaic lands:

cite Friedl email

1. Mosaic land is 50% cropland.
2. The other 50% is a blend of forest, open, and shrub in proportion to the expression of those classes in the same 5-minute cell.

3. In the absence of such information we simply assume that the natural component of the mosaic is an equal blend of all three.

The intention here is to make simplifying assumptions that will allow us to proceed with the evaluation of this analytical framework. Although it may be interesting to vary the proportion used to calculate the proportion of mosaic land to be allocated to crop land we have no principled basis for this as of yet, considering that the defintion implies that this proportion is variable across the MLCT rather than being some unknown single-valued quantity. The choice of the 50% level reflects the assertion that the mosaic is a cultural class grouped with cropland and urban in the IGBP classification scheme without overstating the degree of development. MLCT provides adequate variability in this dimension by commonly pairing cropland and mosaic in the primary/secondary class data. The second assumption imposes that 15″ mosaic cells' non-crop portion will have the same relative composition of forest, open, and shrub as the as the non-mosaic portion of the 5′ grid cell in which it falls. Therefore mosaic pixels in a 5′ cell where only forest is found of the three non-crop mosaic components will be allocated 50% crop and 50% forest. Figure 2.16 and Figure 2.17 show the effect of decomposing the mosaic class in this fashion for A_{min} values of 0.5 and 1.0 respectively.

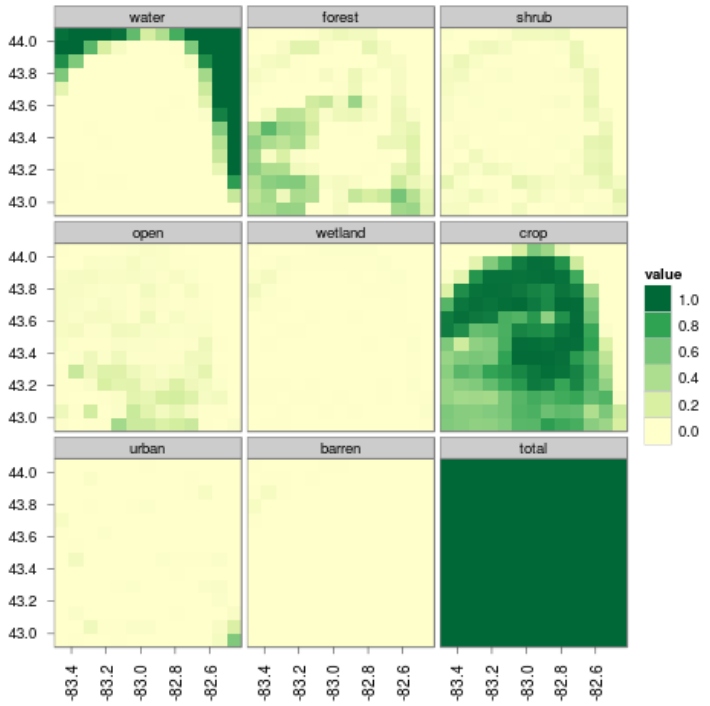


Figure 2.16: Aggregated cover fractions after mosaic decomposition, $A_{min} = 0.5$

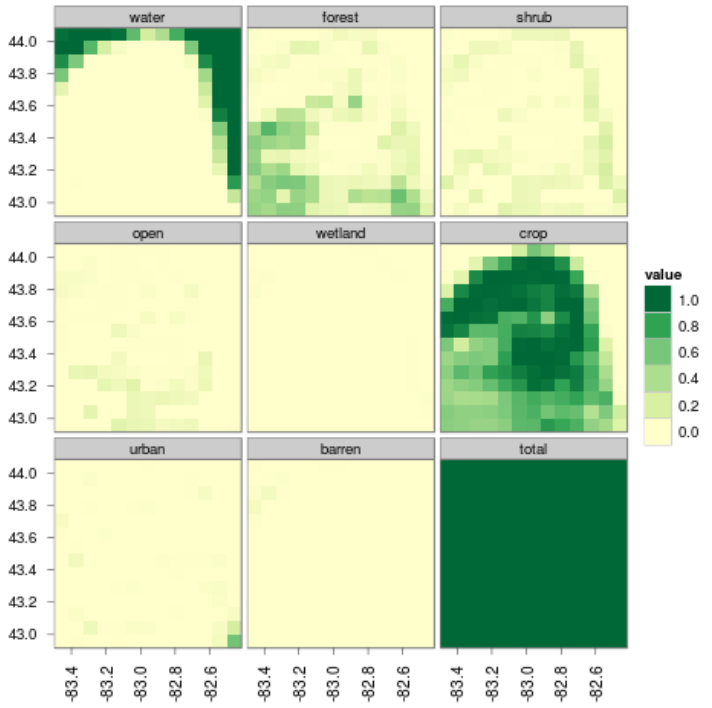


Figure 2.17: Aggregated cover fractions after mosaic decomposition, $A_{min} = 1.0$

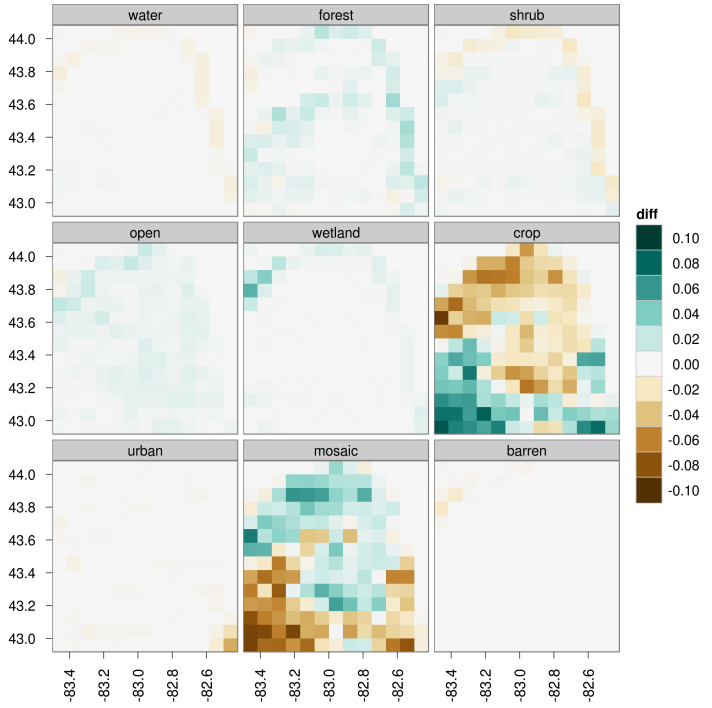


Figure 2.18: Differences of sub-pixel fractions after mosaic decomposition, positive when $f(A_{min} = 0.5)$ is greater

Our hypothesis from the outset is that there is information worth capturing in the secondary class and classification confidence level provided by MLCT. We will test this hypothesis in chapter 3 but in order to do so we need an "observed truth" to provide an independent standard by which to make a comparison on the basis of overall reduction in error at the $5'$ grid cell level. The following section describes such a data set which will be held up against these MLCT-derived data sets in the next chapter.

2.2 Agricultural Lands in the Year 2000 (Agland2000)

The data set described by Ramankutty et al. (2008), referred to in this paper as "Agland2000", is the product of an effort to merge satellite-derived LULC classifications with census data of agricultural actviity compiled at national or sub-national levels according to availability on around the turn of the last century. It uses both an older version of the MLCT (known as BU-MODIS) and the GLC2000 data set mentioned in section 1.1 and a mask based on climatic criteria and delineations of protected areas to allocate the census data to the $5'$ grid for both cropland and pasture. The "open" class in this data set has been renamed from "pasture" in its creator's nomenclature, but it is clear from its ditribution shown in Figure 2.20 that it represents a phenomena that is not apparent in the MLCT data, so we do not attempt to use it or reconcile it here, rather only carry it along to a small degree for sake of comparison. We attribute this discrepancy to commingling of managed pasture lands and natural open land in the MLCT classification. It is important to note that Agland2000 is used as an input into the classification algorithm of the version of MLCT that we are using here and acknowledge the possibility of circularity when comparing the two, but because of its basis in census data we will use the cropland component of Agland2000 as an "observed truth" for the purposes of evaluating our incremental adjustments to the maps we derive from MLCT in chapter 3.

2.3 National Land-cover Database 2001 (NLCD)

Homer et al. (2004)

The NLCD gives a higer-resolution (30m) snapshot of LULC circa 2001. Reclassifying and aggregating this data to 5-arcmin resolution in a fashion similar to that used for the MLCT is expected to give better estimations of aggregate area for detailed features like rural transportation networks and small stream and wetland features. This will compensate for MLCT's bias against these finely

check whether/how urban, water, wetland are informed with priors in NLCD

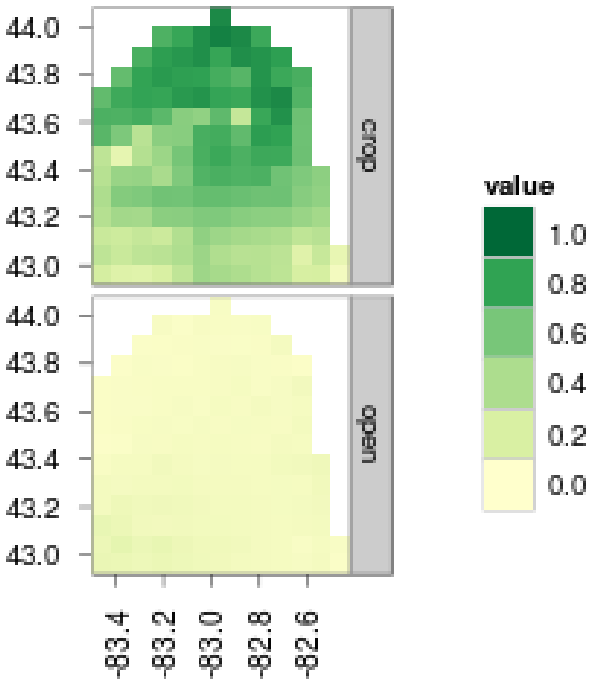


Figure 2.19: Agland2000 distribution in detail area

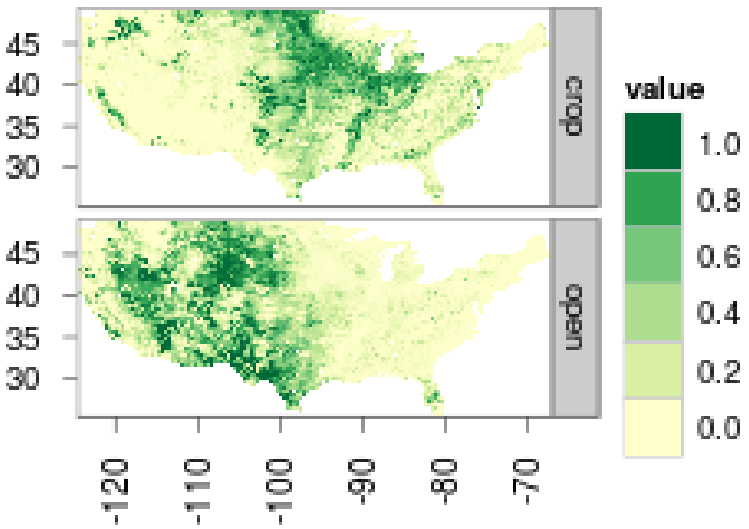


Figure 2.20: Agland2000 distribution in cUSA study area

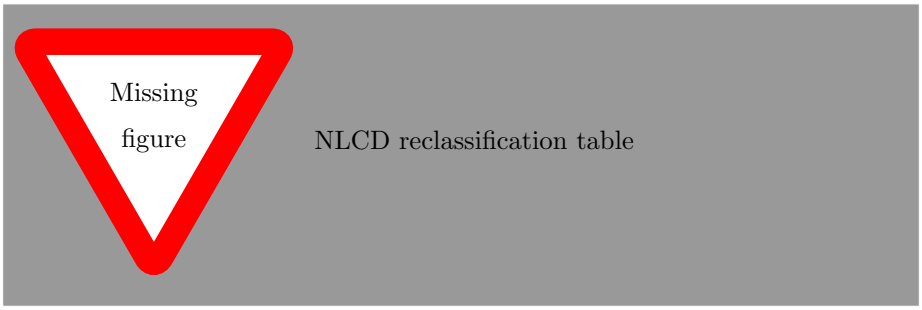
detailed structures due to it's resolution. It is the availability of this information that makes it difficult to apply this analysis beyond the United States without access to a comparable data set with global extents. The analysis is restricted to the conterminous US because of the relative paucity of agricultural activity in Hawaii and Alaska.

As with the MLCT the process of reclassification and aggregation is performed for both the detail region and the complete region.

One limitation of the `raster` library for `R` that we are using is that the aggregation function requires that the output resolution be a multiple of the output resolution. The 30m resolution of the NLCD equates to 1.25361″ and so does not satisfy this requirement. This deficiency was addressed by resampling the input to 1.25″ resolution prior to export from `GRASS` for this analysis using a nearest-neighbor sampling algorithm, which gives an even factor of 240 between the two resolutions.

Incorporate Joshua's suggestion to show further NLCD detail to better illustrate the discrepancy in developed areas

2.3.1 Reclassification



2.3.2 Aggregation

The same code used for refactoring the MLCT when considering only the primary cover type can be applied here.

Repeating this process for the entire study area is computationally expensive due to the NLCD's high resolution.

2.4 Harvested Area and Yields of 175 Crops (175crops2000)

Monfreda et al. (2008)

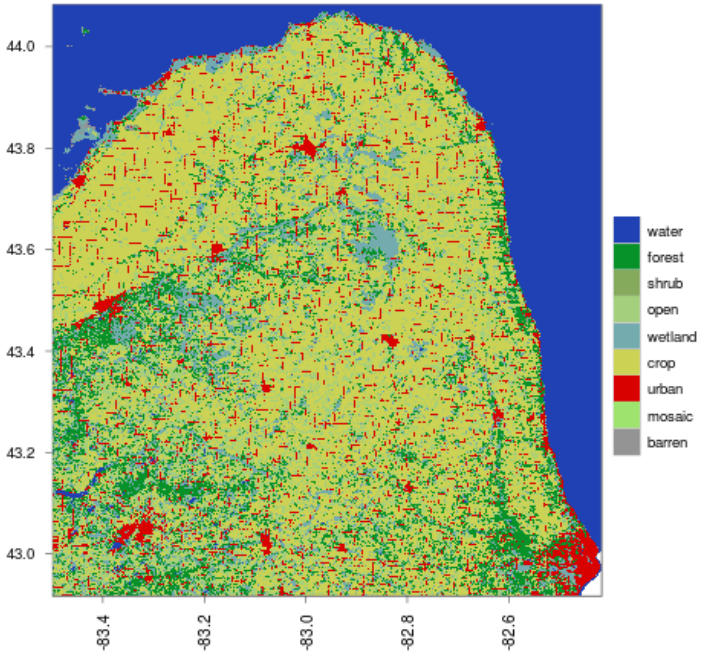


Figure 2.21: NLCD reclassified

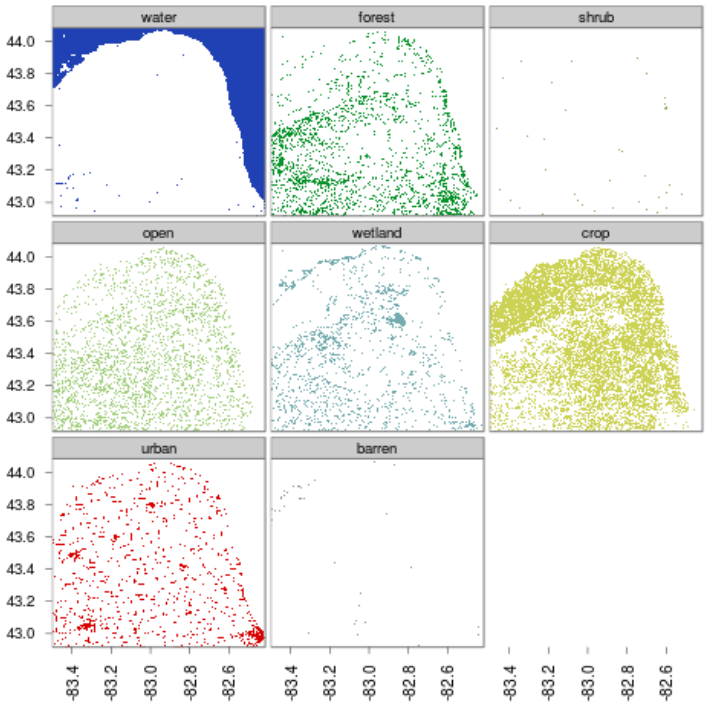


Figure 2.22: NLCD covers shown separately, detail

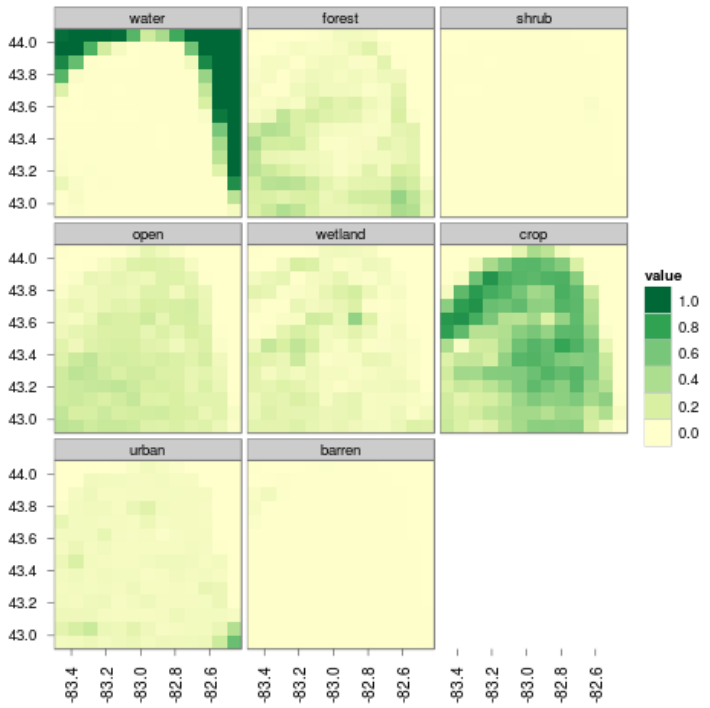


Figure 2.23: NLCD aggregated cover fractions, detail area

Missing figure

Table of crops and types reproduced from (Monfreda et al., 2008)

Missing figure

Summary table of crop aggregations for our model

Address issue of smaller land mask for 175crops2000 and Agland2000

This data set will provide the information needed to disaggregate the cropland area taken from Agland2000. It is not possible to use this data directly

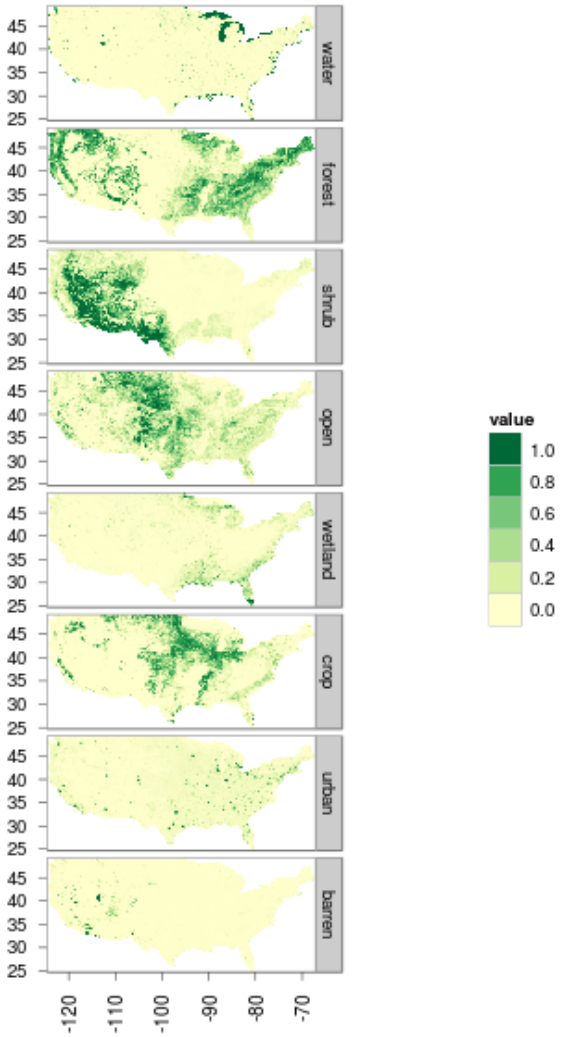


Figure 2.24: NLCD aggregated cover fractions

because it reflects only harvested area and so ignores various types of ancillary agricultural land, rather it will provide proportions for the disaggregation at the grid cell level. Rather than considering the full array of 175 crops we will consider only corn, soy, wheat, rice, and sugarcane individually, combine other cereals into their own class, and combine all remaining crops as a catch-all "other" category. Field crops will be distinguished from orchard / plantation crops that would likely fall under areas classified by MLCT as forest or shrub in this step.

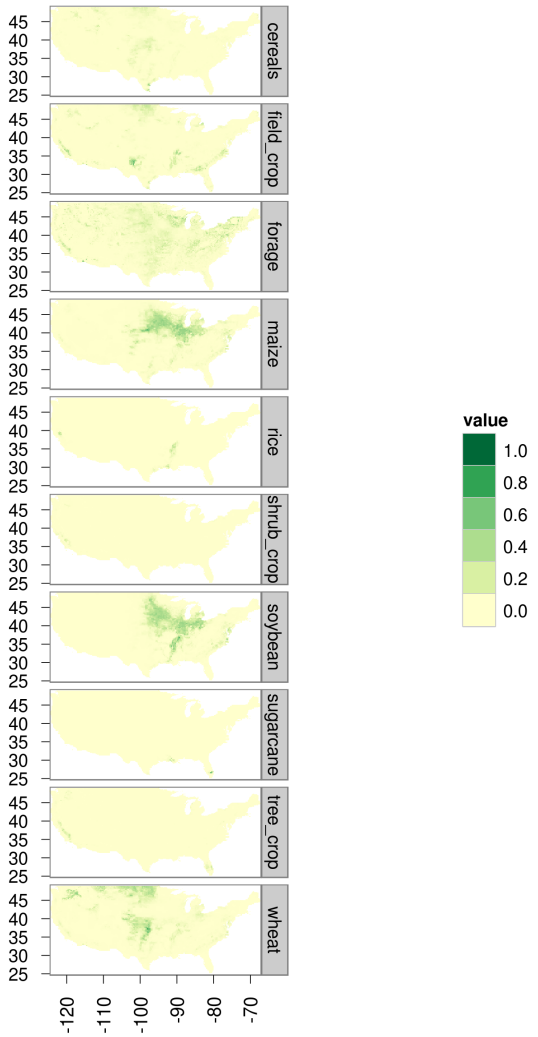


Figure 2.25: 175Crops2000 category maps

Chapter 3

Analysis

	Agland2000	NLCD	Aggregated $A_{min} = 0.5$	Aggregated $A_{min} = 1.0$	No Mosaic $A_{min} = 0.5$	No Mosaic $A_{min} = 1.0$
crop	446.5	310.8	378.9	369.6	495.4	488.1
open	557.1	429.6	516.9	545.8	538.7	561.9
barren	0.0	24.5	32.8	28.9	32.8	28.9
forest	0.0	513.2	344.7	353.6	410.8	429.9
shrub	0.0	420.1	358.7	341.8	387.2	368.0
urban	0.0	102.8	27.3	29.8	27.3	29.8
water	0.0	96.5	74.3	75.0	74.3	75.0
wetland	0.0	95.0	26.0	11.0	26.0	11.0
mosaic	0.0	0.0	232.9	237.0	0.0	0.0
total	1003.7	1992.5	1992.5	1992.5	1992.5	1992.5

Table 3.1: Total Acreages by Map and Cover

After decomposing the mosaic class The MLCT indicates 495.4Ma (200.5Mha) of cropland for $A_{min} = 0.5$ and 488.1Ma (197.5Mha) for $A_{min} = 1.0$ in the cUSA in 2001.

Pasture indicated by Aglands2000 appears to be a broader classification than that of the NLCD's pasture class because much of the grazing land east of the Mississippi river counted in the Aglands2000 pasture map is absent in the NLCD pasture class.

Aglands2000 indicates roughly Ma (Mha) of cropland. The inability of the MLCT data set to resolve rural transportation networks, minor settlements, and small water or wetland features is a major contribution to the surplus of cropland acreage indicated by the MLCT. Due to its greater resolution, 30m vs. 500m, the NLCD is better suited at discerning developed areas in rural landscapes ranging from rural roads to farmsteads to small communities that do not show up in the MLCT data. There is a total area of roughly 74 Ma (30 Mha) of development remaining after subtracting the MLCT urban class from all developed classes in the NLCD where the NLCD shows greater development after they have both been aggregated to the 5-arcmin grid. Applying this area as an offset to the cropland area in Aglands2000 brings us closer to the expected acreage under cultivation in 2001, although this assumes that all of that development intersects with MLCT cropland area.

The purpose for processing the MLCT for two values of A_{min} as described

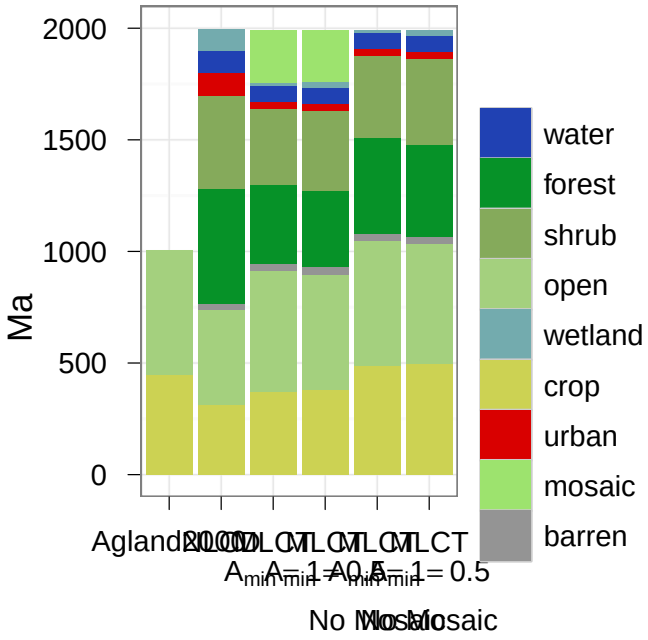
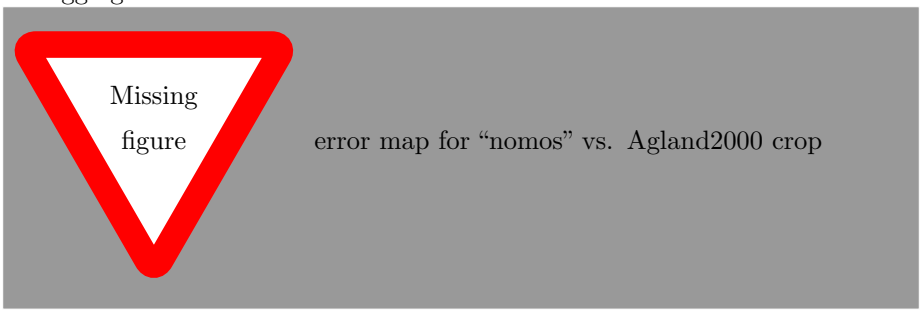


Figure 3.1: Total Acreages by Map and Cover

in the previous chapter is to evaluate whether or not information from the secondary cover type contributes positively to the accuracy of the data set we seek to synthesize. The primary objective of this synthesis is to achieve accuracy in cropland distribution. Because the cropland layer in the Agland2000 data set is derived from county-level production census statistics we adopt this as the ground truth and will endeavor to adjust our product accordingly. Although MLCT overstates cropland acreage for both $A_{min} = 0.5$ and $A_min = 1.0$ the discrimination among the two is made by the distribution of errors rather than the aggregate error.



These maps show the cell-by-cell differences between the MLCT-derived data set that we have calculated after mosaic decomposition and the Agland2000 cropland map. TO summarize and compare these errors we calculate the root

of the mean squared error (RMSE) given by:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{\theta}_i - \theta_i)^2}{n}}$$

where $\hat{\theta}_i$ are the predictions derived from the respective MLCT derivations and θ_i are the observations taken from the Agland2000 data set.

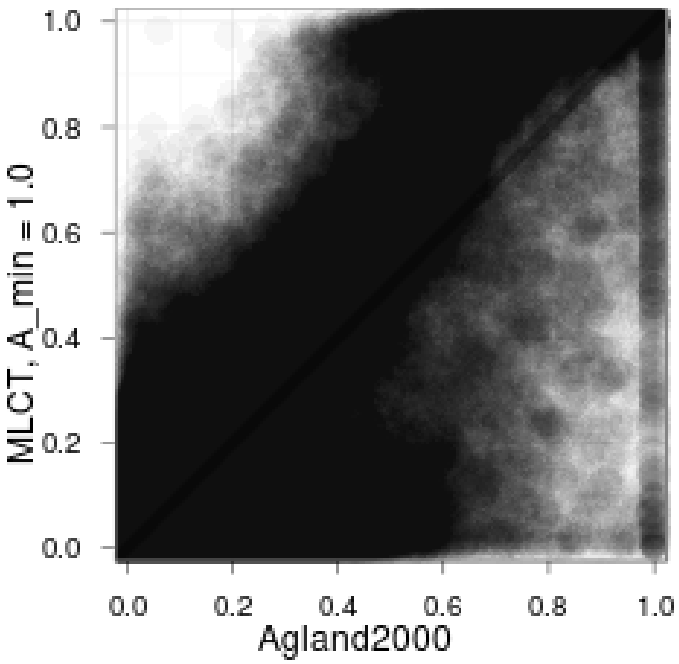


Figure 3.2: Scatter plot of MLCT crop ($A_{min} = 1.0$, no mosaic) versus Agland2000 cropland

A_{min}	RMSE
0.5	0.165
1.0	0.180

Table 3.2: RMSE, MLCT vs. Agland2000 crop

The results on Table 3.2 indicate that $A_{min} = 0.5$ is more representative of the distribution of cropland because although the total area indicated is higher there is less error on a cell-by-cell basis indicating that it does a better job of representing the spatial distribution than $A_{min} = 1.0$. Later when we recalculate the cell proportions by accepting the values for cropland area from Agland2000 as truth we can expect minimal distortion in reconciling its landscape with that given by MLCT. From this point forward we will consider only the statistics derived from setting $A_{min} = 0.5$ for the aggregation of the MLCT data due

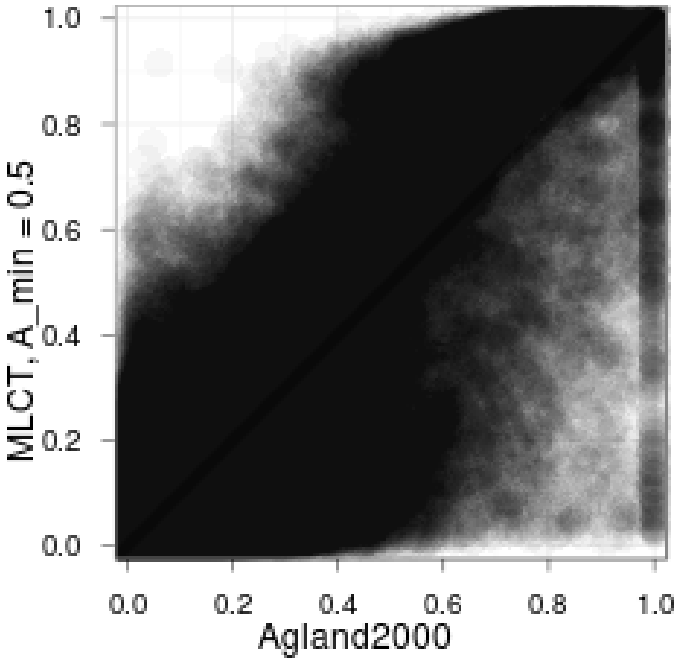


Figure 3.3: Scatter plot of MLCT crop ($A_{min} = 0.5$, no mosaic) versus Agland2000 cropland

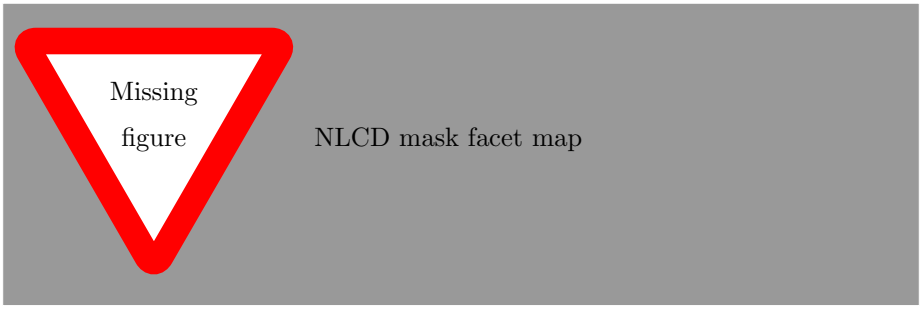
to this improved fit with Agland2000 cropland and its full consideration of all information imparted by the MLCT data.

3.1 NLCD Offsets

From Table 3.1 it is apparent that the MLCT results are negatively biased in the areas assigned to water, wetland, and urban features relative to the NLCD. It is clear from visual inspection that features of these classes tend to have smaller characteristic dimensions which causes them to be overlooked in the the MLCT classification. The most obvious example are the rural transportation networks in areas delineated by the Public Land Survey System (PLSS) where roads have been laid out on a regular grid of square miles called sections. In the PEEL classification this infrastructure is included in the urban class as another form of developed land.

The process of merging this information from the NLCD is as follows:

1. Create a mask comprised of pixels classified as water, wetland, or urban in the reclassified NLCD
2. Resample the MLCT layers to NLCD resolution ($\tilde{1}.25$ arcsecs) using this mask
3. Compute class-by-class offsets by accepting each NLCD pixel in the mask as a positive increment and each in the MLCT as a negative in proportion to the shares given by the formulas for A_{pri} and A_{sec}. Pixels outside the mask or where the data sets agree are assigned a zero value in this step.
4. Aggregate these offsets to 5-arcmin resolution by taking the mean of offset values across a given output grid cell
5. Add these offsets to the aggregated MLCT maps prior to the mosaic decomposition step
6. Recalculate the mosaic decomposition



MLCT resampled and masked facet(?) map(s?)

Facet map of thumb offsets

Difference map, thumb adjusted vs. original after mosaic decomposition

Due to performance constraints it was not possible to carry out this operation on the full cUSA study area. The equivalent operation of resampling the MLCT to the NLCD resolution of 1.25 arcsecs, calculating the offsets for water, wetland, and urban (developed) features implemented in a Bash script for use in the GRASS GIS environment is given in the appendix.

Reference / hyperlink NLCD offset GRASS script in appendix

The resulting offsets are added to the aggregated fractions calculated from the MLCT with $A_{min} = 0.5$.

Facet map of cUSA NLCD offsets

Following these algebraic acrobatics it seems prudent to check our accounting with some simple arithmetic. Working backwards from the final result of adding NLCD-derived offsets to the raster stack derived from the MLCT with $A_{min} = 0.5$ and decomposing the remaining mosaic fractions into their constituent cover types, subtracting the deltas that came from the mosaic decomposition, subtracting offsets calculated from the NLCD, and subtracting the aggregated MLCT data from the previous chapter should produce zeroes everywhere, plus or minus the noise of floating point math.

hyperlink to section where MLCT was aggregated

```
class       : RasterBrick
dimensions  : 298, 695, 9  (nrow, ncol, nlayers)
resolution  : 0.08333333, 0.08333333  (x, y)
extent      : -124.8333, -66.91667, 24.5, 49.33333  (xmin, xmax, ymin, ymax)
projection  : +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0
values      : in memory
min values  : -2.1e-10 -1.1e-16 -1.1e-16 -1.1e-16 -3.9e-09 -1.1e-16 -1.1e-16 -1.1e-16 -1.
max values  : 3.6e-11 1.9e-09 2.3e-09 2.5e-09 5.4e-11 5.1e-10 1.1e-16 1.8e-09 1.4e-09
```

	class	min	max
water	0	-2.09E-10	3.58E-11
forest	1	-1.11E-16	1.92E-09
shrub	2	-1.11E-16	2.32E-09
open	3	-1.11E-16	2.49E-09
wetland	4	-3.88E-09	5.36E-11
crop	5	-1.11E-16	5.05E-10
urban	6	-1.11E-16	1.11E-16
mosaic	7	-1.11E-16	1.85E-09
barren	8	-1.11E-16	1.35E-09

Table 3.3: Balance of adjustment fractions and original MLCT aggregation

To assess whether the process of adding in the NLCD offsets has improved overall cropland accuracy we can perform the same error calculation from above and extend Table 3.2 with the new result, giving us Table 3.4.

offset	A_{min}	$RMSE_{crop}$	$RMSE_{open}$
TRUE	0.5	0.150	0.235
FALSE	0.5	0.165	0.242
FALSE	1.0	0.180	0.267

Table 3.4: RMSE, MLCT vs. Agland2000 crop with NLCD offsets

Should the RMSE tables be rearranged?: Would it make more sense to have the row order and independent variables (first three) reversed in Table 3.2 and 3.4?

Seeing that this modifcation to the data set has improved our overall accuracy of the distribution of croplands the next step is to examine the total areas for all classes compared with the input data sets.

blah blah blah

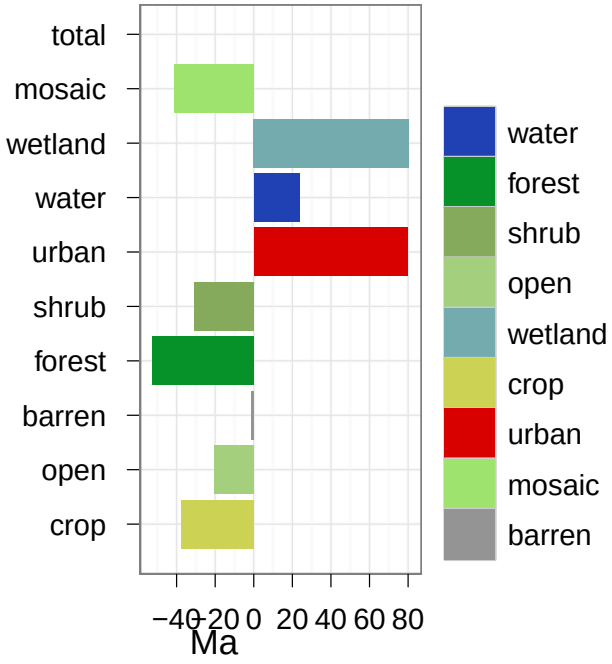


Figure 3.4: Total offsets calculated from NLCD

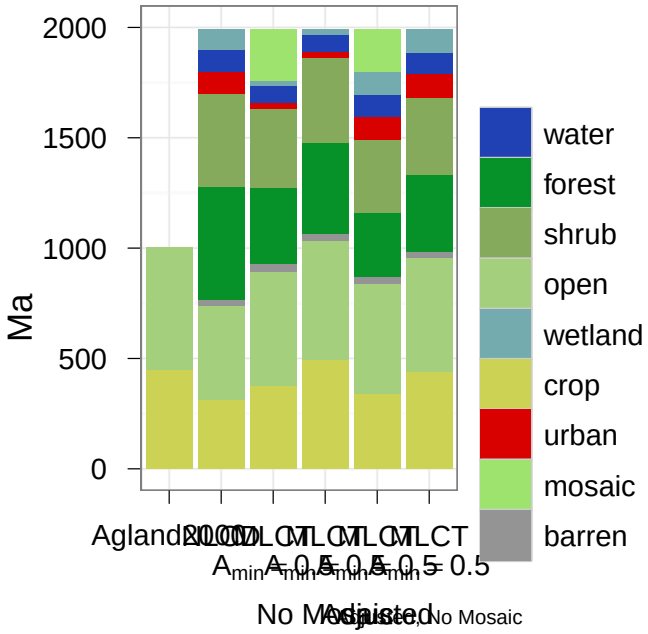


Figure 3.5: Area totals after NLCD adjustment

	Agland2000	NLCD	MLCT	MLCT No Mosaic	NLCD Offsets	MLCT Adjusted	MLCT Adjusted No Mosaic
water	0.0	96.5	74.3	74.3	23.7	98.0	98.0
forest	0.0	513.2	344.7	410.8	-52.4	292.3	346.4
shrub	0.0	420.1	358.7	387.2	-31.0	327.6	349.8
open	557.1	429.6	516.9	538.7	-20.6	496.3	515.9
wetland	0.0	95.0	26.0	26.0	80.5	106.5	106.5
crop	446.5	310.8	378.9	495.4	-37.3	341.6	437.5
urban	0.0	102.8	27.3	27.3	79.7	107.1	107.1
mosaic	0.0	0.0	232.9	0.0	-41.1	191.8	0.0
barren	0.0	24.5	32.8	32.8	-1.4	31.4	31.4
(all)	1003.7	1992.5	1992.5	1992.5	-0.0	1992.5	1992.5

Table 3.5: Effect of NLCD offsets on total acreages, $A_{min} = 0.5$

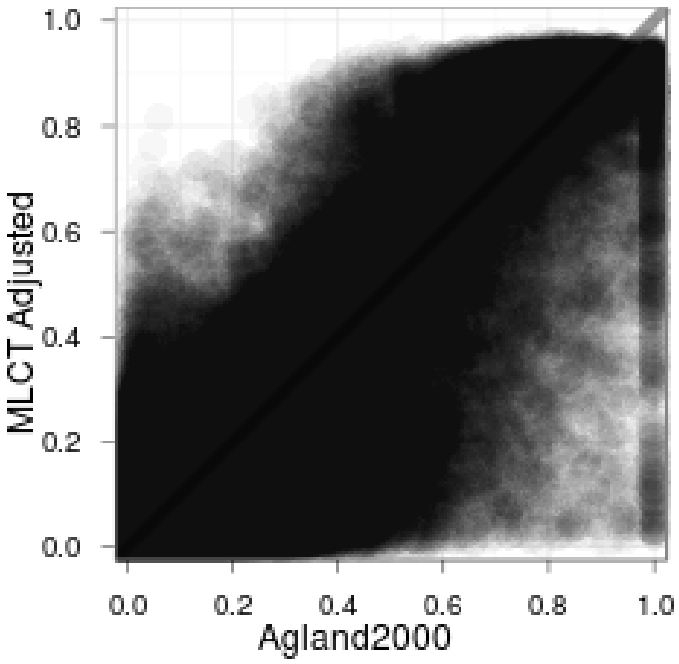


Figure 3.6: Scatter plot of MLCT adjusted crop versus Agland2000 cropland

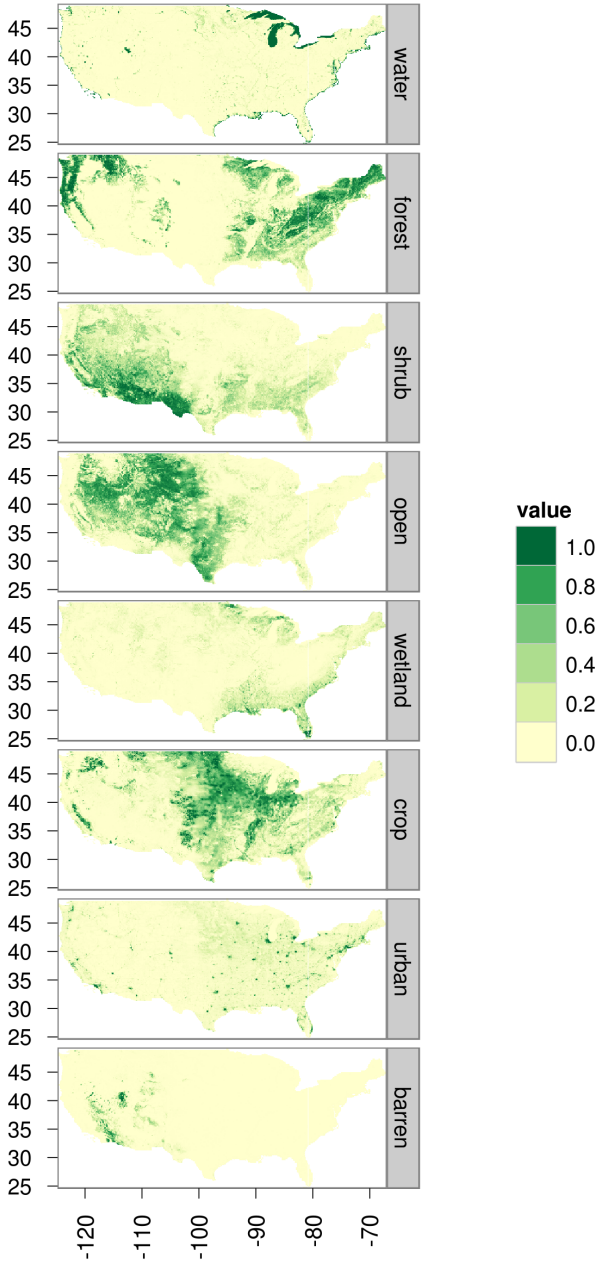


Figure 3.7: Agland Complete cover maps

Chapter 4

Conclusions

References

Bartholomé, E. and A. S. Belward (2005). Glc2000: a new approach to global land cover mapping from earth observation data. *International Journal of Remote Sensing 26*(9), 1959 – 1977.

Biradar, C. M., P. S. Thenkabail, P. Noojipady, Y. Li, V. Dheeravath, H. Turral, M. Velpuri, M. K. Gumma, O. R. P. Gangalakunta, X. L. Cai, X. Xiao, M. A. Schull, R. D. Alankara, S. Gunasinghe, and S. Mohideen (2009). A global map of rainfed cropland areas (gmrca) at the end of last millennium using remote sensing. *International Journal of Applied Earth Observation and Geoinformation 11*(2), 114 – 129.

DAAC, L. (2008). Modis land cover type (mlct, mcd12q1 v005). `https://lpdaac.usgs.gov/lpdaac/products/modis_products_table/land_cover/yearly_l3_global_500_m/mcd12q1`. These data are distributed by the Land Processes Distributed Active Archive Center (LP DAAC), located at the U.S. Geological Survey (USGS) Earth Resources Observation and Science (EROS) Center (lpdaac.usgs.gov).

European Commission, J. R. C. (2003). Global land cover 2000 database.

Fisher, P. F., A. J. C. and R. Wadsworth (2005). *Re-presenting GIS*, Chapter Land Use and Land Cover: Contradiction or Complement, pp. 85–98. John Wiley & Sons Ltd.

Friedl, M. (2002, November). Global land cover mapping from MODIS: algorithms and early results. *Remote Sensing of Environment 83*(1-2), 287–302.

Friedl, M. a., D. Sulla-Menashe, B. Tan, A. Schneider, N. Ramankutty, A. Sibley, and X. Huang (2010, January). MODIS Collection 5 global land cover: Algorithm refinements and characterization of new datasets. *Remote Sensing of Environment 114*(1), 168–182.

Gentleman, R. and D. Temple Lang (2007, March). Statistical Analyses and Reproducible Research. *Journal of Computational and Graphical Statistics 16*(1), 1–23.

GRASS Development Team (2010). *Geographic Resources Analysis Support System (GRASS GIS) Software, Version 6.4.0*. USA: Open Source Geospatial Foundation.

Hansen, M., R. DeFries, J. R. G. Townshend, and R. Sohlberg (2000). Global land cover classification at 1 km resolution using a decision tree classifier. *Int J Rem Sens 21*, 1331–1365.

Homer, C., J. Dewitz, J. Fry, M. Coan, N. Hossain, C. Larson, N. Herold, A. McKerrow, J. VanDriel, and J. Wickham (2007). Completion of the 2001 National Land Cover Database for the Counterminous United States. *Photogrammetric Engineering and Remote Sensing 73*(4), 337–341.

Homer, C., C. Huang, L. Yang, B. Wylie, and M. Coan (2004). Development of a 2001 National Land-Cover Database for the United States. *Photogrammetric Engineering Remote Sensing 70*(7), 829–840.

Lamport, L. (1994, July). *LaTeX: A Document Preparation System (2nd Edition)* (2 ed.). Addison-Wesley Professional.

Leisch, F. (2002). Sweave: Dynamic generation of statistical reports using literate data analysis. In W. Härdle and B. Rönz (Eds.), *Compstat 2002 — Proceedings in Computational Statistics*, pp. 575–580. Physica Verlag, Heidelberg. ISBN 3-7908-1517-9.

Monfreda, C., N. Ramankutty, and J. a. Foley (2008, March). Farming the planet: 2. Geographic distribution of crop areas, yields, physiological types, and net primary production in the year 2000. *Global Biogeochemical Cycles 22*(1), 1–19.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Ramankutty, N., A. T. Evan, C. Monfreda, and J. A. Foley (2008, January). Farming the planet: 1. Geographic distribution of global agricultural lands in the year 2000. *Global Biogeochemical Cycles 22*(1), 101029/.

Sarkar, D. (2008). *Lattice: Multivariate Data Visualization with R*. New York, NY: Springer New York.

Sellers, P. J., R. E. Dickinson, D. A. Randall, A. K. Betts, F. G. Hall, J. A. Berry, G. J. Collatz, A. S. Denning, H. A. Mooney, C. A. Nobre, N. Sato, C. B. Field, and A. Henderson-Sellers (1997, January). Modeling the exchanges of energy, water, and carbon between continents and the atmosphere. *Science 275*(5299), 502–509.

Thenkabail, P., C. Biradar, P. Noojipady, V. Dheeravath, Y. Li, M. Velpuri, G. Reddy, X. L. Cai, M. Gumma, H. Turral, J. Vithanage, M. Schull, and R. Dutta (2008). A Global Irrigated Area Map (GIAM) Using Remote Sensing at the End of the Last Millennium.

van Etten, R. J. H. . J. (2011). *raster: Geographic analysis and modeling with raster data*. R package version 1.7-35/r1520.

Wickham, H. (2009). *ggplot2: elegant graphics for data analysis*. Springer New York.

Wilkinson, L. and G. Wills (2005). *The grammar of graphics*. Statistics and computing. Springer.

You, L. and S. Wood (2006, October). An entropy approach to spatial disaggregation of agricultural production. *Agricultural Systems 90*(1-3), 329–347.

You, L., S. Wood, and U. Wood-Sichra (2006). Generating global crop distribution maps: from census to grid. In *Selected paper at IAEA 2006 Conference at Brisbane, Australia*, Number 202, pp. 1–16.

Source Code: Data Sets

R helper functions

```
library( raster)
##library( rgdal)
##library( lattice)
library( ggplot2)
##library( reshape2)
library( xtable)
##library( Defaults)

library( RColorBrewer)

library( foreach)
library(doMC)
registerDoMC( cores= 9)
#registerDoSEQ()

mlctList <- function( priFile, secFile, pctFile) {
                                        # creates a list
                                            of raster
                                            objects
                                        # stacking messes
                                             up the
                                            overlay
                                            functions
  priRaster <- raster( priFile)
  mlct <-
    list( pri= priRaster,
          sec= if( missing( secFile)) raster( priRaster)
              else raster( secFile),
          pct= if( missing( pctFile)) raster( priRaster)
              else raster( pctFile))
  ##sapply( mlct, setMinMax)
```

```
}


mlctReclassMatrix <-
  matrix( c( 0,  0,  0,
             1,  5,  1,
             6,  8,  2,
             9, 10,  3,
            11, 11,  4,
            12, 12,  5,
            13, 13,  6,
            14, 14,  7,
            15, 16,  8,
            253, 253, NA),
          ncol=3, byrow=TRUE)
peelClasses <- mlctReclassMatrix[ 1:9, 3]
names( peelClasses) <- c("water", "forest", "shrub", "
   open", "wetland", "crop", "urban", "mosaic", "barren")

#peelLegend <- igbpLegend[ mlctReclassMatrix[, 2] +1]


## just in case, save these for later
## paste( deparse( peelLegend), collapse="")

peelLegend <- c("#2041B3", "#069228", "#85AA5B", "#A4D07E
   ", "#73ABAE", "#CCD253", "#D90000", "#9DE36E", "
   #949494")
names( peelLegend) <- names( peelClasses)

nlcdReclassMatrix <-
  matrix( c( 11,  11,  0,              # water
             98,  99,  0,
             41,  43,  1,              # forest
             51,  52,  2,              # shrub
             94,  94,  2,
             71,  74,  3,              # open
             81,  81,  3,
             90,  93,  4,              # wetland
             95,  97,  4,
             82,  82,  5,              # crop
             21,  24,  6,              # urban
                                       # no mosaic
```

```
            12,  12,  8,                # barren
            31,  32,  8),
         ncol=3, byrow=TRUE)

mlctReclass <- function( mlct, reclassMatrix, overwrite=
    FALSE, ...) {
                                        # replaces
                                           primary and
                                           secondary
                                           rasters
                                        # but color
                                           tables are
                                           lost
  reclassFilename <- function( r) {
    parts <- unlist( strsplit( basename( filename( r)), "
        .", fixed=TRUE))
    paste( parts[ 1], "_reclass.tif", sep="")
  }
  if( overwrite) {
    mlct$pri <- reclass( mlct$pri, reclassMatrix,
                        filename= reclassFilename( mlct$
                            pri),
                        #datatype= "INT1U",
                        overwrite= TRUE, ...)
    if( "sec" %in% names( mlct)) {
      mlct$sec <- reclass( mlct$sec, reclassMatrix,
                          filename= reclassFilename( mlct
                              $sec),
                          #datatype= "INT1U",
                          overwrite= TRUE, ...)
    }
  } else {
    mlct$pri <- raster( reclassFilename( mlct$pri))
    if( "sec" %in% names( mlct))
          mlct$sec <- raster( reclassFilename( mlct$sec))
  }
  mlct
}

## overlayFunction <- function( pri, sec, pct) {
##   ifelse( is.na(pri), NA,
##          ifelse( is.na( sec), 1,
##                 ifelse( is.na( pct), Amin,
```

```
##                         Amin +( 1 -Amin) *pct /100)))
## }

## priFracOvFunc <- function( x) {
##   pri <- x[ 1]
##   sec <- x[ 2]
##   pct <- x[ 3]
##   ifelse( is.na(pri), NA,
##           ifelse( is.na( sec), 1,
##                   ifelse( is.na( pct), Amin,
##                           Amin +( 1 -Amin) *pct /100)))
## }


primaryFraction <- function( mlct, Amin=1.0, overwrite=
    FALSE, ...) {
                                          # appends an A_p
                                             raster to the
                                             MLCT list
                                          # and returns the
                                              appended list
  primaryFractionFile <-
    paste( deparse( substitute( mlct)), "Amin",
          paste( Amin, "tif", sep= "."),
          sep= "_")
  mlct$Amin <- Amin
  ## priFracOvFunc <- function( r) {
  ##   pri <- r[ 1]
  ##   sec <- r[ 2]
  ##   pct <- r[ 3]
  ##   ifelse( is.na(pri), NA,
  ##           ifelse( is.na( sec), 1,
  ##                   ifelse( is.na( pct), Amin,
  ##                           Amin +( 1 -Amin) *pct /100)))
  ## }
  priFracCalcFunc <- function( st) {
    pri <- st[ 1]
    sec <- st[ 2]
    pct <- st[ 3]
    ifelse( is.na(pri), NA,
           ifelse( is.na( sec), 1,
                  ifelse( is.na( pct), Amin,
                         Amin +( 1 -Amin) *pct /100)))
```

```
  }

  if( Amin <1 && overwrite)
    ## mlct$Ap <- overlay( mlct$pri, mlct$sec, mlct$pct,
    ##                     fun= priFracOvFunc, #
       overlayFunction,
    ##                     filename= primaryFractionFile,
    ##                     overwrite= TRUE,
    ##                     ...)
    mlct$Ap <- calc( stack(mlct$pri, mlct$sec, mlct$pct),
                      fun= priFracCalcFunc,
                      filename= primaryFractionFile,
                      overwrite= TRUE,
                      ...)
  else if( Amin <1 && !overwrite)
    mlct$Ap <- raster( primaryFractionFile)
  else mlct$Ap <- NULL
  mlct
}

## primaryOnlyFracsFun <-
##   function( mosaic) {
##     function( pri) {
##       v <- rep( ifelse( is.na( pri), NA, 0), times=
   length( peelClasses))
##       if( !is.na( pri)) v[ pri +1] <- 1
##       if( mosaic) v else v[ names( peelClasses) != "
   mosaic"]
##     }
##   }

    ## if( mosaic) b
    ## else b[ ,peelClasses[ names( peelClasses) != "
       mosaic"] +1]

    ##   res <- matrix( 0, nrow= length( pri), ncol= cols
       )
    ##   res[ is.na( pri),] <- rep( NA, times= cols)


    ## aaply( pri, 1,
    ##       function( x) {
    ##         ifelse( is.na( x), NA,
```

```
    ##                 ifelse( x == pri, 1, 0))
    ##       })}

coverFractions <- function( mlct, mosaic= TRUE, overwrite
   = FALSE, ...)  {
  Amin <- mlct$Amin
  mlctName <- deparse( substitute( mlct))
  classes <- peelClasses[ if( mosaic) 1:length(
     peelClasses)
                          else names( peelClasses) != "
                             mosaic"]
  fracsBrickFile <-
    if( Amin < 1)
      paste( mlctName,
             "Amin", mlct$Amin, "fracs.tif",
             sep="_")
    else
      paste( mlctName,
             "fracs.tif", sep="_")
  if( overwrite) {
    if( Amin < 1.0) {
      fracDoparFun <- function( priFilename, secFilename,
          ApFilename, ...) {
        foreach( cover= names( classes), .packages= "
           raster") %dopar% {
          class <- classes[[ cover]]
          frac <-
            calc( stack( raster( priFilename),
                         raster( secFilename),
                         raster( ApFilename)),
                 fun= function( st) {
                   pri <- st[ 1]
                   sec <- st[ 2]
                   Ap <- st[ 3]
                   res <- ifelse( is.na( pri), NA,
                                  ifelse( pri ==class, Ap,
                                      0)
                                  +ifelse( !is.na(sec) &
                                     sec ==class, 1 -Ap,
                                     0))
                   #if( res > 1 || res < 0) browser()
                   return( res)
                 },
```

```
                filename= paste( mlctName, cover, "Amin"
                    ,
                  paste( Amin, ".tif", sep=""),
                  sep="_"),
                overwrite= TRUE, ...)
          return( filename( frac))
        }
      }
    } else {
      fracDoparFun <- function( priFilename, ...) {
        foreach( cover= names( classes), .packages= "
           raster") %dopar% {
          class <- classes[[ cover]]
          frac <-
            calc( raster( priFilename),
                  function( pri) {
                    ifelse( is.na( pri), NA,
                           ifelse( pri ==class, 1, 0))
                  },
                  filename= paste(
                    mlctName,
                    paste( cover, ".tif", sep=""),
                    sep="_"),
                  overwrite= TRUE, ...)
          return( filename( frac))
        }
      }
    }
    mlct$fracs <-
      brick( stack( fracDoparFun( filename( mlct$pri),
                                  secFilename= filename(
                                     mlct$sec),
                                  ApFilename= filename(
                                     mlct$Ap),
                                  ...)),
            filename= fracsBrickFile,
            overwrite= TRUE,
            ...)
  } else {
    mlct$fracs <- brick( fracsBrickFile)
  }
  layerNames( mlct$fracs) <- names( classes)
  mlct
```

```
}


aggregateFractions <- function( mlct, aggRes= 5/60,
   overwrite= FALSE, ...) {
  aggBrickFile <-
    if( mlct$Amin < 1)
      paste( deparse( substitute( mlct)),
            "Amin", mlct$Amin, "agg.tif",
            sep="_")
    else
      paste( deparse( substitute( mlct)),
            "agg.tif", sep="_")
  mlct$agg <-
    if( overwrite)
      aggregate( mlct$fracs,
                fact= as.integer( round( aggRes /res(mlct
                   $fracs))),
                fun= mean,
                expand= FALSE,
                filename= aggBrickFile,
                overwrite= TRUE, ...)
    else
      brick( list.files( getwd(), patt=aggBrickFile,
                        full.names= TRUE))
  layerNames( mlct$agg) <- layerNames( mlct$fracs)
  mlct
}

peelBrickLayer <- function( peel, class) {
  peel[[ peelClasses[[ class]] +1]]
}

decomposeMosaic <- function( mlct, overwrite= FALSE, ...)
    {
  deltaBrickFile <- paste( deparse( substitute( mlct)),
                          "Amin", mlct$Amin, "delta.tif",
                          sep="_")
  nomosBrickFile <- paste( deparse( substitute( mlct)),
                          "Amin", mlct$Amin, "nomosaic.
                             tif",
                          sep="_")
  if( overwrite) {
```

```
overlayForest <- function( water, forest, shrub,
                           open, wetland, crop,
                           urban, mosaic, barren) {
  fso <- forest +shrub +open
  ifelse( fso ==0,
         forest +mosaic /6,
         forest *( 1 +mosaic /2 /fso))
}
overlayShrub <- function( water, forest, shrub,
                          open, wetland, crop,
                          urban, mosaic, barren) {
  fso <- forest +shrub +open
  ifelse( fso ==0,
         shrub +mosaic /6,
         shrub *( 1 +mosaic /2 /fso))
}
overlayOpen <- function( water, forest, shrub,
                         open, wetland, crop,
                         urban, mosaic, barren) {
  fso <- forest +shrub +open
  ifelse( fso ==0,
         open +mosaic /6,
         open *( 1 +mosaic /2 /fso))
}
overlayCrop <- function( water, forest, shrub,
                         open, wetland, crop,
                         urban, mosaic, barren) {
  crop +mosaic /2
}
mlct$nomos <-
  brick(
        peelBrickLayer( mlct$agg, "water"),
        overlay( mlct$agg, fun= overlayForest,
                filename= "newAgg_forest.tif",
                overwrite= TRUE),
        overlay( mlct$agg,
                fun= overlayShrub,
                filename= "newAgg_shrub.tif",
                overwrite= TRUE),
        overlay( mlct$agg,
                fun= overlayOpen,
                filename= "newAgg_open.tif",
                overwrite= TRUE),
```

```
            peelBrickLayer( mlct$agg, "wetland"),
            overlay( mlct$agg,
                    fun= overlayCrop,
                    filename= "newAgg_crop.tif",
                    overwrite= TRUE),
            peelBrickLayer( mlct$agg, "urban"),
            peelBrickLayer( mlct$agg, "barren"),
            overlay( mlct$agg,
                    fun= sum,
                    filename= "newAgg_total.tif",
                    overwrite= TRUE),
            filename= nomosBrickFile,
            overwrite= overwrite, ...)
    mlct$delta <-
     brick(
            peelBrickLayer( mlct$nomos, "forest") -
               peelBrickLayer( mlct$agg, "forest"),
            peelBrickLayer( mlct$nomos, "shrub") -
               peelBrickLayer( mlct$agg, "shrub"),
            peelBrickLayer( mlct$nomos, "open") -
               peelBrickLayer( mlct$agg, "open"),
            0 -peelBrickLayer( mlct$agg, "mosaic"),
            peelBrickLayer( mlct$nomos, "crop") -
               peelBrickLayer( mlct$agg, "crop"),
            filename= deltaBrickFile,
            overwrite= overwrite, ...)
} else {
  mlct$nomos <- brick( list.files( getwd(),
                                   patt= nomosBrickFile,
                                   full.names= TRUE,
                                   recursive= TRUE ))
  mlct$delta <- brick( list.files( getwd(),
                                   patt= deltaBrickFile,
                                   full.names= TRUE,
                                   recursive= TRUE))
}
layerNames( mlct$nomos) <-
  c( names( peelClasses)[ names( peelClasses) !="mosaic
      "],
     "total")
layerNames( mlct$delta) <- c( "forest", "shrub", "open"
    , "mosaic", "crop")
mlct
```

```
}


ggplotRaster <- function( r, samp) {
  df <- data.frame( as( sampleRegular( r, ncell( r)*samp,
                                       asRaster=TRUE),
                        "SpatialGridDataFrame"))
  ## names(df)[ 1:length( layerNames( r))] <- layerNames(
      r)
  ext <- extent( r)
  ggplot( data= df) +
    geom_tile( aes( x= s1, y= s2, fill= values)) +
    theme_bw() +
    scale_x_continuous( limits= c( ext@xmin, ext@xmax),
                        expand= c( 0,0)) +
    scale_y_continuous(  limits= c( ext@ymin, ext@ymax),
                        expand= c( 0,0)) +
    opts( panel.grid.minor= theme_blank(),
          panel.grid.major= theme_blank(),
          panel.background= theme_blank(),
              axis.title.x= theme_blank(),
               axis.text.x= theme_text( angle= 90, hjust
                  =1),
              axis.title.y= theme_blank())
}


peelMap <- function( r, samp) {
  p <- ggplotRaster( r, samp)
  p$data$values <- factor( p$data$values,
                           levels= peelClasses,
                           labels= names( peelClasses))
  p +
    geom_tile( aes( x= s1, y= s2,
                    fill= values)) +
    scale_fill_manual( "",
                      values= peelLegend,
                      breaks= names( peelClasses))
}

coverMaps <- function( r, samp=1, ...) {
  df <- data.frame( as( sampleRegular( r, ncell( r)*samp,
                                       asRaster=TRUE),
```

```
                        "SpatialGridDataFrame"))
  names( df)[ grep( "^values", names( df))] <- layerNames
      ( r)
  df <- melt( df, id.vars= c("s1", "s2"))
  ## name paraameters seem to have no effect -- need to
     upgrade?
  ## p <- p %+% reshape2::melt( p$data,
  ##                            id.vars= c("s1", "s2"),
  ##                            variable.name= "cover",
  ##                            value.name= "frac")
  ggplot( data= df) +
    geom_tile( aes( x= s1, y= s2, fill= value)) +
    theme_bw() +
    scale_x_continuous( expand= c( 0,0)) +
    scale_y_continuous( expand= c( 0,0)) +
    opts( panel.grid.minor= theme_blank(),
          panel.grid.major= theme_blank(),
          panel.background= theme_blank(),
              axis.title.x= theme_blank(),
               axis.text.x= theme_text( angle= 90, hjust
                  =1),
              axis.title.y= theme_blank()) +
    scale_fill_gradientn( colours= rev( brewer.pal( 6, "
       YlGn")),
                          limits= c( 1, 0),
                          breaks= seq( 1, 0, by= -0.2)) +
    facet_wrap(~ variable)
}

my.ggsave <- function(filename = default_name(plot),
                      height= 3.5, width= 3.5, dpi= 72,
                         ...) {
  ggsave(filename=filename, height=height, width=width,
      dpi=dpi, ...)
}

## coverDiffMaps <- function( r, samp= 1, ...) {
##   coverMaps( r, samp, ...) +
##     scale_fill_gradientn( colours= rev( brewer.pal(
   11, "BrBG")),
##                           limits= c( 1, -1),
##                           breaks= seq( 1, -1, by= -0.2)
   )
```

```
acreageTable <- function( rasterNames) {

  dataSets <- sapply( rasterNames,
                      function( n) eval( parse( text=n)))

  areas <- llply( dataSets,
                  function( d) {
                    res <- cellStats( d *acres, sum)
                    names( res) <- layerNames( d)
                    res
                  })

  areasDf <- ldply( areas, function( a) melt( t( as.data.
     frame( a))))

  areasCt <- cast( areasDf, X2 ~ .id, subset= X2 != "
     total", sum, margins="grand_row")
  rownames( areasCt) <- areasCt[, "X2"]
  areasCt <- areasCt[, -1]
  areasCt <- areasCt[ c( names( peelClasses), "(all)"),
     rasterNames]
}
```

R code embedded in the chapter

```
###################################################
### chunk number 1: initialize
###################################################
#line 14 "/home/nbest/thesis/datasets.Rnw"

                                          # load helper
                                             functions
                                          # code will
                                             appear in
                                             appendix
source("~/thesis/code/peel.R")
setwd( "~/thesis/datasets")

##setwd( "/gpfs/pads/projects/see/nbest/thesis/data")
##quartz.options( type="png")
```

```
overwriteRasters <- FALSE
overwriteFigures <- FALSE



#################################################
### chunk number 2: thumb
#################################################
#line 88 "/home/nbest/thesis/datasets.Rnw"

## this works but it's slow
##
## thumb <- crop( raster("2001_lct1.tif"),
##                extent(-83.5, -(82+25/60), 42+55/60,
    44+5/60))
##

## these are subsets exported from GRASS

texWd <- setwd("../data")
dataWd <- getwd()

thumb <- mlctList( "thumb_2001_lct1.tif",
                   "thumb_2001_lct1_sec.tif",
                   "thumb_2001_lct1_pct.tif")

igbpLegend <- thumb$pri@legend@colortable
igbpLegend <- igbpLegend[ igbpLegend != "#000000"]

## just in case, save these for later
##  paste( deparse( igbpLegend), collapse="")
## igbpLegend <- c("#2041B3", "#006A0F", "#007C25", "#00
    A25B", "#00A125", "#069228", "#9E9668", "#C1C48F",
    "#85AA5B", "#B1B741", "#A4D07E", "#73ABAE", "#CCD253",
     "#D90000", "#9DE36E", "#B6B5C2", "#949494")"



#################################################
### chunk number 3: mlct-reclass
#################################################
#line 124 "/home/nbest/thesis/datasets.Rnw"
```

```
thumb <- mlctReclass( thumb, mlctReclassMatrix, overwrite
    = overwriteRasters)

if( overwriteFigures) {
  thumbPlots <- list( pri= peelMap( thumb$pri, 0.4),
                      sec= peelMap( thumb$sec, 0.4))

  thumbPlots$pct <- ggplotRaster( thumb$pct, 0.4) +
    scale_fill_gradientn( "% confidence",
                          colours= rev( brewer.pal( 7, "
                             YlGn")),
                          limits= c( 100, 0),
                          breaks= seq( 100, 0, by= -20))
}
```

```
#####################################################
### chunk number 4: fig_thumb_pri_reclass
#####################################################
#line 153 "/home/nbest/thesis/datasets.Rnw"
setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_pri_reclass.png")
  print( thumbPlots$pri)
  dev.off()
}
```

```
#####################################################
### chunk number 5: fig_thumb_sec_reclass
#####################################################
#line 173 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_sec_reclass.png")
  print( thumbPlots$sec)
  dev.off()
}
```

```
##################################################
### chunk number 6: fig_thumb_pct
##################################################
#line 194 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_pct.png")
  print( thumbPlots$pct)
  dev.off()
}



##################################################
### chunk number 7: fig_thumb_pri_facet
##################################################
#line 222 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_pri_facet.png")
  print( thumbPlots$pri +
        facet_wrap(~ values) +
        opts( legend.position= "none"))
  dev.off()
}



##################################################
### chunk number 8: fig_thumb_sec_facet
##################################################
#line 245 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_sec_facet.png")
  print( thumbPlots$sec +
        facet_wrap(~ values) +
        opts( legend.position= "none"))
```

```
  dev.off()
}



##################################################
### chunk number 9: mlct
##################################################
#line 271 "/home/nbest/thesis/datasets.Rnw"

## repeat for cUSA
setwd( dataWd)
mlct <- mlctList( "2001_lct1.tif",
                  "2001_lct1_sec.tif",
                  "2001_lct1_pct.tif")
mlct  <- mlctReclass( mlct, mlctReclassMatrix, overwrite=
    overwriteRasters, datatype="INT1U", progress="text")

if( overwriteFigures) {
  mlctPlots <- list( pri= peelMap( mlct$pri, 0.01),
                     sec= peelMap( mlct$sec, 0.01))
  mlctPlots$pct <- ggplotRaster( mlct$pct, 0.01) +
    scale_fill_gradientn( "% confidence", colours=rev(
       brewer.pal( 7, "YlGn")),
                          limits= c( 100, 0),
                          breaks= seq( 100, 0, by= -20))
}



##################################################
### chunk number 10: fig_mlct_pri_reclass
##################################################
#line 296 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_mlct_pri_reclass.png")
  print( mlctPlots$pri + coord_equal())
  dev.off()
  system( "convert -trim fig_mlct_pri_reclass.png fig_
     mlct_pri_reclass_trim.png")
```

```
}



##################################################
### chunk number 11: fig_mlct_sec_reclass
##################################################
#line 318 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_mlct_sec_reclass.png")
  print( mlctPlots$sec + coord_equal())
  dev.off()
  system( "convert␣-trim␣fig_mlct_sec_reclass.png␣fig_
     mlct_sec_reclass_trim.png")

}



##################################################
### chunk number 12: fig_mlct_pct
##################################################
#line 341 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_mlct_pct.png")
  print( mlctPlots$pct + coord_equal())
  dev.off()
  system( "convert␣-trim␣fig_mlct_pct.png␣fig_mlct_pct_
     trim.png")
}



##################################################
### chunk number 13: fig_mlct_pri_facet
##################################################
#line 367 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
```

```
if( overwriteFigures) {
  png( file="fig_mlct_pri_facet.png")
  print( mlctPlots$pri +
        facet_wrap(~ values) +
        coord_equal() +
        opts( legend.position= "none"))
  dev.off()
}



##################################################
### chunk number 14: fig_mlct_sec_facet
##################################################
#line 391 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_mlct_sec_facet.png")
                                        # had to remove
                                           some NAs from
                                           the data
  mlctPlots$sec$data <- mlctPlots$sec$data[ !is.na(
     mlctPlots$sec$data$values),]
  print( mlctPlots$sec +
        facet_wrap(~ values) +
        coord_equal() +
        opts( legend.position= "none"))
  dev.off()
}



##################################################
### chunk number 15: thumbPlots
##################################################
#line 489 "/home/nbest/thesis/datasets.Rnw"


setwd( dataWd)

## calculate cover fractions and aggregate for detail
   area
```

```
thumb  <- primaryFraction( thumb, Amin=0.5, overwrite=
   overwriteRasters, progress= "text")
thumb1 <- primaryFraction( thumb, Amin=1.0, overwrite=
   overwriteRasters, progress= "text")
thumb  <- coverFractions( thumb, overwrite=
   overwriteRasters, progress= "text")
thumb1 <- coverFractions( thumb1, overwrite=
   overwriteRasters, progress= "text")
thumb  <- aggregateFractions( thumb, overwrite=
   overwriteRasters, progress= "text")
thumb1 <- aggregateFractions( thumb1, overwrite=
   overwriteRasters, progress= "text")

## seems like brick() has a bug such that the filenames
   have no paths
## thumb$fracs <- brick( paste( getwd(), filename( thumb$
   fracs), sep="/"))
## thumb1$fracs <- brick( paste( getwd(), filename(
   thumb1$fracs), sep="/"))
## thumb$agg <- brick( paste( getwd(), filename( thumb$
   agg), sep="/"))
## thumb1$agg <- brick( paste( getwd(), filename( thumb1$
   agg), sep="/"))

if( overwriteFigures) {
  thumbPlots <- list( fracs= coverMaps( thumb$fracs, 0.4)
     ,
                     agg= coverMaps( thumb$agg, 1))
  thumbPlots1 <- list( fracs= coverMaps( thumb1$fracs,
     0.4),
                      agg= coverMaps( thumb1$agg, 1))
}



##################################################
### chunk number 16: fig_thumb_fracs
##################################################
#line 521 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
```

```
  png( file="fig_thumb_fracs.png")
  print( thumbPlots$fracs)
  dev.off()
}



##################################################
### chunk number 17: fig_thumb1_fracs
##################################################
#line 541 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb1_fracs.png")
  print( thumbPlots1$fracs)
  dev.off()
}



##################################################
### chunk number 18: fig_thumb_agg
##################################################
#line 562 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_agg.png")
  print( thumbPlots$agg)
  dev.off()
}



##################################################
### chunk number 19: fig_thumb1_agg
##################################################
#line 582 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb1_agg.png")
```

```
  print( thumbPlots1$agg)
  dev.off()
}



##################################################
### chunk number 20: thumbAggDiff
##################################################
#line 603 "/home/nbest/thesis/datasets.Rnw"

setwd( dataWd)

thumbAggDiff <-
  if( overwriteRasters) {
    overlay( thumb$agg, thumb1$agg,
            fun= function( t, t1) t -t1,
            filename= "thumb_agg_diff.tif",
            overwrite= TRUE)
  } else brick( "thumb_agg_diff.tif")
layerNames( thumbAggDiff) <- layerNames( thumb$agg)

if( overwriteFigures) {
  thumbAggDiffPlot <- coverMaps( thumbAggDiff) +
    scale_fill_gradientn( "diff", colours= rev( brewer.
       pal( 11, "BrBG")),
                         limits= c( 0.1, -0.1),
                         breaks= seq( 0.1, -0.1, by=
                            -0.02))
}



##################################################
### chunk number 21: fig_thumb_agg_diff
##################################################
#line 628 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_agg_diff.png")
  print( thumbAggDiffPlot)
  dev.off()
```

```
}



################################################
### chunk number 22: thumbNomos
################################################
#line 686 "/home/nbest/thesis/datasets.Rnw"

setwd( dataWd)
thumb  <- decomposeMosaic( thumb, overwrite=
    overwriteRasters, progress= "text")
thumb1 <- decomposeMosaic( thumb1, overwrite=
    overwriteRasters, progress= "text")

## thumb$nomos <- brick( paste( getwd(), filename( thumb$
    nomos), sep="/"))
## thumb1$nomos <- brick( paste( getwd(), filename(
    thumb1$nomos), sep="/"))
## thumb$delta <- brick( paste( getwd(), filename( thumb$
    delta), sep="/"))
## thumb1$delta <- brick( paste( getwd(), filename(
    thumb1$delta), sep="/"))

if( overwriteFigures) {
  thumbPlots$nomos <- coverMaps( thumb$agg, 1)
  thumbPlots1$nomos <- coverMaps( thumb1$agg, 1)
}

thumbNomosDiff <-
  if( overwriteRasters) {
    overlay( thumb$agg, thumb1$agg,
            fun= function( t, t1) t -t1,
            filename= "thumb_nomos_diff.tif",
            overwrite= TRUE)
  } else brick( "thumb_nomos_diff.tif")
layerNames( thumbNomosDiff) <- layerNames( thumb$agg)

if( overwriteFigures) {
  thumbNomosDiffPlot <- coverMaps( thumbNomosDiff) +
    scale_fill_gradientn( "diff", colours= rev( brewer.
       pal( 11, "BrBG")),
                         limits= c( 0.32, -0.32),
```

```
                        ## breaks= { b <- c( 0.3, 0.15,
                            0.075, 0.075/2, 0.075/4)
                        ##            c( b, 0, rev( -b))
                        ##          })
                        breaks= { b <- c( 0.01, 0.02,
                            0.04, 0.08, 0.16, 0.32)
                                   c( rev( b), 0, -b)
                                 })
}



#####################################################
### chunk number 23: fig_thumb_nomos
#####################################################
#line 728 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_nomos.png")
  print( thumbPlots$nomos)
  dev.off()
}



#####################################################
### chunk number 24: fig_thumb1_nomos
#####################################################
#line 749 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb1_nomos.png")
  print( thumbPlots1$nomos)
  dev.off()
}



#####################################################
### chunk number 25: fig_thumb_nomos_diff
#####################################################
```

```
#line 770 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures ) {
  png( file="fig_thumb_nomos_diff.png")
  print( thumbNomosDiffPlot)
  dev.off()
}



################################################
### chunk number 26: mlct1
################################################
#line 793 "/home/nbest/thesis/datasets.Rnw"


setwd( dataWd)

mlct  <- primaryFraction( mlct, Amin= 0.5, overwrite=
    overwriteRasters, progress="text")
mlct  <- coverFractions( mlct, overwrite=
    overwriteRasters, progress="text")
mlct  <- aggregateFractions( mlct, overwrite=
    overwriteRasters, progress="text")
mlct  <- decomposeMosaic( mlct, overwrite=
    overwriteRasters, progress="text")

mlct1  <- primaryFraction( mlct, Amin=1.0, overwrite=
    overwriteRasters, progress="text")
mlct1  <- coverFractions( mlct1, overwrite=
    overwriteRasters, progress="text")
mlct1  <- aggregateFractions( mlct1, overwrite=
    overwriteRasters, progress="text")
mlct1  <- decomposeMosaic( mlct1, overwrite=
    overwriteRasters, progress="text")

## might be useful to cross-tabulate the primary and
    secondary
## frequencies for the cUSA

## table(thumbDf@data$pri, thumbDf@data$sec)
```

```
################################################
### chunk number 27: agland
################################################
#line 837 "/home/nbest/thesis/datasets.Rnw"

setwd( dataWd)
setwd( "agland")

agland <- stack( list.files( patt="(cropland|pasture).tif
   $"))
layerNames(agland) <- c("crop", "open")
agland <- setMinMax( agland)

thumbAgland <-
  if( overwriteRasters) {
    crop( agland,
         extent(-83.5, -(82+25/60),
                42+55/60, 44+5/60),
         filename= "thumbAgland.tif",
         progress="text",
         overwrite= overwriteRasters)
  } else brick( list.files( getwd(),
                           "thumbAgland.tif",
                           full.names= TRUE,
                           recursive= TRUE))
layerNames( thumbAgland) <- c("crop", "open")
                                        # crop() returns
                                            a brick

## overwriteFigures <- TRUE

if( overwriteFigures) {
  thumbAglandPlot <- coverMaps( thumbAgland, 1) + coord_
     equal()
  aglandPlot <- coverMaps( agland, 0.4) + coord_equal()
}

##     sapply( layerNames( thumbAgland),
##             function( cover) {
##               ggplotRaster( agland[[ cover]]) +
##                 scale_fill_gradientn( paste("%", cover)
```

```
    ,
##                                          colours=rev(
    brewer.pal( 7, "YlGn")),
##                                          limits= c( 100, 0)
    ,
##                                          breaks= seq( 100,
    0, by= -20))
##             })
## }
```

```
#####################################################
### chunk number 28: fig_thumb_agland
#####################################################
#line 885 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_agland.png")
  print( thumbAglandPlot)
  dev.off()
}
```

```
#####################################################
### chunk number 29: fig_agland
#####################################################
#line 905 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_agland.png")
  print( aglandPlot +coord_equal() +facet_wrap( ~
      variable, ncol= 1))
  dev.off()
}
```

```
#####################################################
```

```
### chunk number 30: mlu eval=FALSE
#################################################
## #line 934 "/home/nbest/thesis/datasets.Rnw"
##
## setwd( texWd)
##
## cusaDf <- readOGR("PG:host=db dbname=cim","gadm.cusa")
##
## cusa <- raster(mlct$pri)
## res(cusa) <- 5/60
##
## cusa <- rasterize( cusaDf, cusa, field= "id_1",
   filename= "gadm1_cusa.tif")
##
## foo <- raster("nlcd_agg.tif")
## gadm <- raster( foo)
## res(gadm) <- 15/3600
## gadm[] <- 0
## writeRaster(gadm, filename= "foo.tif", overwrite=TRUE,
    NAflag=0)
## system( "gdal_translate -ot UInt16 -a_nodata 0 foo.tif
    gadm1_cusa.tif")
##
## system( "gdal_rasterize -at -a id_1 -ot UInt16 -l gadm
   .cusa 'PG:host=db dbname=cim' gadm1_cusa.tif")
##


#################################################
### chunk number 31: thumb_nlcd
#################################################
#line 979 "/home/nbest/thesis/datasets.Rnw"

setwd( dataWd)
setwd( "nlcd")
nlcdWd <- getwd()

thumbNlcd <- list( pri=raster( "thumbNlcd.tif"))
## thumbNlcd <- sapply( thumbNlcd, setMinMax)



#################################################
```

```
### chunk number 32: thumb_nlcd_reclass
#################################################
#line 996 "/home/nbest/thesis/datasets.Rnw"

setwd( nlcdWd)


thumbNlcd <- mlctReclass( thumbNlcd, nlcdReclassMatrix,
                          overwrite= overwriteRasters,
                          progress="text")

if( overwriteFigures) {
  thumbNlcdPlot <- peelMap(thumbNlcd$pri, 0.05)
}



#################################################
### chunk number 33: fig_thumb_nlcd_reclass
#################################################
#line 1015 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_nlcd_reclass.png")
  print( thumbNlcdPlot + coord_equal())
  dev.off()
}



#################################################
### chunk number 34: fig_thumb_nlcd_facet
#################################################
#line 1035 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_nlcd_facet.png")
  print( thumbNlcdPlot +
         facet_wrap(~ values) +
         opts( legend.position= "none"))
  dev.off()
```

```
}



##################################################
### chunk number 35: thumb_nlcd_aggr
##################################################
#line 1066 "/home/nbest/thesis/datasets.Rnw"

setwd( nlcdWd)
## overwriteRasters <- TRUE
thumbNlcd$Amin <- 1
thumbNlcd <-
  coverFractions( thumbNlcd, mosaic=FALSE,
                 overwrite= overwriteRasters,
                 progress= "text")
thumbNlcd <-
  aggregateFractions( thumbNlcd,
                     overwrite= overwriteRasters,
                     progress="text")

if( overwriteFigures) {
  thumbNlcdAggPlot <- coverMaps( thumbNlcd$agg, 1)
}

## overwriteRasters <- FALSE



##################################################
### chunk number 36: fig_thumb_nlcd_agg
##################################################
#line 1090 "/home/nbest/thesis/datasets.Rnw"

## overwriteRasters <- TRUE
## overwriteFigures <- TRUE


setwd( texWd)
if( overwriteFigures) {
  png( file="fig_thumb_nlcd_agg.png")
  print( thumbNlcdAggPlot)
  dev.off()
}
```

```
#################################################
### chunk number 37: nlcd
#################################################
#line 1115 "/home/nbest/thesis/datasets.Rnw"

setwd( dataWd)

## nlcd <- stack( paste( "nlcd", names( peelClasses[ -8])
    , "5min.tif", sep="_"))

nlcd <- stack( sapply( names( peelClasses[ -8]),
                      function( cover) {
                        list.files( paste( dataWd, "nlcd"
                            , sep="/"),
                                   patt= paste( "nlcd",
                                       cover, "5min.tif$",
                                        sep="_"),
                                   full.names= TRUE)
                      }))
nlcd <- setMinMax( nlcd)
layerNames(nlcd) <- names( peelClasses[ -8])

nlcdPlot <- coverMaps( nlcd, 0.1)



#################################################
### chunk number 38: fig_nlcd
#################################################
#line 1138 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  png( file="fig_nlcd.png")
  print( nlcdPlot)
  dev.off()
}
```

```
##################################################
### chunk number 39: cdl eval=FALSE
##################################################
## #line 1179 "/home/nbest/thesis/datasets.Rnw"
##
##
## cdl <- list( pri= raster( "/gpfs/pads/projects/see/
   data/raw/cdl/vrt/cdl_2001.vrt"))
##
## cdlReclassMatrix <-
##   matrix( c(
##             83,  83,  0,                # water
##             85,  85,  0,
##             63,  63,  1,                # forest
##                                         # no shrub
##             62,  62,  3,                # open
##             88,  88,  3,
##             87,  87,  4,                # wetland
##              1,  61,  5,                # crop
##             90,  90,  5,
##             82,  82,  6,                # urban
##             84,  84,  6,
##             86,  86,  6),
##                                         # no mosaic
##                                         # no barren
##          ncol= 3, byrow= TRUE)
##
## cdl  <- mlctReclass( cdl, cdlReclassMatrix, overwrite=
    overwriteRasters, progress= "text")
##
## cdl$Amin <- 1
## cdl <- coverFractions( cdl, mosaic= FALSE, overwrite=
   TRUE, progress="text")
## cdl <- aggregateFractions( cdl, overwrite=TRUE,
   progress="text")
##
##
## cdl_il <- list( pri= raster( "cdl_il_2001.tif"))
## cdl_il <- mlctReclass( cdl_il, cdlReclassMatrix,
   overwrite= overwriteRasters, progress= "text")
## cdl_il$Amin <- 1
## cdl_il <- coverFractions( cdl_il, mosaic= FALSE,
```

```
   overwrite= TRUE, progress="text")
## cdl_il <- aggregateFractions( cdl_il, overwrite=TRUE,
   progress="text")
##


#################################################
### chunk number 40: 175crops eval=FALSE
#################################################
## #line 1249 "/home/nbest/thesis/datasets.Rnw"
##
## cropsWd <- path.expand( "~/see/data/raw/175crops2000/
   nc")
## list.files( cropsWd, "vrt$")
##
## cropTable <- read.csv( "/home/nbest/see/data/raw/175
   crops2000/monfreda2008 table1.csv", header= TRUE)
##
## ## For now we consider only herbaceous crops
## ## assume that forage crops will be classified "open"
##
## herbNotForage <- cropTable$type=="herbaceous" &
   cropTable$group != "Forage"
##
##
## cropTable$cat <- NA
## cropTable <- within( cropTable, {
##   cat[ map == "maize"] <- "maize"
##   cat[ map == "soybean"] <- "soybean"
##   cat[ map == "wheat"] <- "wheat"
##   cat[ map == "rice"] <- "rice"
##   cat[ group == "Cereals" & is.na( cat)] <- "cereals"
##   cat[ map == "sugarcane"] <- "sugarcane"
##   cat[ type == "herbaceous" & group == "Forage"] <- "
   forage"
##   cat[ type == "herbaceous" & is.na( cat)] <- "field_
   crop"
##   cat[ type == "shrub"] <- "shrub_crop"
##   cat[ type == "tree"] <- "tree_crop"
## })
##
##
## catLists <- dlply( cropTable, .(cat), function( row)
```

```
   row$map)
##
## mapNcName <- function( map) {
##   paste( cropsWd,
##         paste( map, "5min.vrt",
##               sep="_"),
##         sep="/")
## }
##
## catStacks <- llply( catLists, function( maps) {
##   if( length( maps) ==1) {
##     subset( brick( mapNcName( maps[ 1])), 1)
##   } else {
##     do.call( stack, llply( maps, function( map) {
##       subset( brick( mapNcName( map)), 1)
##     }))
##   }})
##
## cusaMask <- raster( list.files( getwd(),
##                                 "mask_cusa.tif",
##                                 recursive= TRUE))
## cusaExtent <- extent( cusaMask)
##
## catCropped <- llply( names( catStacks), function( c) {
##   fn <- paste( c, "crop.tif", sep="_")
##   if( overwriteRasters) {
##     crop( catStacks[[ c]], cusaExtent,
##          filename= fn,
##          overwrite= TRUE)
##   } else brick( list.files( getwd(), fn, full.names=
   TRUE))
## })
## names( catCropped) <- names( catStacks)
##
## catMasked <- llply( names( catCropped), function( c) {
##   r <- if( nlayers( catCropped[[ c]]) ==1) {
##     catCropped[[ c]]
##   } else overlay( catCropped[[ c]], fun= sum)
##   mask( r, cusaMask,
##        filename= paste( c, "tif", sep="."),
##        overwrite= TRUE)
## })
## names( catMasked) <- names( catStacks)
```

```
##
##


#################################################
### chunk number 41: fig_crops
#################################################
#line 1326 "/home/nbest/thesis/datasets.Rnw"

setwd( texWd)
if( overwriteFigures) {
  cropsMap <- coverMaps( stack( catMasked), 0.2) +
    coord_equal() +
      facet_grid( variable ~ .)
  ggsave( "fig_crops.png", width=5.5, height=8)
}
setwd( dataWd)
```

GRASS code used to process NLCD for cUSA

```
g.mapset NLCD

cat <<EOF | r.reclass input=nlcd2001 output=nlcd2001_r
11 98 99 = 0 water
41 thru 43 = 5 forest
51 52 94 = 7 shrub
71 thru 74 81 = 10 open
90 thru 93 95 thru 97 = 11 wetland
82 = 12 crop
21 thru 24 = 13 urban
12 31 32 = 16 barren
* = 99 other
EOF

g.mapset nlcd

for lct_label in $(r.category nlcd2001_r cats=0-99 fs=, |
    grep -v ',$'); do
    lct=$(echo $lct_label | cut -d, -f1)
    label=$(echo $lct_label | cut -d, -f2)
    cat <<EOF | r.reclass --overwrite input=nlcd2001_r
       output=nlcd_${label}
$lct = 1
```

```
* = 0
EOF
done

g.region -p res=0:05 align=grid_5min
for label in $(r.category map=nlcd2001_r cats=0-99 fs=, |
    grep -v ',$' | cut -d, -f2); do
    r.resamp.stats -w input=nlcd_${label} output=nlcd_${
       label}_5min method=average
done
```

Source Code: Analysis

R helper functions

```
library( raster)
library( rgdal)
library( ggplot2)
library( xtable)

## library( lattice)
## library( RColorBrewer)
## divTheme <- sp.theme(
##     regions= list(
##         col= colorRampPalette( brewer.pal( 5, "BrBG"),
##           space= "Lab")(100)))
## seqTheme <- sp.theme(
##     regions= list(
##         col= colorRampPalette( brewer.pal( 5, "YlGn"),
##           space="Lab")(100)))

##                                          # list the
   subdatasets and split on equal sign
##                                          # e.g.
   SUBDATASET_2_NAME=RASTERLITE:cusa.sqlite,table=agc_
   crop
## maps <- function() {
##   unlist( lapply( strsplit( grep( "NAME",
##                                  attr( GDALinfo( db,
   silent=TRUE),
##                                       "subdsmdata"),
##                                  value=T),
##                            "="),
##                  function(x) return(x[3])))
## }
```

```
## covers <- unlist( lapply( strsplit( maps()[ grep( "^
   agc", maps())], "_"),
##                          function(x) return( x[2])))

##                                          # arg: map
   name string
##                                          # res: fully-
   qualified DSN in db
## dataset  <- function( s) {
##   return( paste( "RASTERLITE:", db, ",table=", s,
##                 sep=""))
## }

##                                          # arg: map
   name string
##                                          # res: rgdal
   handle
## handle <- function(map) {
##   return( new( "GDALReadOnlyDataset", dataset( map)))
## }

##                                          # arg: rgdal
   handle
##                                          # res: spatial
    grid data frame
## as.spgdf <- function( handle) {
##   result <- as( handle, "SpatialGridDataFrame")
##   #names( result) <- names( handle)
##   return( result)
## }

## #nlcdCrop <- raster( as.spgdf( handle( "nlcd_crop")))
## #agcCrop <-  raster( as.spgdf( handle( "agc_crop")))


## grepHandles <- function( regex)
##   return( sapply( maps()[ grep( regex, maps())],
##                  handle))

## stackHandles <- function( handles) {
##   result <- stack( sapply( handles, function(x) raster
   ( as.spgdf( x))))
```

```
##   attr( result, "layernames") <- names( handles)
##   return( result)
## }

## areaAcres <- function( rast) {
##   result <- rast *area( rast) *247.105381
##   attr( result, "layernames") <- attr( rast, "
   layernames")
##   return( result)
##}

printAreas <- function( acres) {
  return( paste( round( acres /10^6, digits=1), "Ma␣(",
                round( acres /10^6 *0.404685642, digits
                   =1), "Mha)",
                sep=""))
}

getPeelBand <- function( peelBrick, cover) {
  unstack( peelBrick)[[ peelBands[[ cover]]]]
}

rmseRast <- function(obsRast, predRast) {
  sqErr <- overlay( obsRast, predRast,
                   fun=function( obs, pred) return(( obs
                      -pred) ^2))
  return( sqrt( cellStats( sqErr, 'mean')))
}

biasRast <- function(obsRast, predRast) {
  err <- overlay( obsRast, predRast,
                   fun=function( obs, pred) return( obs -
                      pred))
  return( cellStats( err, 'mean'))
}



rmseSummary <- function( obsNameFun, predNameFun) {
  sapply( covers,
         function( c) {
           obsRast  <- raster( as.spgdf( handle(
              obsNameFun(c))))
```

```
            predRast <- raster( as.spgdf( handle(
                predNameFun(c))))
            if( extent( obsRast) != extent( predRast)) {
              intExt <- intersectExtent( obsRast, predRast
                 )
              obsRast <- crop( obsRast, intExt)
              predRast <- crop( predRast, intExt)
            }
            return( c( rmse_frac= rmseRast( obsRast,
                predRast),
                       bias_frac= biasRast( obsRast,
                           predRast),
                      rmse_acres= rmseRast( areaAcres(
                          obsRast), areaAcres( predRast)),
                      bias_acres= biasRast( areaAcres(
                          obsRast), areaAcres( predRast))))
          })
}
```

R code embedded in the chapter

```
##################################################
### chunk number 1: init
##################################################
#line 16 "/home/nbest/thesis/analysis.Rnw"

setwd( "~/thesis/analysis")

texWd <- setwd("../data")
dataWd <- getwd()
setwd( dataWd)

#rm(list=ls())
db <- "cusa.sqlite"
source( "~/thesis/code/analysis.R")
source( "~/thesis/code/peel.R")

## calculate areas in millions of acres (Ma)
areas <- sapply( list( ag="^ag_", As00="As00$", As05="
    As05$", agc="^agc", nlcd="^nlcd"),
                 function( re) return( cellStats(
                     areaAcres( stackHandles( grepHandles(
                      re))),
```

```
                                                    sum)
                                          /10 ^6))
names(areas$ag) <- c("crop", "open")
names(areas$As00) <- sort( c(covers, "mosaic"))
names(areas$As05) <- sort( c(covers, "mosaic"))
names(areas$agc) <- covers
names(areas$nlcd) <- covers

## merge the resulting structure into a data frame

areasDf <- merge( t( unlist( areas$As00)),
                  t( unlist( areas$As05)),
                  all=TRUE, sort=FALSE)
areasDf <- merge( areasDf, t( unlist( areas$ag)),
                 all=TRUE, sort=FALSE)
areasDf <- merge( areasDf, t( unlist( areas$agc)),
                 all=TRUE, sort=FALSE)
areasDf <- merge( areasDf, t( unlist( areas$nlcd)),
                 all=TRUE, sort=FALSE)
rownames(areasDf) <- c( "As00", "As05", "ag", "agc", "
    nlcd")


printAreas <- function( Ma) {
  return( paste( round( Ma, digits=1), "Ma (", round( Ma
      *0.404685642, digits=1), "Mha)", sep=""))
}



##################################################
### chunk number 2: tabTotal
##################################################
#line 60 "/home/nbest/thesis/analysis.Rnw"
print( xtable( areasDf,
              caption= "Total Acreages by Map and Cover",
              label= "tab:total",
              digits= 1))


##################################################
### chunk number 3: bias
##################################################
```

```
#line 98 "/home/nbest/thesis/analysis.Rnw"

# calculate RMSE/bias summaries
# comparing everything to NLCD


rmseAgc <- rmseSummary( function(c) paste(  "agc", c, sep
   ="_"),
                        function(c) paste( "nlcd", c, sep
                           ="_"))

rmseAs00 <- rmseSummary( function(c) paste( "mlct_2001",
   c, "As00", sep="_"),
                         function(c) paste( "nlcd", c,
                            sep="_"))

rmseAs05 <- rmseSummary( function(c) paste( "mlct_2001",
   c, "As05", sep="_"),
                         function(c) paste( "nlcd", c,
                            sep="_"))


##################################################
### chunk number 4: biasTab
##################################################
#line 114 "/home/nbest/thesis/analysis.Rnw"
## t( rmseAgc)
## t( rmseAs00)
## t( rmseAs05)

print( xtable( t( rmseAgc),
              caption= "Errors␣and␣Biases␣of␣Aglands␣
                 Complete␣relative␣to␣NLCD",
              label= "tab:ebagc",
              digits= c( 0, 2, -2, 0, 0)))

print( xtable( t( rmseAs00),
              caption= "Errors␣and␣Biases␣of␣MLCT,␣$A_s␣=
                 ␣0.0$␣relative␣to␣NLCD",
              label= "tab:ebmlct00",
              digits= c( 0, 2, -2, 0, 0)))

print( xtable( t( rmseAs05),
```

```
              caption= "Errors␣and␣Biases␣of␣MLCT,␣$A_s␣=
                 ␣0.5$␣relative␣to␣NLCD",
              label= "tab:ebmlct05",
              digits= c( 0, 2, -2, 0, 0)))


#################################################
### chunk number 5: stack
#################################################
#line 135 "/home/nbest/thesis/analysis.Rnw"
## agcAvgAcres <-
##   sapply( paste( "agc_", covers, sep=""),
##           function( map) {
##             mapRast <- raster( as.spgdf( handle( map)))
##             return( cellStats( areaAcres( mapRast), sum
   )
##                   /( ncell( mapRast) - cellStats(
   mapRast, 'countNA')))
##           })


## getting ready to plot

stackAgc <- stackHandles( grepHandles(  "^agc"))
attr( stackAgc, "layernames") <-  covers

stackNlcd <- stackHandles( grepHandles( "^nlcd"))
attr( stackNlcd, "layernames") <-  covers

stackDiff <- stackAgc -stackNlcd
attr( stackDiff, "layernames") <-  covers



#################################################
### chunk number 6: fig_agc
#################################################
#line 162 "/home/nbest/thesis/analysis.Rnw"

#spgdfAgc <- as.spgdf( stackAgc)
#names( spgdfAgc) <- layerNames( stackAgc)
setwd( texWd)
png( file="fig_agc.png")
```

```
print( coverMaps( stackAgc, 0.4))
dev.off()


################################################
### chunk number 7: fig_nlcd
################################################
#line 182 "/home/nbest/thesis/analysis.Rnw"

##spgdfNlcd <- as.spgdf( stackNlcd)
##names( spgdfNlcd) <- layerNames( stackNlcd)
setwd( texWd)
png( file="fig_nlcd.png")
print( coverMaps( stackNlcd, 0.4))
dev.off()


################################################
### chunk number 8: fig_diff
################################################
#line 202 "/home/nbest/thesis/analysis.Rnw"

##spgdfDiff <- as.spgdf( stackDiff)
##names( spgdfDiff) <- layerNames( stackDiff)
setwd( texWd)
png( file="fig_diff.png")
print( coverMaps( stackDiff, 0.4) +
  scale_fill_gradientn( "diff", colours= rev( brewer.pal(
       11, "BrBG")),
                        limits= c( 0.1, -0.1),
                        breaks= seq( 0.1, -0.1, by=
                            -0.02)))
dev.off()


################################################
### chunk number 9: fig_cordiff
################################################
#line 225 "/home/nbest/thesis/analysis.Rnw"

## look for correlations across the difference maps

corDiff <- cor( as.data.frame( as.spgdf( stackDiff))
```

```
   [,1:8])
colnames( corDiff) <- unlist( lapply(
    strsplit( colnames( corDiff), "\\."),
    function( x) return( x[ 2])))
rownames( corDiff) <- unlist( lapply(
    strsplit( rownames( corDiff), "\\."),
    function( x) return( x[ 2])))
ord <- order.dendrogram( as.dendrogram( hclust( dist(
   corDiff))))

corDiffPlot <-
  ggplot( melt( corDiff),
         aes( x=X1, y=X2, fill= value)) +
  geom_tile() +
  theme_bw() +
  opts( panel.grid.minor= theme_blank(),
       panel.grid.major= theme_blank(),
       panel.background= theme_blank(),
       axis.title.x= theme_blank(),
       axis.text.x= theme_text( angle= 90, hjust=1),
       axis.title.y= theme_blank()) +
  scale_x_discrete( limits= colnames(corDiff)[ord]) +
  scale_y_discrete( limits= colnames(corDiff)[ord]) +
  scale_fill_gradientn( "cor", colours= rev( brewer.pal(
     11, "BrBG")),
                        limits= c( 1.0, -1.0),
                        breaks= seq( 1.0, -1.0, by=
                           -0.2))


setwd( texWd)
png( file="fig_cordiff.png")
print( corDiffPlot)
dev.off()


##################################################
### chunk number 10: analysis
##################################################
#line 272 "/home/nbest/thesis/analysis.Rnw"
setwd( "~/thesis")
```

GRASS code used to compute NLCD offsets

```
g.mapset -c thesis

# by setting the resolution for the computation region
# the inputs are automatically resampled from 15 arcsecs
# to 1.25 arcsecs, which is the NLCD's nominal resolution

g.region -p lower48_5min@peel res=0:00:01.25


# the inputs are the per-class fractions computed based
    on
# $A_min=0.5$

r.in.gdal input=mlct_Amin_0.5_fracs.tif output=mlct_fracs


# we are only interested in certain classes from the NLCD

r.reclass input=nlcd2001@NLCD output=nlcd_offsets_mask <<
    EOF
11 = 0 water
90 thru 93 95 thru 97 = 4 wetland
21 thru 24 = 6 urban
EOF


class=( water forest shrub open wetland crop urban mosaic
     barren )

# generate the GRASS commands to calculate the 1.25-
    arcsec per-pixel fractional adjustments.
# the outer if() clause converts NULL to 0 for pixels not
     in the NLCD classes in question.
# the inner if() clause accounts for the possibility that
     NLCD and MLCT agree to some degree.

for nlcd in 0 4 6; do
    cmd="r.mask␣-o␣input=nlcd_offsets_mask␣maskcats=$nlcd
        "
    echo $cmd
    for mlct in $(seq 1 9); do
```

```
        cmd="r.mapcalc␣nlcd_offset_${class[$nlcd]}_${mlct
            }=\"if( isnull(MASK@thesis), 0, if( ${nlcd} ==
            $((mlct-1)), 1, 0) -mlct_fracs.${mlct})\"␣&"
        echo $cmd
    done
    echo
done


# the following groups of commands are intended to be run
    manually
# running this script automatically will produce
   unintended results due to changes
# in region and mask settings before completion of prior
   steps

r.mask -o input=nlcd_offsets_mask maskcats=0
r.mapcalc nlcd_offset_water_1="if(␣isnull(MASK@thesis),␣
   0,␣if(␣0␣==0,␣1,␣0)␣-mlct_fracs.1)" &
r.mapcalc nlcd_offset_water_2="if(␣isnull(MASK@thesis),␣
   0,␣if(␣0␣==1,␣1,␣0)␣-mlct_fracs.2)" &
r.mapcalc nlcd_offset_water_3="if(␣isnull(MASK@thesis),␣
   0,␣if(␣0␣==2,␣1,␣0)␣-mlct_fracs.3)" &
r.mapcalc nlcd_offset_water_4="if(␣isnull(MASK@thesis),␣
   0,␣if(␣0␣==3,␣1,␣0)␣-mlct_fracs.4)" &
r.mapcalc nlcd_offset_water_5="if(␣isnull(MASK@thesis),␣
   0,␣if(␣0␣==4,␣1,␣0)␣-mlct_fracs.5)" &
r.mapcalc nlcd_offset_water_6="if(␣isnull(MASK@thesis),␣
   0,␣if(␣0␣==5,␣1,␣0)␣-mlct_fracs.6)" &
r.mapcalc nlcd_offset_water_7="if(␣isnull(MASK@thesis),␣
   0,␣if(␣0␣==6,␣1,␣0)␣-mlct_fracs.7)" &
r.mapcalc nlcd_offset_water_8="if(␣isnull(MASK@thesis),␣
   0,␣if(␣0␣==7,␣1,␣0)␣-mlct_fracs.8)" &
r.mapcalc nlcd_offset_water_9="if(␣isnull(MASK@thesis),␣
   0,␣if(␣0␣==8,␣1,␣0)␣-mlct_fracs.9)" &

r.mask -o input=nlcd_offsets_mask maskcats=4
r.mapcalc nlcd_offset_wetland_1="if(␣isnull(MASK@thesis),
   ␣0,␣if(␣4␣==0,␣1,␣0)␣-mlct_fracs.1)" &
r.mapcalc nlcd_offset_wetland_2="if(␣isnull(MASK@thesis),
   ␣0,␣if(␣4␣==1,␣1,␣0)␣-mlct_fracs.2)" &
r.mapcalc nlcd_offset_wetland_3="if(␣isnull(MASK@thesis),
   ␣0,␣if(␣4␣==2,␣1,␣0)␣-mlct_fracs.3)" &
```

```
r.mapcalc nlcd_offset_wetland_4="if(␣isnull(MASK@thesis),
   ␣0,␣if(␣4␣==3,␣1,␣0)␣-mlct_fracs.4)" &
r.mapcalc nlcd_offset_wetland_5="if(␣isnull(MASK@thesis),
   ␣0,␣if(␣4␣==4,␣1,␣0)␣-mlct_fracs.5)" &
r.mapcalc nlcd_offset_wetland_6="if(␣isnull(MASK@thesis),
   ␣0,␣if(␣4␣==5,␣1,␣0)␣-mlct_fracs.6)" &
r.mapcalc nlcd_offset_wetland_7="if(␣isnull(MASK@thesis),
   ␣0,␣if(␣4␣==6,␣1,␣0)␣-mlct_fracs.7)" &
r.mapcalc nlcd_offset_wetland_8="if(␣isnull(MASK@thesis),
   ␣0,␣if(␣4␣==7,␣1,␣0)␣-mlct_fracs.8)" &
r.mapcalc nlcd_offset_wetland_9="if(␣isnull(MASK@thesis),
   ␣0,␣if(␣4␣==8,␣1,␣0)␣-mlct_fracs.9)" &

r.mask -o input=nlcd_offsets_mask maskcats=6
r.mapcalc nlcd_offset_urban_1="if(␣isnull(MASK@thesis),␣
   0,␣if(␣6␣==0,␣1,␣0)␣-mlct_fracs.1)" &
r.mapcalc nlcd_offset_urban_2="if(␣isnull(MASK@thesis),␣
   0,␣if(␣6␣==1,␣1,␣0)␣-mlct_fracs.2)" &
r.mapcalc nlcd_offset_urban_3="if(␣isnull(MASK@thesis),␣
   0,␣if(␣6␣==2,␣1,␣0)␣-mlct_fracs.3)" &
r.mapcalc nlcd_offset_urban_4="if(␣isnull(MASK@thesis),␣
   0,␣if(␣6␣==3,␣1,␣0)␣-mlct_fracs.4)" &
r.mapcalc nlcd_offset_urban_5="if(␣isnull(MASK@thesis),␣
   0,␣if(␣6␣==4,␣1,␣0)␣-mlct_fracs.5)" &
r.mapcalc nlcd_offset_urban_6="if(␣isnull(MASK@thesis),␣
   0,␣if(␣6␣==5,␣1,␣0)␣-mlct_fracs.6)" &
r.mapcalc nlcd_offset_urban_7="if(␣isnull(MASK@thesis),␣
   0,␣if(␣6␣==6,␣1,␣0)␣-mlct_fracs.7)" &
r.mapcalc nlcd_offset_urban_8="if(␣isnull(MASK@thesis),␣
   0,␣if(␣6␣==7,␣1,␣0)␣-mlct_fracs.8)" &
r.mapcalc nlcd_offset_urban_9="if(␣isnull(MASK@thesis),␣
   0,␣if(␣6␣==8,␣1,␣0)␣-mlct_fracs.9)" &



r.mask -r

# generate the commands to merge the per-NLCD class
   offsets into a total offset for each MLCT class

for mlct in $(seq 1 9); do
    cmd="r.mapcalc␣nlcd_offset_total_${mlct}=\"(nlcd_
       offset_water_${mlct}+nlcd_offset_wetland_${mlct}+
```

```
        nlcd_offset_urban_${mlct})\" &"
    echo $cmd
done


# only NULLs are outside the lower48 mask since they were
    replaced with zeroes everywhere else
# so we don't have to worry about trapping NULLs in this
   step

r.mapcalc nlcd_offset_total_1="(nlcd_offset_water_1+nlcd_
   offset_wetland_1+nlcd_offset_urban_1)" &
r.mapcalc nlcd_offset_total_2="(nlcd_offset_water_2+nlcd_
   offset_wetland_2+nlcd_offset_urban_2)" &
r.mapcalc nlcd_offset_total_3="(nlcd_offset_water_3+nlcd_
   offset_wetland_3+nlcd_offset_urban_3)" &
r.mapcalc nlcd_offset_total_4="(nlcd_offset_water_4+nlcd_
   offset_wetland_4+nlcd_offset_urban_4)" &
r.mapcalc nlcd_offset_total_5="(nlcd_offset_water_5+nlcd_
   offset_wetland_5+nlcd_offset_urban_5)" &
r.mapcalc nlcd_offset_total_6="(nlcd_offset_water_6+nlcd_
   offset_wetland_6+nlcd_offset_urban_6)" &
r.mapcalc nlcd_offset_total_7="(nlcd_offset_water_7+nlcd_
   offset_wetland_7+nlcd_offset_urban_7)" &
r.mapcalc nlcd_offset_total_8="(nlcd_offset_water_8+nlcd_
   offset_wetland_8+nlcd_offset_urban_8)" &
r.mapcalc nlcd_offset_total_9="(nlcd_offset_water_9+nlcd_
   offset_wetland_9+nlcd_offset_urban_9)" &


# aggregate to 5-arcmin resolution

g.region -p res=0:05
r.mask -o mask_lower48_5min@GADM

for mlct in $(seq 1 9); do
    cmd="r.resamp.stats --o input=nlcd_offset_total_${
       mlct} output=nlcd_offset_${mlct} method=average &"
    echo $cmd
done

r.resamp.stats --o input=nlcd_offset_total_1 output=nlcd_
   offset_1 method=average &
```

```
r.resamp.stats --o input=nlcd_offset_total_2 output=nlcd_
   offset_2 method=average &
r.resamp.stats --o input=nlcd_offset_total_3 output=nlcd_
   offset_3 method=average &
r.resamp.stats --o input=nlcd_offset_total_4 output=nlcd_
   offset_4 method=average &
r.resamp.stats --o input=nlcd_offset_total_5 output=nlcd_
   offset_5 method=average &
r.resamp.stats --o input=nlcd_offset_total_6 output=nlcd_
   offset_6 method=average &
r.resamp.stats --o input=nlcd_offset_total_7 output=nlcd_
   offset_7 method=average &
r.resamp.stats --o input=nlcd_offset_total_8 output=nlcd_
   offset_8 method=average &
r.resamp.stats --o input=nlcd_offset_total_9 output=nlcd_
   offset_9 method=average &


# check the results; should sum to zero everywhere


offsets=$(g.mlist -r type=rast mapset=thesis patt="nlcd_
   offset_.$" sep=,)
r.series input=$offsets output=nlcd_offset_total method=
   sum
r.univar nlcd_offset_total

# total null and non-null cells: 207110
# total null cells: 85179
#
# Of the non-null cells:
# ----------------------
# n: 121931
# minimum: -1.19766e-08
# maximum: 1.31818e-08
# range: 2.51585e-08
# mean: -1.72333e-11
# mean of absolute values: 1.89202e-10
# standard deviation: 0
# variance: 0
# variation coefficient: -0 %
# sum: -0.0000021013
```

```
# stack and export back to file system for consumption by
    R

cd ~/thesis/data/analysis

i.group group=nlcd_offset input=${offsets}
r.out.gdal input=nlcd_offset output=nlcd_offset.tif
```